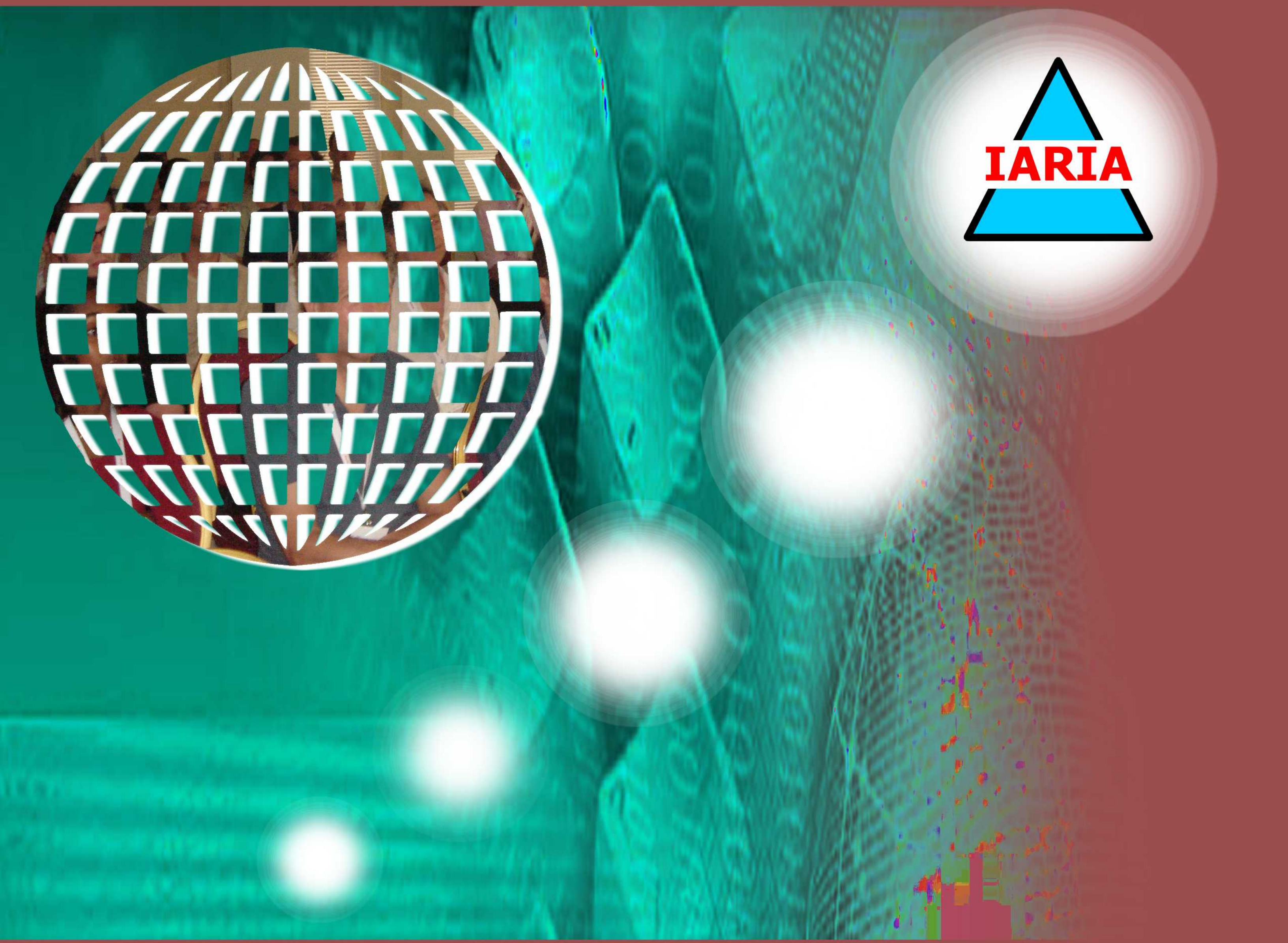
International Journal on Advances in Software



2021 vol. 14 nr. 1&2

The *International Journal on Advances in Software* is published by IARIA.
ISSN: 1942-2628
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Software
Volume 14, Number 1 & 2, 2021

Dayang Norhayati Abang Jawawi, Universiti Teknologi Malaysia (UTM), Malaysia
Jinyuan Jia, Tongji University. Shanghai, China
Maria Joao Ferreira, Universidade Portucalense, Portugal
Ahmed Kamel, Concordia College, Moorhead, Minnesota, USA
Teemu Kanstrén, VTT Technical Research Centre of Finland, Finland
Nittaya Kerdprasop, Suranaree University of Technology, Thailand
Ayad ali Keshlaf, Newcastle University, UK
Nhien An Le Khac, University College Dublin, Ireland
Sadegh Kharazmi, RMIT University - Melbourne, Australia
Kyoung-Sook Kim, National Institute of Information and Communications Technology, Japan
Youngjae Kim, Oak Ridge National Laboratory, USA
Cornel Klein, Siemens AG, Germany
Alexander Knapp, University of Augsburg, Germany
Radek Koci, Brno University of Technology, Czech Republic
Christian Kop, University of Klagenfurt, Austria
Michal Krátký, VŠB - Technical University of Ostrava, Czech Republic
Narayanan Kulathuramaiyer, Universiti Malaysia Sarawak, Malaysia
Satoshi Kurihara, Osaka University, Japan
Eugenijus Kurilovas, Vilnius University, Lithuania
Alla Lake, Linfo Systems, LLC, USA
Fritz Laux, Reutlingen University, Germany
Luigi Lavazza, Università dell'Insubria, Italy
Fábio Luiz Leite Júnior, Universidade Estadual da Paraiba,Brazil
Alain Lelu, University of Franche-Comté / LORIA, France
Cynthia Y. Lester, Georgia Perimeter College, USA
Clement Leung, Hong Kong Baptist University, Hong Kong
Weidong Li, University of Connecticut, USA
Corrado Loglisci, University of Bari, Italy
Francesco Longo, University of Calabria, Italy
Sérgio F. Lopes, University of Minho, Portugal
Pericles Loucopoulos, Loughborough University, UK
Alen Lovrencic, University of Zagreb, Croatia
Qifeng Lu, MacroSys, LLC, USA
Xun Luo, Qualcomm Inc., USA
Stephane Maag, Telecom SudParis, France
Ricardo J. Machado, University of Minho, Portugal
Maryam Tayefeh Mahmoudi, Research Institute for ICT, Iran
Nicos Malevris, Athens University of Economics and Business, Greece
Herwig Mannaert, University of Antwerp, Belgium
José Manuel Molina López, Universidad Carlos III de Madrid, Spain
Francesco Marcelloni, University of Pisa, Italy
Eda Marchetti, Consiglio Nazionale delle Ricerche (CNR), Italy
Gerasimos Marketos, University of Piraeus, Greece
Abel Marrero, Bombardier Transportation, Germany
Adriana Martin, Universidad Nacional de la Patagonia Austral / Universidad Nacional del Comahue, Argentina
Goran Martinovic, J.J. Strossmayer University of Osijek, Croatia
Paulo Martins, University of Trás-os-Montes e Alto Douro (UTAD), Portugal
Stephan Mäs, Technical University of Dresden, Germany
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Jose Merseguer, Universidad de Zaragoza, Spain
Seyedeh Leili Mirtaheri, Iran University of Science & Technology, Iran
Lars Moench, University of Hagen, Germany
Yasuhiko Morimoto, Hiroshima University, Japan

Kewei Sha, Oklahoma City University, USA
Simeon Simoff, University of Western Sydney, Australia
Jacques Simonin, Institut Telecom / Telecom Bretagne, France
Cosmin Stoica Spahiu, University of Craiova, Romania
George Spanoudakis, City University London, UK
Cristian Stanciu, University Politehnica of Bucharest, Romania
Lena Strömbäck, SMHI, Sweden
Osamu Takaki, Japan Advanced Institute of Science and Technology, Japan
Antonio J. Tallón-Ballesteros, University of Seville, Spain
Wasif Tanveer, University of Engineering & Technology - Lahore, Pakistan
Ergin Tari, Istanbul Technical University, Turkey
Steffen Thiel, Furtwangen University of Applied Sciences, Germany
Jean-Claude Thill, Univ. of North Carolina at Charlotte, USA
Pierre Tiako, Langston University, USA
Božo Tomas, HT Mostar, Bosnia and Herzegovina
Davide Tosi, Università degli Studi dell'Insubria, Italy
Guglielmo Trentin, National Research Council, Italy
Dragos Truscan, Åbo Akademi University, Finland
Chrisa Tsinaraki, Technical University of Crete, Greece
Roland Ukor, FirstLinq Limited, UK
Torsten Ullrich, Fraunhofer Austria Research GmbH, Austria
José Valente de Oliveira, Universidade do Algarve, Portugal
Dieter Van Nuffel, University of Antwerp, Belgium
Shirshu Varma, Indian Institute of Information Technology, Allahabad, India
Konstantina Vassilopoulou, Harokopio University of Athens, Greece
Miroslav Velev, Aries Design Automation, USA
Tanja E. J. Vos, Universidad Politécnica de Valencia, Spain
Krzysztof Walczak, Poznan University of Economics, Poland
Yandong Wang, Wuhan University, China
Rainer Weinreich, Johannes Kepler University Linz, Austria
Stefan Wesarg, Fraunhofer IGD, Germany
Wojciech Wiza, Poznan University of Economics, Poland
Martin Wojtczyk, Technische Universität München, Germany
Hao Wu, School of Information Science and Engineering, Yunnan University, China
Mudasser F. Wyne, National University, USA
Zhengchuan Xu, Fudan University, P.R.China
Yiping Yao, National University of Defense Technology, Changsha, Hunan, China
Stoyan Yordanov Garbatov, Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento, INESC-ID, Portugal
Weihai Yu, University of Tromsø, Norway
Wenbing Zhao, Cleveland State University, USA
Hong Zhu, Oxford Brookes University, UK
Martin Zinner, Technische Universität Dresden, Germany

International Journal on Advances in Software
Volume 14, Numbers 1 & 2, 2021

CONTENTS

Tugkan Tuglular, Izmir Institute of Technology, Turkey
Nazım Umut Ekici, Izmir Institute of Technology, Turkey

Approaches to Coherent Conceptual Knowledge Integration for Prehistory, Archaeology, Natural Sciences, and Humanities:

Information Science Based Computation of Structural Knowledge and Spatial Context Information

Claus-Peter Rückemann
Westfälische Wilhelms-Universität Münster (WWU), Germany;
Unabhängiges Deutsches Institut für Multi-disziplinäre Forschung (DIMF), Germany;
Leibniz Universität Hannover, Germany
Email: ruckema@uni-muenster.de

Abstract—**This extended paper delivers new results from the long-term information science research on creation and deployment of structure and cognostic addressing of structural knowledge ('nucleal cognstructure') for problem solving. The paper presents approaches towards the new methodological base and methodology focussing on structure-based fusion solutions and implementations for computational scenarios, especially advanced knowledge-centric mining. The paper presents two major cases of advanced multi-disciplinary coherent conceptual knowledge contextualisation and integration for prehistory, namely structure-based fusion realisation and integrated spatial context computation. Both cases include results on practical implementation components, and associated advances on computation and processing of multi-disciplinary and multi-lingual knowledge object entities and resources. The goal of this fundamental research is to create structure-based methods for efficient problem solving. Case studies implement the methods to consistently integrate knowledge context from prehistory and archaeology disciplines with knowledge in natural sciences and humanities. Previously unpublished insights on prehistoric cists tombs, contexts, and methods are available with this publication.**

Keywords–Prehistory; Speleology; Multi-disciplinary Knowledge Integration; Structure-based Information Science Methods; Computation and Processing.

I. Introduction

In information science, context is becoming increasingly relevant for many kinds of knowledge integration. This paper is an extended presentation of the research based on the publication and presentation at the INFOCOMP 2020 conference in Lisbon, Portugal [1]. The long-term project addresses two major challenges of coherent conceptual knowledge integration for prehistory, archaeology, natural sciences, and humanities: The integration of respective conceptual knowledge with structural knowledge and spatial information. The major goal is to create a framework of suitable long-term components for multi-disciplinary knowledge complements. It has become common practice during the last decades to tackle challenges regarding knowledge and related content solely with procedural approaches, besides the fact that creation processes, handling, and management may allow more effective and efficient measures in context of computation, processing, analysis, and long-term development or resources. Common ways, implementing procedural [2] approaches as plain technical solutions are often neither effective nor efficient. In addition, such approaches often lack long-term adaptability and scalability.

Procedure-based approaches are largely not addressing the knowledge and information content. Structure-based approaches can fill the gap. This motivation is supported by the experience that inefficiencies of procedural approaches regarding their creation, development, and execution can often be avoided by focussing on structure. Compared to procedure-based approaches, structure-based approaches are in general substantially different. Structure and formatting contain valuable information and closely correspond with logic, which should not be lost in many cases, e.g., this is especially the case for any sustainable long-term knowledge. That means, it is not intended to convert structures or to change formatting of resources. For information science and a universal knowledge context, meaning knowledge complements including conceptual knowledge, we also require a consistent, advanced definition of structure. It is important to understand what structure and form mean in theory and practice. The fundaments will be delivered before practical implementations are discussed.

The presented methodology addresses the shortcomings of common procedure-based approaches. The new structure-based fusion methodology and method implementations presented here are created as general purpose implementations, which can be realised in any Turing complete machine programming language, supporting creation and development of resources as well as computation, processing, and analysis. For the illustrative case studies, these approaches are used for dealing with various aspects of knowledge management and knowledge mining in information science context.

Knowledge resources' structures are commonly set and have proven long-term flexibility and sustainability. They cover content, context, consistency, and sustainability features for millions of information content, references, and object entities for long periods of time. Therefore, solutions for procedural components have to be found, which do provide comparably defiant long-term flexibility and sustainability. In addition, the procedural components require a very high level of knowledge-centricity and may need to exploit resource features, which are commonly not paid attention. When working with long-term resources, structural knowledge on resources and entities and their organisation has shown to provide a high potential. Inherent structural information also promises to achieve a high level of integration. Therefore, it might seem obvious to consider structure-based approaches for advanced and challenging tasks.

The rest of this paper is organised as follows. Section II gives an overview of previous work and resources development. Section III introduces to structure and addressing. Section IV presents an implementation case for structure-based methodology and method realisation. Section V discusses an implementation case of spatial context computation. Section VI summarises conclusions and future work.

II. Previous work, components, and resources

A. Fundaments

The fundaments of terminology and understanding knowledge are layed out by Aristotle [3], being an essential part of 'Ethics' [4] and can lead to an understanding of the knowledge complements employed in the following approaches. Information sciences can very much benefit from Aristotle's fundaments and a knowledge-centric approach [2] but for building holistic and sustainable solutions, supporting a modern definition of knowledge [5], they need to go beyond the available technology-based approaches and hypothesis [6] as analysed in Platon's Phaidon. For the implementation of case studies, the modules are built by support of a number of major components and resources, which can be used for a wide range of applications, e.g., creation of resources and extraction of entities. Here, it is essential to regard the most important fundaments of structural knowledge and information of language and respective content.

The concept of meaning differs from the concept of signification. Semantic and syntactic structures do not suffice to determine the discursive meaning of an expression [7]. Discourse means a way of speaking. On the one hand, grammatically correct phrases may lack discursive meaning. On the other hand, grammatically incorrect sentences may be discursively meaningful. Knowledge and meaning are closely tied with intrinsic and extrinsic properties. Therefore, understanding of intrinsic and extrinsic properties of entities is significant for any context. This is nevertheless true for any case of natural language, especially considering langage, langue, and parole [8], especially when interpretation [9] and meaning [10] should be considered, especially regarding cognition and insight [11].

B. Components, conceptual knowledge and structures

The Universal Decimal Classification (UDC) [12] is the world's foremost document indexing language in the form of a multi-lingual classification scheme covering all fields of knowledge and constitutes a sophisticated indexing and retrieval tool. The UDC is designed for subject description and indexing of content of information resources irrespective of the carrier, form, format, and language. UDC is an analytico-synthetic and facetted classification. UDC schedules are organised as a coherent system of knowledge with associative relationships and references between concepts and related fields. UDC-based references in this publication are taken from the multi-lingual UDC summary (UDCC Publication No. 088) [12] released by the UDC Consortium under a Creative Commons license [13]. Facets can be created with any auxiliary tables.

A means of choice to achieve overall efficient realisations even for complex scenarios is to use the principles of Superordinate Knowledge, integrating arbitrary knowledge. The core assembly elements of Superordinate Knowledge [14], e.g., for practical mathematical-computational scenarios [15], are:

- Methodology.
- Implementation.
- Realisation.

Comprehensive focussed subsets of conceptual knowledge can also provide excellent modular and standardised complements for information systems component implementations, e.g., for environmental information management and computation [16]. The presented implementations strictly follow the fundamental methodological algorithm base of the the Conceptual Knowledge Pattern Matching (CKPM) methodology [17] providing and accessing knowledge object patterns based on the Superordinate Knowledge Methodology, which allows systematical use and thorough processing. Core eager beaver procedure- and structure-based implementation components, grep and join, are written in C, as commonly known. Module examples are employing Perl Compatible Regular Expressions (PCRE) [18] syntax for specifying common string patterns. This is independent from the procedural realisation using Shell and Perl [19] for component wrapping purposes with case studies.

C. Prehistory long-term resources development

Several major Knowledge Resources (KR) and reference implementations are used with this multi-disciplinary research in prehistory, archaeology, natural sciences, and humanities and in long-term development. In order to overcome shortcomings of public 'data collections' the objects, entities, and respective conceptual knowledge references' excerpts and examples are taken from The Prehistory and Archaeology Knowledge Archive (PAKA). PAKA has been in continuous development for more than three decades [20] and is further developed and released by DIMF [21]. A main reference implementation in development, used in practice with ongoing long-term research, applied for KR is the prehistory-protohistory and archaeology Conceptual Knowledge Reference Implementation (CKRI) edition, including multi-disciplinary context references of natural sciences and humanities (E.0.4.2) [22], [23].

III. Structure and information

Structure is an organisation of interrelated entities in a material or non-material object or system. Structure is essential in logic as it carries unique information. The more, we have to recognise the differences of structure and form. The case of text is a good example. The structure of a text consists of the particular text units and their context, in order to make the text coherent. The form of a text is the arrangement of the text units, which commonly has to follow predefined rules.

A. Structure systematics, meaning, levels

As meaning can be understood in context of langage, langue, and parole, the available rules of structure and form should be used. If whatever non linguistic, artistic expression is primary target then different structure and form could be used. Anyway any linguistic parole context should be aware of the specific conditions: Academic use should be aware of the specific academic context. Commercial use should be aware of the specific commercial context. Marketing use should be aware of the specific marketing context. There are rarely reasonable compromises fitting diametrical approaches to form equally well. Consequences of these fundaments are, especially: Structure is not dependent on physical, non physical, analogous, digital or comparable being and properties. Structure is not dependent on same/uniform ways of structuring. Structure is not intrinsic to a certain scale of information. Structure should not imagined to be dependent of a location or dependent of management. Instead, it is more likely to yield consistent results when we follow a methodology regarding the systematics of structure. In information science and context of knowledge resources we can consider three major levels:

- Object entity structure (entities as part of an object).
- Object structure.
- Supra-object structure (e.g., complexity or inconsistency introduced by application or service scenario).

The methodological approach is beneficial when expressing disciplinary views and targeting purposes as it can

- help to create a consistent understanding of structure,
- address responsibility,
- help to assure logical and consistent development and management of structure, etc.

B. Structure and means of addressing

There are merely higher and lower facility levels of how structures can be addressed, which result from structure levels. For example, structure can be addressed by:

- Logic.
- Names.
- References.
- Address labels.
- Pointers.
- Fuzzy methods.
- Phonetic methods. ...

For example, 'non-structures' can be addressed by:

- Locality.
- Source.
- Context.
- Logic.
- Attributes.
- Size.
- Quantity. ...

Substantial differences of properties and facilities of different levels of structure and non-structure do have multi-fold origin and reason, especially:

- Structure is associated with different formalisation levels and respective consequences.
- Less complementary knowledge realisation, less potential, e.g., for logic.
- Less structure, less potential for approaches.
- Intrinsic and extrinsic properties are not interchangeable.
- Higher levels of structure mostly include tools usable for lower levels.
- Low level structures are limited to low level tools and soft criteria, e.g.,
  - Statistics.
  - Heuristics.
  - ...
- Potential from quality is different than from quantity.

With that background we should be aware that lower structure levels can only be addressed on higher formalisation levels, independent of the fact that structure may either be not available or not recognised. Substantial deficits of lower level structured data cannot be compensated by tools. In consequence, structure is and especially reflects:

- Knowledge (complements of factual, conceptual, procedural, metacognitive, structural, ...).
- Context.
- Experience.
- Persistence.
- Reusability.
- Sustainability.
- Value.
- Formalisation (including abstraction and reduction). ...

In result, it is structure that means features and facilities.

IV. Case: Structure-based realisation

The implementation strictly follows the fundamental methodological algorithm base, summarised in the following passages.

A. Methodological algorithm base and resulting method

The structure-based fusion methodology targets on supporting efficient problem solving, providing and accessing knowledge and knowledge object patterns. The methodology for creating structure-based fusion methods can be summarised on high level by:

- Pre-processing of structures.
- Option routines, for optional steps preparing fusion.
- Structure-based fusion of knowledge complements.
- Post-processing of structures.

This methodology is contrasting to plain procedure-based approaches. Here, structures are adapted for solving problems mostly allowing minimising procedure-based efforts and gaining higher efficiency and performance regarding creation and realisation of solutions. A method based on the structure-based fusion, implemented for knowledge mining context does consist of the basic steps:

- Pre-processing of knowledge mining structures.
- Options' routine, e.g., used for prioritisation and sorting. May contain procedures outside the range of balanced pre- and post-processing.
- Structure-based fusion of object structures, objects represented by knowledge complements.
- Post-processing of fusion result structures.

Besides the strictly structure-based fundament for the steps, realisations can use the whole gamut working for knowledge complements, e.g., comparisons, generic and standard component implementations, and knowledge complements and identification. Table I shows the method implementation and realisation of a structure-based fusion.

Table I. Method implementation and realisation: Steps of structure-based fusion.

Method Implementation Steps	*Realisation Example*
Outer pre-processing	individual, out of scope here
Input	**standard input, echo**
Inner pre-processing of structures	**perl**
Options' routines, prioritisation & sorting	**perl & sort**
Structure-based fusion	**join**
Inner post-processing of structures	**perl**
Output	**standard output**
Outer post-processing	individual, out of scope here

Any input and output can be intermediate, part of a complex mining process. Pre- and post-processing are handling the input and output for the options' routines and consecutive fusion, the central steps. For this implementation and demonstration processing is done via Perl, options' routines via Perl and sort, and the fusion via join.

The following structure based fusion code illustrations of realisations use inline GNU Bourne-Again SHell syntax, I/O and naming features for ease of demonstration and reproducibility.

B. Structure-based realisation: Simple case result

An instructive, simple case example implementation is a comparison and filter process of groups of arbitrary numbers of objects and object entities, which each can be of arbitrary volume and length. A procedural approach would create a procedure handling the structure and form of the entries as they appear and create and call a grep function for each of all the target entries or patterns. In advanced knowledge mining and processing, we often have to deal with sequences of steps creating intermediate results from previous results, all of which may need to be compared, sorted, filtered and so on. In that context the following scenario works as a basic example.

1) At a certain stage in a mining process we may have two groups of different knowledge object line entities.
2) We have to find only those various different string entries contained in one group and list those of the entries, which are also contained in the other group and produce combined object entities containing the content of respective entries from both groups.
3) We have to create an appropriate method and realisation, which, ideally, works for arbitrary numbers of objects with different sizes and content and which is flexible and knowledge-centric.

So, how can such 'search, comparison, filter, and sort' be realised for large numbers of objects, avoiding to call a routine or thread thousands or hundreds of thousand times per intermediate step and deploying 'structural information' instead? Figure 1 shows a structure-based solution. Its realisation is a self-contained regular shell script containing object groups and solution for ease of reconstructing the train of thought.

```
#
# Structure-based fusion sample -- (c) CPR, 2019, 2020
#
cnta="Natural Sciences collection entry 10:05:34 Volcano
Natural Sciences collection entry entities 10:05:35 Soufriere
Media attachment entry 10:06:34 Soufriere Photo
Addendum entry 10:05:30 References
Object entry compendium 10:06:37 comments"
cntb="Object entry 10:05:35 delivered
Excavation slide 10:05:34 updated
Object documentation update 10:06:34 request
System service no date
Object entity mining request 10:06:37 researcher id DF98_007
Object collection status 10:05:28 no resources reference
Object entity documentation request 10:05:30 user id database"
export cnta
export cntb
join -1 1 -2 1 <(echo "$cnta"|perl -pe 's/^(.*?)([0-9][0-9]:[0-9][0-9]:[0-9][0-9])(.*?)$/$2 1_BEFORE{$1} 1_AFTER{$3}/'|sort) <(echo "$cntb"|perl -pe 's/^(.*?)([0-9][0-9]:[0-9][0-9]:[0-9][0-9])(.*?)$/$2 2_BEFORE{$1} 2_AFTER{$3}/'|sort)
exit
```

Figure 1. Structure-based fusion solution, working with arbitrary one-lined object entities (excerpt), overcoming thousands of grep calls.

Here, two content groups of single line object instances are used for demonstration, content "a" and content "b". As can be seen, the content groups are asymmetric regarding object instances, content, and context aspects. For convenience of demonstration an excerpt of the contents is embedded in the code and represented by the exported variables named `cnta` and `cntb`. This excerpt is doing a selection of objects by fusion of arbitrary length and arbitrary number of objects by criteria (time stamps), which are reflected by structure.

The solution for that purpose achieves that result explicitly without the use of 'grep' (Global Regular Expression Parser), 'search', or comparable procedural routine instances. In order to create a straightforward solution and to easily follow the strategy, the steps are implemented using 5 external calls. Figure 2 shows the result of the realisation (Figure 1).

```
10:05:30 1_BEFORE{Addendum entry} 1_AFTER{ References} 2_BEFORE{Object entity documentation request} 2_AFTER{ user id database}
10:05:34 1_BEFORE{Natural Sciences collection entry} 1_AFTER{ Volcano} 2_BEFORE{Excavation slide} 2_AFTER{ updated}
10:05:35 1_BEFORE{Natural Sciences collection entry entities} 1_AFTER{ Soufriere } 2_BEFORE{Object entry} 2_AFTER{ delivered}
10:06:34 1_BEFORE{Media attachment entry} 1_AFTER{ Soufriere Photo} 2_BEFORE{Object documentation update} 2_AFTER{ request}
10:06:37 1_BEFORE{Object entry compendium} 1_AFTER{ comments} 2_BEFORE{Object entity mining request} 2_AFTER{ researcher id DF98_007}
```

Figure 2. Result of the structure-based approach, working with arbitrary one-lined object entities (excerpt).

The result reflects the target task based on structure-based fusion. The solution includes a sort of resulting object entities by respective string entries and appropriate marking of content from the groups for illustration.

C. Structure-based realisation: Multi-line case result

A different kind of complexity is what we commonly face in context of knowledge resources, same task and still with arbitrary length and arbitrary number of objects and entities with multi-line formatting (Figure 3) to be preserved.

```
Nisyros                 [Volcanology, Geology]:
                        Volcano, Type: Strato volcano, Island.
                        Status: Historical, Summit Elevation: 698\UD{m}. ...
                        VNUM: 0102-05=, ...,
                        Craters: ..., ...
                        %%IML: UDC:[550.3],[930.85],[911.2]
                        %%IML: media:...{UDC:[550.3+551.21],[911.2](4+38+23)}...jpg
                        Stefanos Crater, Nisyros, Greece.
                        LATLON: 36.578345,27.1680696
                        %%IML: GoogleMapsLocation: https://www.google.com/...@36.578345,27.1680696,337m/...
                        Little Polyvotis Crater, Nisyros, Greece.
                        LATLON: 36.5834105,27.1660736 ...
```

Figure 3. Knowledge resources' object ('Nisyros'): Multi-line formatting, conceptual knowledge, media object entities, and georeferences (excerpt).

Focus task is to find only those arbitrary object instances, which appear in one content context and also in another content context and to combine the data of those instances in a result object instance. It is preferable if the realisation allows a multi-object fusion, meaning more than one object in a process. A common procedure realisation would, e.g., have to call a 'grep' function (especially a Global Regular Expression Parser) for every of the thousands of object instances in one context for searches in another context

Figure 4 shows a more efficient, structure-based realisation for such objects. As with the previous example realisation above, its realisation is presented as a self-contained regular shell script for ease of demonstration. The excerpt fully confirms with a standard shell and Perl syntax and features and is compact. In order to create a straightforward solution and to easily follow the strategy, the steps are implemented using 10 external calls, which could still be further reduced. As can be seen, these calls already include formatting cleanup with pre- and post-processing, too. The solution targets contexts for larger numbers of multi-line, multi-entity object instances (thousands or hundreds of thousands). As common, results should be considered intermediate for complex knowledge mining procedures.

```
#
# Structure-based fusion mining -- (c) CPR, 2019, 2020
#
cnta="Nirgal                [Etymology]:
                        ...
Nisyros                 [Archaeology, Geology, Volcanology]:
                        Island, Volcano, Greece, Dodecanese Islands. ...
                        The island provides unique archaeological remains, esp. ...
                        History and mythology of the island and volcano are ...
                        %%IML: UDC:[902],[930.85],[911.2]"63"(4+38+23+24)=14
Nisyros_archive         [Archaeology]:
                        Media, Archaeology Digital Object Archive (ADOA). ...
NMR                     [Archaeology]:
                        ..."
cntb="Niggli                [Petrography, Mineralogy]:
                        ...
Nisyros                 [Volcanology, Geology]:
                        Volcano, Type: Strato volcano, Island, Greece.
                        Status: Historical, Summit Elevation: 698\UD{m}. ...
                        VNUM: 0102-05=, ..., Craters: ..., ...
                        %%IML: UDC:[550.3],[930.85],[911.2]
                        %%IML: media:...{UDC:[550.3+551.21],[911.2](4+38+23)}...jpg
                        Stefanos Crater, Nisyros, Greece.
                        LATLON: 36.578345,27.1680696
                        %%IML: GoogleMapsLocation: https://www.google.com/...@36
 .578345,27.1680696,337m/...
                        Little Polyvotis Crater, Nisyros, Greece.
                        LATLON: 36.5834105,27.1660736 ...
Nisyros_archive         [Volcanology]:
                        Media, Geosciences Digital Object Archive (GDOA). ...
NLBR                    [Platetectonics, Volcanology]:
                        ...
N-MORB                  [Platetectonics, Volcanology]:
                        ..."
export cnta
export cntb
join -1 1 -2 1 -t" " \
  <(echo "$cnta"|\
    perl -pe 's/^(\S)/TMPBOL$1/;s/$/TMPEOL/;s/\n//;'|perl -pe 's/TMPBOL//;s/
 TMPBOL/\n/g'|\
    perl -pe 's/^(.*?)(\[.*?\]:)(.*)$/$1$2$3/'|sort -k 1b,1)\
  <(echo "$cntb"| \
    perl -pe 's/^(\S)/TMPBOL$1/;s/$/TMPEOL/;s/\n//;'|perl -pe 's/TMPBOL//;s/
 TMPBOL/\n/g'|\
    perl -pe 's/^(.*?)(\[.*?\]:)(.*)$/$1$2$3/'|sort -k 1b,1)|\
  perl -pe 's/TMPEOL$//g;s/TMPEOL/\n/g;'
exit
```

Figure 4. Structure-based fusion solution, working with arbitrary multi-line knowledge resources' object entities (excerpt), overcoming grep.

With the above scenario the realisation should be fully logical and self explanatory. The easiest way to comprehend the implementation principle is to follow the sequence of instructions and reproduce step by step, directly to the result. The realised solution should be reasonably flexible and robust. Figure 5 shows the result of the realisation (Figure 4).

```
Nisyros                 [Archaeology, Geology, Volcanology]:
                        Island, Volcano, Greece, Dodecanese Islands. ...
                        The island provides unique archaeological remains, esp. ...
                        History and mythology of the island and volcano are ...
                        %%IML: UDC:[902],[930.85],[911.2]63(4+38+23+24)=14
                [Volcanology, Geology]:
                        Volcano, Type: Strato volcano, Island, Greece.
                        Status: Historical, Summit Elevation: 698\UD{m}. ...
                        VNUM: 0102-05=, ..., Craters: ..., ...
                        %%IML: UDC:[550.3],[930.85],[911.2]
                        %%IML: media:...{UDC:[550.3+551.21],[911.2](4+38+23)}...jpg
                        Stefanos Crater, Nisyros, Greece.
                        LATLON: 36.578345,27.1680696
                        %%IML: GoogleMapsLocation: https://www.google.com/...@36
                         .578345,27.1680696,337m/...
                        Little Polyvotis Crater, Nisyros, Greece.
                        LATLON: 36.5834105,27.1660736 ...
Nisyros_archive         [Archaeology]:
                        Media, Archaeology Digital Object Archive (ADOA). ...
      [Volcanology]:
                        Media, Geosciences Digital Object Archive (GDOA). ...
```

Figure 5. Result of the structure-based approach, working with arbitrary multi-line object entities, result shows the multi-object fusion (excerpt).

Objects (`Nisyros` and `Nisyros_archive`) have correctly been identified (criteria name string) and unified. Entities have been preserved and conceptual knowledge of instances have been combined in a unique object instance each.

Even the indentation of the resulting content reflects the operations and is exactly preserved. Besides valuable knowledge content, conceptual knowledge integration is considered a marker regarding the range of knowledge and efficiency of a solution when working with knowledge resources. In this case the knowledge resources use UDC references with millions of object instances and entities. Universally consistent conceptual knowledge, e.g., used for CKPM, is based on UDC references for demonstration, spanning the main tables [24] shown in Table II.

TABLE II. CKRI: IMPLEMENTED UDC CODE REFERENCES, MAIN TABLE (E.0.4.2).

Code / Sign Ref.	*Verbal Description (EN)*
UDC:0	Science and Knowledge. Organization. Computer Science. Information. Documentation. Librarianship. Institutions. Publications
UDC:1	Philosophy. Psychology
UDC:2	Religion. Theology
UDC:3	Social Sciences
UDC:5	Mathematics. Natural Sciences
UDC:6	Applied Sciences. Medicine, Technology
UDC:7	The Arts. Entertainment. Sport
UDC:8	Linguistics. Literature
UDC:9	Geography. Biography. History

Natural sciences related conceptual knowledge pattern entities are created based on UDC references [25] of mathematics and natural sciences. An excerpt of the implementation is shown in Table III.

TABLE III. CKRI: IMPLEMENTED UDC CODE REFERENCES OF MATHEMATICS AND NATURAL SCIENCES (EXCERPT, E.0.4.2).

Code / Sign Ref.	*Verbal Description (EN)*
UDC:51	Mathematics
UDC:53	Physics
UDC:535	Optics
UDC:535.6	Colours and their properties. Colour theory
UDC:539	Physical nature of matter
UDC:54	Chemistry. Crystallography. Mineralogy
UDC:55	Earth Sciences. Geological sciences
UDC:550.3	Geophysics
UDC:551	General geology. Meteorology. Climatology.
UDC:551.21	Vulcanicity. Vulcanism. Volcanoes. Eruptive phenomena. Eruptions
UDC:551.24	Geotectonics
UDC:551.4	Geomorphology. Study of the Earth's physical forms
UDC:551.44	Speleology. Caves. Fissures. Underground waters
UDC:551.46	Physical oceanography. Submarine topography. Ocean floor
UDC:551.7	Historical geology. Stratigraphy. Palaeogeography
UDC:551.8	Palaeogeography
UDC:56	Palaeontology
UDC:57	Biological sciences in general
UDC:58	Botany
UDC:59	Zoology

Conceptual knowledge pattern entities of archaeology and history are created based on UDC references [26] of geography, biography, history (Table IV).

TABLE IV. CKRI: IMPLEMENTED UDC CODE REFERENCES OF GEOGRAPHY, BIOGRAPHY, HISTORY (EXCERPT, E.0.4.2).

Code / Sign Ref.	*Verbal Description (EN)*
UDC:902	Archaeology
UDC:903	Prehistory. Prehistoric remains, artefacts, antiquities
UDC:904	Cultural remains of historical times
UDC:908	Area studies. Study of a locality
UDC:91	Geography. Exploration of the Earth and of individual countries. Travel. Regional geography
UDC:912	Nonliterary, nontextual representations of a region
UDC:92	Biographical studies. Genealogy. Heraldry. Flags
UDC:93/94	History
UDC:94	General history

The geoscientific / prehistory / archaeology integration from the case studies and implementations for geoscientific information systems and application components is used for illustration in the next sections. The example will show a tiny subset of the comprehensive, universal conceptual knowledge used, integrating UDC:902/908 (Archaeology. Prehistory. Cultural remains. Area studies) and UDC:55 (Earth Sciences. Geological sciences) and humanities (UDC main table trees).

D. Multi-line knowledge ranges and computation

As commonly we have to handle many objects, we can illustrate how efficiency and performance scale with numbers of objects. The examples use the above multi-line knowledge case, as knowledge resources' objects regulary have a high variety of content, with different object volumes and lengths. Therefore, this is more for practical experience than a benchmark. The overall number of object instances in the respective primary knowledge ranges for the resources' excerpt is shown in the UDC references' test environment (Table V).

TABLE V. PRIMARY KNOWLEDGE RANGES OF CONCEPTUAL KNOWLEDGE ENTITY REFERENCES IN THE TEST ENVIRONMENT (EXCERPT).

Knowledge Range	*Entities' Count*
UDC:9 (incl. UDC:902/904)	930,000
UDC:5 (incl. UDC:55/56)	1,700,000

The ranges can be comprehended in all details by following the publicly available online conceptual knowledge framework already discussed above. Table VI shows an implementation excerpt and computational footprint for the different approaches. The different case results were achieved on Intel® Xeon® CPU X5570 (2.933 GHz) systems under Linux.

TABLE VI. COMPUTATIONAL FOOTPRINT OF PROCEDURE-/STRUCTURE-BASED SOLUTIONS, CONCEPTUAL KNOWLEDGE REF. CASES (EXCERPT).

Knowledge Range	*Entities' Count*	*Context Calls' Count and Wall Time*			
		Procedure-based		*Structure-based*	
UDC:902	48,000	$\gg$48,000	2,440 s	10	32 s
UDC:55	54,000	$\gg$54,000	3,938 s	10	45 s
UDC:902/904	107,000	$\gg$107,000	24,775 s	10	198 s
UDC:55/56	295,000	$\gg$295,000	189,100 s	10	945 s

The values allow to rate the discussed conventional approach (max. 1,000 loosely parallel pattern matching calls 'practical') using a procedure-based solution and the structure-based approach. The two examples of the approaches to challenging mining cases are using the same range of knowledge/data content each, specified by ranges of referenced conceptual knowledge. Requirements for the consideration of wider knowledge ranges do show a major impact on the procedure-based solution, resulting in relatively larger increase of context calls and wall times. Even if more loosely parallel calls would be logically possible with a mining algorithm it is not practical to increase their number on the same machine with procedure-based solutions. The counts of object entries in the two content resources are of major impact for the efficiency differences. The context calls' count (10) for structure-based fusion is based on the above presented multi-line object solution and can be kept stable. The result of the comparison of the computational footprint is clearly in favor of the structure-based solution. This tendency even improves with increasing numbers of objects involved.

V. CASE: SPATIAL CONTEXT COMPUTATION

Contextualisation, besides prehistoric knowledge computation itself, often requires multi-disciplinary component integration, especially advanced spatial context computation, e.g., integration of Digital Elevation Models (DEM), processing, and analysis. The case of structure-based realisation demonstrated the efficiency of structure-based approaches. The structure-based scenario of KR, e.g., employing PAKA, can further be natively integrated with the realisations of cognostic addressing of structure and structural knowledge ('nucleal cognstructure') [27], for contextualisation supported by standardised components, especially efficient and effective spatial context computation and processing, as practically implemented and realised for the methodological structure-based approach and illustrated in exactly the above examples (Figures 3 and 5).

Efficient computation and processing are relevant for many contextualisation tasks, e.g., with chorological contexts in prehistory and archaeology regarding

- height reference computation,
- resampling,
- illumination,
- spatial parametrisation,
- spatial computation, and
- symbolic representation.

The following components in the workflow enable a per instance and per core parallelisation, e.g., chronological and/or chorological slices and other conceptual knowledge. This allows to create the required number of time slices, especially series of images and other symbolic representation.

Each context may require different parametrisation with respect to the state of the art of research, e.g., prehistorical knowledge resources and chronological and chorological object discretisation and quantification on the one hand, and geoscientific parametrisation for a relevant time interval on the other hand.

A. Prehistory context relevant CKRI entity groups

The following passages excerpt relevant entity groups of the prehistory-protohistory and archaeology Conceptual Knowledge Reference Implementation (CKRI), including multi-disciplinary contexts of natural sciences and humanities (E.0.4.2) [28] [23]. Geodesy related conceptual knowledge pattern entities are created based on UDC references [25], part of the astronomy, astrophysics, space research, and geodesy sections. An excerpt of the implementation is shown in Table VII.

TABLE VII. CKRI: IMPLEMENTED UDC CODE REFERENCES OF GEODESY (EXCERPT, E.0.4.2).

Code / Sign Ref.	*Verbal Description (EN)*
UDC:52	Astronomy. Astrophysics. Space research. Geodesy
UDC:528	Geodesy. Surveying. Photogrammetry. Remote sensing. Cartography
UDC:528.2	Figure of the Earth. Earth measurement. Mathematical geodesy. Physical geodesy. Astronomical geodesy
UDC:528.3	Geodetic surveying
UDC:528.4	Field surveying. Land surveying. Cadastral survey. Topography. Engineering survey. Special fields of surveying
UDC:528.5	Geodetic instruments and equipment
UDC:528.7	Photogrammetry: aerial, terrestrial
UDC:528.8	Remote sensing
UDC:528.9	Cartography. Mapping (textual documents)

An excerpt of major references from the auxiliaries of time [29] is shown in Table VIII.

TABLE VIII. CKRI: IMPLEMENTED UDC CODE REFERENCES, AUXILIARIES OF TIME (EXCERPT, E.0.4.2).

Code / Sign Ref.	*Verbal Description (EN)*
UDC:"0"	First millennium CE
UDC:"1"	Second millennium CE
UDC:"2"	Third millennium CE
UDC:"3/7"	Time divisions other than dates in Christian (Gregorian) reckoning
UDC:"3"	Conventional time divisions and subdivisions: numbered, named, etc.
UDC:"32"	The year. Seasons and other divisions of the year
UDC:"321/324"	Seasons
UDC:"321"	Spring
UDC:"322"	Summer
UDC:"323"	Autumn (fall)
UDC:"324"	Winter
UDC:"325"	Quarters (quarter years, trimesters)
UDC:"327"	Months
UDC:"328"	Weeks
UDC:"329"	Days
UDC:"34"	Day and night phenomena. Hours or times of day
UDC:"344"	Daytime. Daylight hours
UDC:"345"	Night-time. Hours of darkness or semi-darkness
UDC:"36"	Times of peace, war, danger, emergency, difficulties
UDC:"362"	Peacetime. Time of no danger
UDC:"363"	Time of danger, threat
UDC:"364"	Wartime
UDC:"367"	Times according to volume of use, load, demand
UDC:"37"	Time of work activity, occupation, production, daily routine
UDC:"372"	Working hours. Service hours. Time of occupation
UDC:"377"	Rest and recreation time. Spare time. Free time. Time outside working hours
UDC:"38"	Holidays. Festive and commemorative occasions
UDC:"382"	Religious holidays, festive and commemorative occasions
UDC:"383"	Public, national or regional holidays (other than religious)
UDC:"385"	Personal private holidays, vacation or leave time
UDC:"4"	Duration. Time-span. Period. Term. Ages and age-groups
UDC:"5"	Periodicity. Frequency. Recurrence at specified intervals.
UDC:"6"	Geological, archaeological and cultural time divisions
UDC:"61/62"	Geological (lithological / biological / palaeoecological) time division
UDC:"61"	Precambrian to Mesozoic (from more than 600 to 70 MYBP)
UDC:"62"	Cenozoic (Cainozoic). Neozoic (70 MYBP - present)
UDC:"63"	Archaeological, prehistoric, protohistoric periods and ages
UDC:"67/69"	Time reckonings: universal, secular, non-Christian religious
UDC:"67"	Universal time reckoning. Before Present
UDC:"68"	Secular time reckonings other than universal and the Christian (Gregorian) calendar
UDC:"69"	Dates and time units in non-Christian (non-Gregorian) religious time reckonings
UDC:"7"	Phenomena in time. Phenomenology of time

The auxiliaries of time can be used for creating arbitrary facets, including arbitrary time concepts. Facets can be used to integrate addressing different concepts, e.g., absolute and relative time concepts used in prehistoric, archaeological, and natural sciences contexts.

Table IX shows an excerpt of the implementation. Auxiliaries of spatial features/place (UDC (1/9) [30] Place and space in general. Localization. Orientation

TABLE IX. CKRI: IMPLEMENTED UDC CODE REFERENCES OF SPATIAL FEATURES / PLACE: AUXILIARIES OF PLACE, BOUNDARIES AND SPATIAL FORMS (EXCERPT, E.0.4.2).

Code / Sign Ref.	*Verbal Description (EN)*
UDC:(1)	Place and space in general. Localization. Orientation
UDC:(100)	Universal as to place. International. All countries in general
UDC:(2)	Physiographic designation
UDC:(20)	Ecosphere
UDC:(21)	Surface of the Earth in general. Land areas in particular. Natural zones and regions
UDC:(23)	Above sea level. Surface relief. Above ground generally. Mountains
UDC:(24)	Below sea level. Underground. Subterranean
UDC:(25)	Natural flat ground (at, above or below sea level). The ground in its natural condition, cultivated or inhabited
UDC:(26)	Oceans, seas and interconnections
UDC:(28)	Inland waters
UDC:(3/9)	Individual places of the ancient and modern world
UDC:(3)	Places of the ancient and mediaeval world
UDC:(31)	Ancient China and Japan
UDC:(32)	Ancient Egypt
UDC:(33)	Ancient Roman Province of Judaea. The Holy Land. Region of the Israelites
UDC:(34)	Ancient India
UDC:(35)	Medo-Persia
UDC:(36)	Regions of the so-called barbarians
UDC:(37)	Italia. Ancient Rome and Italy
UDC:(38)	Ancient Greece
UDC:(399)	Other regions. Ancient geographical divisions other than those of classical antiquity
UDC:(4/9)	Countries and places of the modern world
UDC:(4)	Europe
UDC:(5)	Asia
UDC:(6)	Africa
UDC:(7/8)	America, North and South. The Americas
UDC:(7)	North and Central America
UDC:(8)	South America
UDC:(9)	States and regions of the South Pacific and Australia. Arctic. Antarctic

These entities can be used for arbitrary facets, including orientation and relative position.

B. *Prehistory context multi-disciplinary views*

Table X shows an excerpt of UDC:903.. . .:2 ritual/burial object and subgroup examples, and conceptual view groups [12] for prehistory and protohistory (PAKA, [20] [21]).

TABLE X. PREHISTORY AND PROTOHISTORY RITUAL/BURIAL OBJECT AND SUBGROUP EXAMPLES, AND CONCEPTUAL VIEW GROUPS [12] (EXCERPT).

Major Object Group	*Selected Objects*	*Conceptual View Group*
Ritual places, burials	yes	UDC:903.. . .:2
Cemetery	–	UDC:903.. . .:2
Barrow	–	UDC:903.. . .:2
round	–	UDC:903.. . .:2
long	–	UDC:903.. . .:2
Cist	–	UDC:903.. . .:2
Dolmen	–	UDC:903.. . .:2
Tomb	–	UDC:903.. . .:2
chamber	–	UDC:903.. . .:2
court	–	UDC:903.. . .:2
portal	–	UDC:903.. . .:2
rock cut	–	UDC:903.. . .:2
wedge	–	UDC:903.. . .:2
Pithos burial	–	UDC:903.. . .:2
Cave	–	UDC:903.. . .:2
Body finding	–	UDC:903.. . .:2
Urn	–	UDC:903.. . .:2
. . .	–	UDC:903.. . .:2

For this illustrative object scenario, the excerpt does not show individual micro-groups. Besides different distributions and different origins, object context can be referred, e.g., artificial origin and natural origins as well as relevant object properties, materials, and soil contexts can be considered. The components can be integrated via coherent conceptual knowledge.

For illustration of this case, we want to discover the contexts of a possible range of comparable tombs in a target area with those for the suggested type and/or time of the Odyssey in the respective region. Here, the target is the distribution of groups of cist tombs in North-Rhine Westphalia and Lower Saxony, Germany and The Netherlands in context with the distribution of groups of cist tombs in the time of king Odysseus in the Homeric Ithaca, Kefalonia (Cephalonia), Paliki region. The result configuration should provide homogeneous context topography and context soil references and contain the respective groups in the regions and the distribution of caves in the respective countries.

C. Resulting coherent knowledge context integration

Enabled by coherent conceptual knowledge contexts, cist tombs can be selected from PAKA (Table X) and prehistoric caves in borders of today's North-Rhine Westphalia and Lower Saxony, Germany, The Netherlands, Greece.

Respective conceptual and symbolic representation criteria, e.g., spatial criteria can be deployed the integrating knowledge complements from available KR.

The available multi-disciplinary context components are shown. The geodesic distance calculations based on the ellipsoidal parameters of Earth's sphere select cist tombs in association of the Kefalonia acropolis and sites like Tzanata. In this case, considered cists are "Early Geometric cist tombs".

Each conceptual space-time slice requires a number of contributions, especially:

- Resampling,
- spatial computation (DEM),
- intensity computation,
- parametrisation,
- colourisation palette creation,
- spatial selection,
- spatial computation (vector),
- Point on Interest (POI), line, and polygon criteria selection.
- soil context computation,
- KR computation,
- projection, and
- computation of symbolic representation.

Figure 1 shows a generated, resulting coherent conceptual knowledge integration sketch for the realisation based on the KR. Selection criteria can be complex, a decent illustrative example employing a workflows of spatial criteria, e.g.,

- polygon criteria, e.g., in North-Rhine Westphalia and Lower Saxony, Germany and The Netherlands,
- spherical distance criteria, e.g., 100 km. For such criteria, the calculation of distances on planetary bodies like Earth depends on the ellipsoidal parameters of the body and the respective method of computation. Here, required geodesic distance calculations are based on the ellipsoidal parameters of Earth's sphere [31].

Context components, workflows, and procedures can be standardised. Nevertheless, for considering any scientific task and associated questions, a solid understanding of all the algorithms and the consequences of integration should be mandatory, especially regarding resulting analysis and interpretation.

The sketch considers the major conceptual references for illustration. Detailed research can further detail on prehistoric object groups, characteristics, and properties, topographic properties, soil properties, and many more. Therefore, the conceptual sketch view can result in levels of arbitrary numbers of different integrations of complements and associated properties as resulting from the KR, which are discussed in the following. The result integrates required KR components based on coherent conceptual knowledge and systematical chorological knowledge for multi-disciplinary contexts, e.g., arbitrary group representations, classification based representations, and geospatial representations.

Knowledge objects and contexts are provided by The Prehistory and Archaeology Knowledge Archive (PAKA) [20] [21]. The multi-disciplinary coherent contextualisation employs the base of a new soil system reference development with soil types (UDC:631.4. . .) of the World Reference Base (WRB) standard, reference contexts, especially for UDC:903.. . .:2,551.7+"628".. . ., prehistorical, protohistorical time spans and respective objects and artefacts for this case, related to religion and rituals, geology, especially stratigraphy and paleogeography, quaternary, especially late glacial and Holocene.

The integrated natural sciences KR further provide information on caves in the respective region. Contextualisation is enabled by the Conceptual Knowledge Reference Implementation (CKRI), including multi-disciplinary contexts of natural sciences and humanities (E.0.4.2) [23]. The conceptual knowledge base is The Universal Decimal Classification (UDC) [12].

Associated information, e.g., on soil specifications and properties, can be found as reference in the WRB for soil resources [32], [33] from the Food and Agriculture Organisation (FAO), United Nations.

In this illustration plain Digital Chart of the World (DCW) data are used [34]. The coastline database is the Global Self-consistent Hierarchical High-resolution Geography (GSHHG) [35] [36], which was mainly compiled from the World Vector Shorelines (WVS) [37], the CIA World Data Bank II (WDBII) [38], and the Atlas of the Cryosphere (AC).

An equal area projection (Eckert IV) is advised due to the type of discipline knowledge representation. The compilation uses the World Geodetic System (WGS). The symbolic representation of the contextualisation is done via LX Professional Scientific Content-Context-Suite (LX PSCC Suite) deploying the Generic Mapping Tools (GMT) [39] for visualisation.

D. Prehistory context multi-disciplinary integration facets

Table XI shows the reference facets of a respective multi-disciplinary target contextualisation.

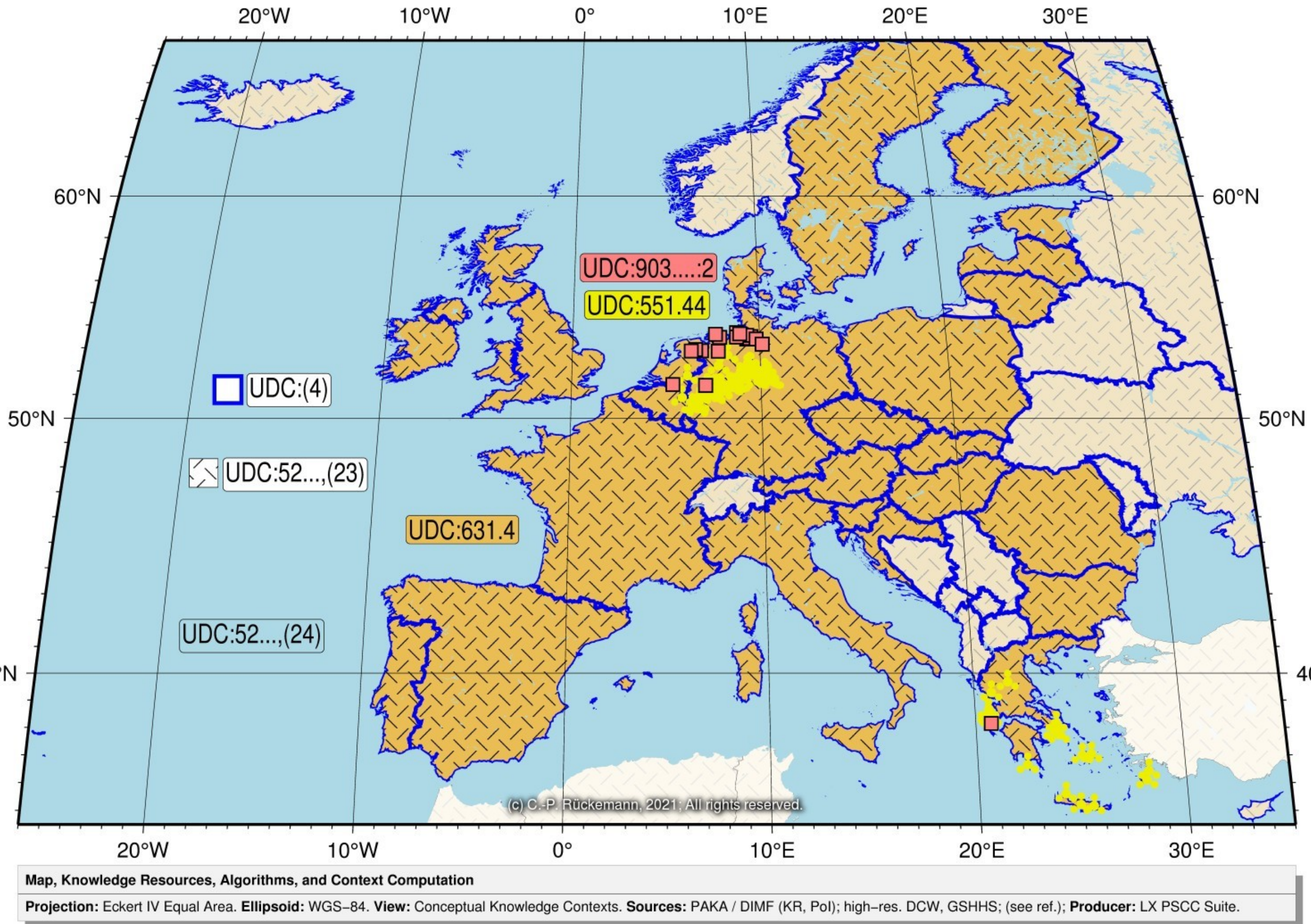


Figure 1. Resulting coherent conceptual knowledge integration sketch diagram showing knowledge resources for a prehistoric, natural sciences, and spatial contextualisation for excerpts of prehistoric cist tomb distributions, remote sensing data, and soil properties with respective knowledge references.

TABLE XI. REFERENCE FACETS OF A MULTI-DISCIPLINARY TARGET CONTEXTUALISATION OF PREHISTORY CONTEXT, BASED ON CKRI (E.0.4.2), IMPLEMENTED USING UDC CODE REFERENCES (EXCERPT).

Code / Sign Ref.	*Verbal Description (EN)*
	Geography. Biography. History
UDC:903.. . .	Prehistory, prehistoric remains, artefacts, antiquities
. . .:2. . .CIST	referring to religion and rituals . . . cist tomb
. . .,“62. . .”	from Holocene
. . .,(4. . .DENW)	. . . in North-Rhine Westphalia, Germany
. . .,(4. . .DENI)	. . . in Lower Saxony, Germany
. . .,(4. . .NL)	. . . in The Netherlands
. . .,(38):	Ancient Greece
. . .,(495)	Greece. Hellenic Republic. Elliniki Dimokratia
	Earth sciences, geological sciences
UDC:551.44	Speleology, caves, fissures, underground waters
	Applied sciences, agriculture in general
UDC:631.4	Soil research data
	Geodesy. Photogrammetry
UDC:52. . .,(23)	Remote sensing data, above sea level
UDC:52. . .,(24)	Remote sensing data, below sea level
	Contextualisation Place
UDC:(4)	Europe

The contextualisation uses coherent conceptual knowledge and refers to the chorological references for consequent knowledge integration and symbolic representation. Respecting the principles of formalisation, methodology, research focus, and the integrated context components, this implementation result allows numerous individual solution facilities for further investigation.

E. *Components' spatial contexts and computation*

This case can employ the structure-based fusion for the respective KR computation and processing. The integrated spatial components' computation and processing can deploy parallelisation, e.g., for creation of slices.

Table XII shows an implementation excerpt and computational footprint for the components in the given case. For comparison to the above values, the results are given for an Intel® Xeon® CPU X5570 (2.933 GHz) systems under Linux.

TABLE XII. COMPUTATIONAL FOOTPRINT OF PROCESSING AND COMPUTATION FOR COMPONENTS WITH PARALLEL SUPPORT (EXCERPT).

Contributing Components	*Description / Resources*	*Wall Time*
Preprocessing	DEM, KR, PAKA	280 s
Slice (parallelised instance)		52 s
UDC:52.. . .	DEM, raster, vector, KR	38 s
UDC:903.. . .:. . .	Entities, KR, PAKA	5 s
UDC:551.44	Entities, KR	3 s
UDC:631.4	Entities, soil context, KR	6 s
Postprocessing	[Symbolic representation]	350 s

The number of slices can range from one to n, commonly many thousands for standard applications. The respective overall wall clock times are about n-times of parallelised instances for slices and benefit from the parallelisation, e.g., via OpenMP [40], [41], [42]. In this case the wall times for even small numbers of slices, e.g., 360 slices result in $360{\cdot}52\,\text{s}{=}18{,}720\,\text{s}$ versus 52 s for parallel instances. Therefore, pre- and postprocessing do not contribute significantly in demanding cases.

The contributing components allow for prehistory and archaeology to consider the knowledge processing, parametrisation, and the required chorological contexts, e.g., height reference computation, resampling, illumination, spatial parametrisation, spatial computation, and symbolic representation during preprocessing, spatial context integration, and postprocessing in an efficient and scalable way.

VI. Conclusion

It is generally preferable to integrate multi-disciplinary knowledge with the expertise of the respective participated disciplines. This research has shown to allow a coherent integration of multi-disciplinary knowledge with the respective backgrounds.

This paper presented two successful cases of prehistory-protohistory contextualisation, structure-based fusion realisation (prehistory-archaeology knowledge resources and volcanology contexts) and integrated spatial context computation (cist tombs, speleology, spatial, and soil contexts).

This research achieved the goal to create performant methods for efficient problem solving deploying the new structure-based fusion methodology and spatial context computation. Structure-based fusion can provide a valuable, scalable option alternative to procedure-based approaches. The presented case realisation successfully considered conceptual knowledge, especially the core component of UDC references, which is most important in context of handling advanced structures for universal, multi-disciplinary, and multi-lingual knowledge for many objects. The case implementations illustrated that even complex scenarios with computational challenges and large numbers of involved objects can be efficiently created and realised. Structure-based methods increase the means to address structure and to beneficially use structural knowledge and information, which are otherwise not easily deployable by procedure-based approaches. The solutions showed the flexibility of knowledge- and data-centricity. The implementations of the methods proved being able to minimise the number of calls and threads. The methodology and efficiency in creating and adapting implementations that way can have significant impact on sustainability and consistency of long-term solutions.

The structure-based fusion solutions not just provide facilities for fast, resource efficient operation, even if not optimised as the shown realisations. They are modular, long-term sustainable, and widely programming / language implementation independent. Realisations can be easily adapted to different environments (programming languages / shells and operating systems). For the research group and partners the solutions proved adaptability and efficiency in many practical realisation, for years, new and rewritten, in context of resources development and knowledge mining and many solutions beyond.

The coherent conceptual knowledge integration of components also showed the vast potential for understanding multi-disciplinary contexts and processes – not only for prehistory, archaeology, natural sciences, and humanities.

Future research will continue creating structure-based fusion solutions for knowledge mining and day-to-day challenges. Further research will be done on multi-disciplinary and spatial context computation for enabling an ongoing analysis refinement with the continuous development of integrated components contributed by many disciplines, especially taking new findings and context integration in prehistory and archaeology into account.

Acknowledgements

We are grateful to the "Knowledge in Motion" (KiM) long-term project, Unabhängiges Deutsches Institut für Multi-disziplinäre Forschung (DIMF), for partially funding this research, implementation, case studies, and publication under grants D2012F1P04497, D2014F2P04562, D2018F1P04962 and D2020F1P05228 and to its senior scientific members and members of the permanent commission of the science council, especially to Dr. Friedrich Hülsmann, Gottfried Wilhelm Leibniz Bibliothek (GWLB) Hannover, to Dipl.-Biol. Birgit Gersbeck-Schierholz, Leibniz Universität Hannover, to Dipl.-Ing. Martin Hofmeister, Hannover, and to Olaf Lau, Hannover, Germany, for fruitful discussion, inspiration, practical multi-disciplinary case studies, and the analysis of advanced concepts. We are grateful to Dipl.-Geogr. Burkhard Hentzschel and Dipl.-Ing. Eckhard Dunkhorst, Minden, Germany, for prolific discussion and exchange of practical spatial and drone case scenarios. We are grateful to Dipl.-Ing. Hans-Günther Müller, Göttingen, Germany, for providing specialised, manufactured high end computation and storage solutions. We are grateful to the members of the Eastern Mediterranean research and studies campaign 2018–2020, DIMF, and all national and international partners in the Geo Exploration and Information cooperations for their constructive and trans-disciplinary support. We are grateful to The Science and High Performance Supercomputing Centre (SHPSC) for long-term support. / DIMF-PIID-DF98_007; URL: https://scienceparagon.de/cpr .

References

[1] C.-P. Rückemann, "The Impact of Information Science Accompanied Structural Information on Computation of Knowledge Pattern Matching and Processing: A Prehistory, Archaeology, Natural Sciences, and Humanities Conceptual Integration Perspective," in Proceedings of The Tenth International Conference on Advanced Communications and Computation (INFOCOMP 2020), September 27 - October 1, 2020, Lisbon, Portugal. XPS Press, Wilmington, Delaware, USA, 2020, ISSN: 2308-3484, ISBN: 978-1-61208-807-5.

[2] L. W. Anderson and D. R. Krathwohl, Eds., A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Allyn & Bacon, Boston, MA (Pearson Education Group), USA, 2001, ISBN: 978-0801319037.

[3] Aristotle, Nicomachean Ethics, Volume 1, 2009, Project Gutenberg, eBook, EBook-No.: 28626, Release Date: April 27, 2009, Digitised Version of the Original Publication, Produced by Sophia Canoni, Book provided by Iason Konstantinidis, Translator: Kyriakos Zambas, URL: http://www.gutenberg.org/ebooks/12699 [accessed: 2021-11-27].

[4] Aristotle, The Ethics of Aristotle, 2005, Project Gutenberg, eBook, EBook-No.: 8438, Rel. Date: Jul., 2005, Digit. Vers. of the Orig. Publ., Produced by Ted Garvin, David Widger, and the DP Team, Edition 10, URL: http://www.gutenberg.org/ebooks/8438 [accessed: 2021-11-27].

[5] C.-P. Rückemann, F. Hülsmann, B. Gersbeck-Schierholz, P. Skurowski, and M. Staniszewski, Knowledge and Computing. Post-Summit Results, Delegates' Summit: Best Practice and Definitions of Knowledge and Computing, Sept. 23, 2015, The Fifth Symp. on Adv. Comp. and Inf. in Natural and Applied Sciences (SACINAS), The 13th Int. Conf. of Num. Analysis and Appl. Math. (ICNAAM), Sept. 23–29, 2015, Rhodes, Greece, 2015, pp. 1–7, DOI: 10.15488/3409.

[6] Plato, Phaedo, 2008, (Written 360 B.C.E.), Translated by Benjamin Jowett, Provided by The Internet Classics Archive, URL: http://classics.mit.edu/Plato/phaedo.html [accessed: 2021-11-27].

[7] M. Foucault, The Archaeology of Knowledge. Routledge Classics, 2002, ISBN: 978-0-415-28752-4, Translated by A. M. Sheridan Smith.

[8] F. de Saussure, Cours de linguistique générale, 1916, (title in English: Course in General Linguistics), C. Bally and A. Sechehaye (eds.).

[9] Aristotle, On Interpretation, 2008, Peri Hermeneias, Organon, (Written 350 B.C.E.), Translated by E. M. Edghill Provided by The Internet Classics Archive, URL: http://classics.mit.edu/Aristotle/interpretation.html [accessed: 2021-11-27].

[10] C. K. Ogden and I. A. Richards, The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism, 1923.

[11] R. Gleser, Rekonstruktion der Vergangenheit: Zur methodischen Eigenart prähistorischen Erkennens. mentis, Paderborn, 2018, pp. 199–237, (title in English: Reconstruction of the Past: On the Methodical Peculiarity of Prehistorical Cognition), in: A.-S. Naujoks, J. Stelling, and O. R. Scholz (eds.), Von der Quelle zur Theorie. Vom Verhältnis zwischen Objektivität und Subjektivität in den historischen Wissenschaften, ISBN: 978-3-95743-136-3.

[12] "Multilingual Universal Decimal Classification Summary," 2012, UDC Consortium, 2012, Web resource, v. 1.1. The Hague: UDC Consortium (UDCC Publication No. 088), URL: http://www.udcc.org/udcsummary/php/index.php [accessed: 2021-11-27].

[13] "Creative Commons Attribution Share Alike 3.0 license," 2012, URL: http://creativecommons.org/licenses/by-sa/3.0/ [accessed: 2021-11-27], (first release 2009, subsequent update 2012).

[14] C.-P. Rückemann, "Principles of Superordinate Knowledge: Separation of Methodology, Implementation, and Realisation," in The Eighth Symp. on Adv. Comp. and Inf. in Natural and Applied Sciences (SACINAS), Proceedings of The 16th Int. Conf. of Num. Analysis and Appl. Math. (ICNAAM), Sept. 13–18, 2018, Rhodes, Greece, American Institute of Physics Conference Proceedings, Vol. 2116. AIP Press, Melville, New York, USA, 2019, ISBN: 978-0-7354-1854-7, ISSN: 0094-243X (AIP CP, print), DOI: 10.1063/1.5114033.

[15] C.-P. Rückemann, "Superordinate Knowledge Based Comprehensive Subset of Conceptual Knowledge for Practical Mathematical-Computational Scenarios," in The Ninth Symp. on Adv. Comp. and Inf. in Natural and Applied Sciences (SACINAS), Proceedings of The 17th Int. Conf. of Num. Analysis and Appl. Math. (ICNAAM), Sept. 23–28, 2019, Rhodes, Greece, American Inst. of Physics Conf. Proc. AIP Press, Melville, New York, USA, 2020, ISSN: 0094-243X, (to appear).

[16] C.-P. Rückemann, Sustainable Knowledge and Resources Management for Environmental Information and Computation. Business Expert Press, Manhattan, New York, USA, Mar. 2018, Ch. 3, pp. 45–88, in: Huong Ha (ed.), Climate Change Management: Special Topics in the Context of Asia, ISBN: 978-1-94784-327-1, in: Robert Sroufe (ed.), Business Expert Press Environmental and Social Sustainability for Business Advantage Collection, ISSN: 2327-333X (collection, print).

[17] C.-P. Rückemann, "From Knowledge and Meaning Towards Knowledge Pattern Matching: Processing and Developing Knowledge Objects Targeting Geoscientific Context and Georeferencing," in Proc. of The Twelfth Internat. Conf. on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2020), November 21–25, 2020, Valencia, Spain. XPS Press, Wilmington, Delaware, USA, 2020, pp. 36–41, ISSN: 2308-393X, ISBN-13: 978-1-61208-762-7, URL: http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2020_3_10_30035 [accessed: 2021-11-27].

[18] "Perl Compatible Regular Expressions (PCRE)," 2021, URL: https://www.pcre.org/ [accessed: 2021-11-27].

[19] "The Perl Programming Language," 2021, URL: https://www.perl.org/ [accessed: 2021-11-27].

[20] C.-P. Rückemann, "Information Science and Inter-disciplinary Long-term Strategies – Key to Insight, Consistency, and Sustainability: Conceptual Knowledge Reference Methodology Spanning Prehistory, Archaeology, Natural Sciences, and Humanities," International Tutorial, DataSys Congress 2020, Sept. 27 – Oct. 1, 2020, Lisbon, Portugal, 2020, URL: http://www.iaria.org/conferences2020/ProgramINFOCOMP20.html [accessed: 2021-11-27].

[21] "The Prehistory and Archaeology Knowledge Archive (PAKA) license," 2021, (release 2021), Unabhängiges Deutsches Institut für Multidisziplinäre Forschung (DIMF): All rights reserved. Rights retain to the contributing creators. URL: http://scienceparagon.de/cpr/x/rprojs/en/rprojs.html#PAKA [accessed: 2021-11-27].

[22] C.-P. Rückemann, "Keynote Lecture: The Information Science Paragon: Allow Knowledge to Prevail, from Prehistory to Future – Approaches to Universality, Consistency, and Long-term Sustainability," October 20, 2020, Informational Modeling – Theory and Practice, Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, October 20–21, 2020, Sofia, Bulgaria, URL: http://math.bas.bg [accessed: 2021-11-27], 2020.

[23] C.-P. Rückemann, "The Information Science Paragon: Allow Knowledge to Prevail, from Prehistory to Future – Approaches to Universality, Consistency, and Long-term Sustainability," The International Journal "Information Models and Analyses" (IJ IMA), vol. 9, no. 3, 2020, pp. 203–226, Markov, K. (ed.), ISSN: 1314-6416 (print), Submitted accepted article: November 18, 2020, Publication date: August 17, 2021, URL: http://www.foibg.com/ijima/vol09/ijima09-03-p01.pdf [accessed: 2021-11-27].

[24] "UDC Summary Linked Data, Main Tables," 2021, URL: https://udcdata.info/078887 [accessed: 2021-11-27].

[25] "UDC 5: Mathematics. Natural sciences," 2021, URL: http://udcdata.info/025403 [accessed: 2021-11-27].

[26] "UDC 9: Geography. Biography. History," 2021, URL: http://udcdata.info/068076 [accessed: 2021-11-27].

[27] C.-P. Rückemann, R. Pavani, Z. Kovacheva, B. Gersbeck-Schierholz, F. Hülsmann, and I. Naydenova, Best Practice and Definitions – Concepts of Cognostic Addressing Structured and Non-structured Data. Post-Summit Results, Delegates' Summit: Best Practice and Definitions – Concepts of Cognostic Addressing Structured and Non-structured Data, The Eleventh Symposium on Advanced Computation and Information in Natural and Applied Sciences (SACINAS), The 19th International Conference of Numerical Analysis and Applied Mathematics (ICNAAM), September 20–26, 2021, Rhodes, Greece, 2021, (to appear).

[28] C.-P. Rückemann, "Coherent Knowledge Solutions From Prehistory to Future – Towards Coherent Multi-disciplinary Knowledge Reference Implementation Blueprints for Industrial Learning: Insight from Consistent Coherent Conceptual Integration of Prehistory, Archaeology, Natural Sciences, and Humanities," ML4I – Machine Learning for Industry Forum 2021; hosted by the High-Performance Computing Innovation Center (HPCIC) and Data Science Institute (DSI), at the Lawrence Livermore National Laboratory (LLNL), August 10–12, 2021, Livermore, California, U.S.A., (Invited Speech), URL: http://www.llnl.gov, 2021.

[29] "UDC "...": Common auxiliaries of time," 2021, Universal Decimal Classification (UDC), UDC Consortium, URL: http://udcdata.info/011472 [accessed: 2021-11-27].

[30] "UDC (1): Place and space in general. Localization. Orientation," 2021, Universal Decimal Classification (UDC), UDC Consortium, URL: http://udcdata.info/001952 [accessed: 2021-11-27].

[31] T. Vincenty, "Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations," Surv. Rev., vol. XXII, no. 176, 1975, pp. 88–93.

[32] "World reference base for soil resources," 1998, World Soil Resources Report No. 84, ISSS-ISRIC-FAO, Rome.

[33] "World reference base for soil resources," 2006, World Soil Resources Report No. 103, ISSS-ISRIC-FAO, Rome.

[34] P. Wessel, "The Digital Chart of the World for GMT 5 or later," 2018, URL: ftp://ftp.soest.hawaii.edu/gmt [accessed: 2021-11-27], URL: http://www.soest.hawaii.edu/pwessel/dcw/ [accessed: 2021-11-27].

[35] P. Wessel, "Global Self-consistent Hierarchical High-resolution Geogra-

phy," 2016, URL: ftp://ftp.soest.hawaii.edu/gmt [accessed: 2021-11-27], URL: http://www.soest.hawaii.edu/pwessel/gshhg/ [accessed: 2021-11-27].

[36] P. Wessel and W. H. F. Smith, "A global, self-consistent, hierarchical, high-resolution shoreline database," J. Geophys. Res., vol. 101, no. B4, 1996, pp. 8741–8743.

[37] E. A. Soluri and V. A. Woodson, "World Vector Shoreline," Int. Hydrograph. Rev., vol. LXVII, no. 1, 1990, pp. 27–35.

[38] A. J. Gorny, "World Data Bank II General User Guide," 1977.

[39] P. Wessel, W. H. F. Smith, R. Scharroo, J. Luis, and F. Wobbe, "The Generic Mapping Tools (GMT)," 2020, URL: http://www.generic-mapping-tools.org/ [accessed: 2021-11-27], URL: http://gmt.soest.hawaii.edu/ [accessed: 2021-11-27].

[40] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," Computational Science & Engineering, (IEEE), vol. 5, no. 1, 1998, pp. 46–55.

[41] OpenMP Architecture Review Board, "OpenMP application program interface version 3.0," May 2008, URL: http://www.openmp.org/mp-documents/spec30.pdf [accessed: 2021-11-27].

[42] OpenMP Architecture Review Board, "OpenMP API 5.1 Specification," Nov. 2020, URL: https://www.openmp.org/wp-content/uploads/OpenMP-API-Specification-5-1.pdf [accessed: 2021-11-27].

The Typed Graph Model – a Supermodel for Model Management and Data Integration

Fritz Laux
Fakultät Informatik
Reutlingen University
D-72762 Reutlingen, Germany
email: fritz.laux@fh-reutlingen.de

***Abstract*—In recent years, the Graph Model has become increasingly popular, especially in the application domain of social networks. The model has been semantically augmented with properties and labels attached to the graph elements. It is difficult to ensure data quality for the properties and the data structure because the model does not need a schema. In this paper, we propose a schema bound Typed Graph Model with properties and labels. These enhancements improve not only data quality but also the quality of graph analysis. The power of this model is provided by using hyper-nodes and hyper-edges, which allows to present data structures on different abstraction levels. We prove that the model is at least equivalent in expressive power to most popular data models. Therefore, it can be used as a supermodel for model management and data integration. We illustrate by example the superiority of this model over the property graph data model of Hidders and other prevalent data models, namely the relational, object-oriented, XML model, and RDF Schema.**

Keywords–typed hyper-graph model; semantic enhancement; data quality.

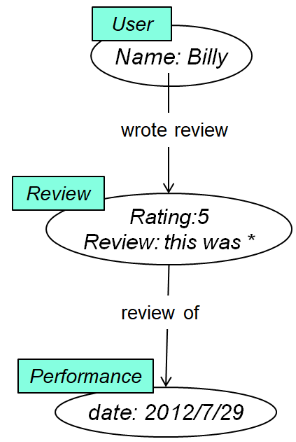


Figure 1. Example graph taken partially from [6], p. 42.

I. Introduction

The popularity of the Graph Model (GM) stems primarily from its application to social networks, medicine, scientific literature analysis, drug analysis, power and telephone networks. The flexibility of the GM contributes to its popularity, but its schema-less implementations are prone to data quality problems. This was pointed out in our DBKDA paper [1] in which we introduced the Typed Graph Model (TGM) with schema support. The present work extends our findings about the TGM and provides a proof of the expressive power and demonstrates its superiority over most popular data models.

Commercial graph database products like Neo4J [2], ArangoDB [3], JanusGraph [4], Amazon Neptune [5], and others have been successfully applied to many domains. Advocates of the GM like Robinson et al. of Neo4J recommend in their book [6] to use specification by example, which builds on example objects. But this reaches not far enough as the following example taken from Robinson's book shows. It is depicted in Figure 1 and shows a *User* named Billy with its 5-star *Review* on a *Performance* dated 2012/7/29. From this example we cannot know if Billy is allowed to have multiple reviews (on the same performance). For good data quality, a review should depend on the existence of a user and a performance. But this cannot be derived from one example. This means that we have to deal with class things (like a generic Person) and not only with real objects (like Billy) and specify if a relationship is mandatory or optional.

In order to express structural information, it is necessary to abstract from a particular situation and specify integrity constraints. The use of a schema would help to ensure data integrity and would clarify the intended situation of the example. Daniel et al. [7] complain that ”there are only few solutions that target conceptual modeling for NoSQL databases and even less focusing on graph databases”. They also point out the importance of a schema for data consistency and efficient implementation of a graph database and propose a framework, that translates an UML schema definition into a graph representation, and generate database-level queries from business rules and invariants.

Another weakness of the GM is that it has no notation to support different levels of detail and abstraction, which is apparently important for modeling large and complex data structures.

A. Contribution

To overcome these limitations we introduce in this paper a new typed graph model allowing hyper-nodes with complex structured properties (even sub-graphs) and hyper-edges connecting (recursively) one, two or more hyper-nodes. The graph schema provides data types, which allow type checking for instance elements. This ensures a formal data quality. We prove that the model is at least equivalent in expressive power to most popular data models. It can be used on both, the instance and schema level. Its semantic power makes it suitable for a loss-less model management and high quality data integration. Our model has a higher semantic expressiveness and precision than the prevalent data models, namely the relational, object oriented, XML data model, and RDF Schema. This will be demonstrated with typical modeling patterns.

B. Structure of the Paper

With the following overview of Related Work (Section II) the context for our new typed graph model will be settled. Section III introduces and defines formally the Typed Graph Model (TGM) consisting of a typed schema and a hypergraph instance connected to the schema. We present a compact and easy to read visualization of the model using UML. The definitions are illustrated by some examples and the abstraction capability is demonstrated by the data model of a commercial enterprise.

In Section IV we prove a semantic preserving schema translation for prevalent data models to the TGM. Properties of the TGM are explained and the translation process is illustrated for an Entity Relationship example. In the next Section V our TGM is compared to the Graph Data Model (GDM) of J. Hidders [8]. Then, the semantic expressiveness of the TGM is demonstrated with typical data structures and compared with the prevalent data models, namely the relational, object oriented, and XML data model. The paper ends with a summary of our findings and gives an outlook on ideas for future work.

II. Related Work

Since the beginning of 1980 many papers on the GM have been published. DBLP [9] alone retrieves 920 matches (retrieved July 30, 2021) for the key words ”graph data model”. If we ignore the papers that present specific applications for the GM incl. XML or Hypertext applications a few dozen of relevant papers remain. In the following, we discuss only papers that present the GM and its extensions (e. g., the Property Graph Model (PGM)) with a formal foundation or papers that use a graph schema:

The notion of PGM was informally introduced by Rodriguez and Neubauer [10]. Spyratos and Sugibuchi [11] use property graphs with hyper-nodes and hyper-edges for their graph data model. The main difference to our approach is that no schema is used and properties have no predefined data type. Another approach with hyper-edges is presented by Bu et al. [12] who treats a label like a node connecting a set of nodes, which he calls hyper-edge. The nodes itself can be of different types. In this case Bu calls the graph a unified hyper-graph. The unified hyper-graph model is then applied to problems of ranking music content and combining it with social media information. Compared to our TGM the unified hyper-graph of Bu is only defined for graph instances. It is not not clear if the nodes have any type checking and if the whole graph is ruled by a schema.

Ghrab et al. [13] present GRAB, a schemaless graph database based on the PGM. It supports integrity constraints but cannot ensure data quality because of missing data types for properties and labels. Neo4J [6] has similar foundations and features. It has optional support for integrity constraints and comes with a powerful and easy to use graph query language, called *Cypher*.

All these PGM variants originate as instance graphs and no special attention is given to the graph schema. No attempt is made to specify the different types of edges and the multiplicity of connections (edges) between different node types. Nodes are not typed and labels are not a proper substitute for data types. Therefore it is important to combine the PGM with a schema.

Amann and Scholl [14] seem to be the first authors who connect a graph schema with its graph database instance. Nodes and edges do not have properties but both must conform to the schema. Their model is used for an algebra (hyperwalk algebra) for traversing the graph.

Marc Gyssens et al. [15] and Jan Hidders [8] use a labeled GM to represent a database schema where each property of an object is modeled as a node in the graph. Labels are used to name node classes and edges. The models become confusing because a node represents either an object, a property or a data type. Still, it is not possible to restrict the cardinality of schema edges (relationships). Hidders’ model is explained in more detail and compared to our TGM in Section V.

Similar to Amann and Scholl the paper of Pabón et al. [16] uses a graph schema to query the graph database. They distinguish different node types, which they call ”sort”. The supported types are: *object class nodes* (complex objects), *composite-value class nodes* (for aggregate values), and *basic-value class nodes* (primitive data types). This model seems to be equivalent to (complex) nodes with properties governed by a schema. A mechanism to abstract and group sub-graphs is missing, but would help to make the model easier to communicate.

Pokorný [17] uses a binary ER-Model as graph conceptual schema. For the graphical rendering he uses a compact entity representation for the nodes with attribute names inside the entity box. This solves the problem using the same node symbol for entities and attributes (properties) as it is the case with Gyssens [15] and Hidders [8] models. The edge cardinality is represented in a form of crow-foot notation.

In order to make the GM usable for real life scenarios with hundreds of schema elements, it is necessary to group or combine graph elements to higher abstracted objects. This would make the model easier to handle.

The need for grouping graph elements is addressed by Junghanns et al. [18]. Their model allows to form logical sub-graphs (graph collections) with heterogeneous nodes and edges. With this it is possible to aggregate sub-graphs, e. g., user communities. The authors use UML-like graphical rendering of nodes to make the model better readable but their model fails to specify the cardinality of schema edges.

A step toward to complex composite nodes as an alternative approach to aggregation presents Levene [19] by allowing the graph vertices to be recursively defined as a finite set of graphs. These hyper-nodes do not form a well-founded set as a node may contain itself, which violates the foundation axiom for the Zermelo-Fraenkel set theory.

A relatively new formal definition including integrity constraints was given by Angles [20]. However, his model does not allow structured objects and grouping or aggregation. In the following section, we simplify his definitions and use it as basis for our TGM.

A. Comparison with Ontology Languages

Ontology languages like the Resource Description Framework Schema (RDFS) [21] and the Web Object Language

(OWL) [22] are designed to specify ontologies and have their strength in allowing reasoning over instances of it. They are often used to semantically describe Linked Open Data (LOD) and the statement triples are usually visualized as graph structures. RDFS and OWL provide a general type system that could be used to form user defined types. This would allow to use it as basis for a graph schema language. But if we look at the W3C OWL 2 Structural Specification [23] it seems difficult to define user specific classes and W3C itself uses UML class diagrams to illustrate OWL structures.

Most approaches that map RDF to property graphs only support instance graphs. This is the case for the papers of Chiba et al., Schätzle et al., and Nguyen et al. The paper of Chiba et al. [24] uses G2GML, a graph-to-graph mapping language where the source graph is a RDF-graph and the destination graph is a Property Graph (PG). With G2GML RDF patterns are specified in SPARQL syntax and the corresponding patterns of the PG are described in openCypher [25]. These patterns are mapped through *node maps* and *edge maps*. The main benefit is that the pattern specifications are domain-specific and declarative. There is no support for schema mappings.

Schätzle et al. [26] define a mapping from RDF to the property graph model of GraphX. Their aim is to provide better analysis performance with S2X, a SPARQL implementation on top of GraphX and Spark, despite the schema-free PGM. Again, no schema mapping is supported.

The paper of Nguyen et al. [27] represents each RDF triple element is as a separate node, which justifies the name Labeled Directed Multigraph with Triple Nodes (LDM-3N). While other models represent predicates as labeled arcs, Nguyen et al. map them to nodes. Assertions about RDF statements (triples) are modeled with the singleton property and not by reification. This approach adds an extra computation step and doubles the number of triples, which bloats the graph model. The main application domain seems to be the analysis of RDF triples by mapping it to the LDM-3N graph model allowing the use of graph analysis algorithms. Finally, it does not support schema mapping.

The specification of data structures is not the core intention of RDF. In RDFS for instance it is not possible to define the cardinality of relationships. Likewise, OWL Lite has strong limitations on allowing only 0 or 1 as multiplicity of properties. Simple unique requirements and relations like one-to-one, one-to-many and many-to-one are cumbersome to define even in OWL Full. Complex data structures need a modeling language that allows to define different levels of abstraction, which is not the strength of these ontology languages. Most examples of RDFS or OWL do not care about the multiplicity of relationships (cardinalities may be guessed via property names) and grouping of attributes seem to be on the same level as objects or subjects.

All these arguments and examples make it clear that we need schema support to ensure high data quality when using graph databases. This can be achieved with the Typed Graph Model (TGM), which we develop in the next section.

III. The Typed Graph Model

Our TGM informally constitutes a directed property hypergraph that conforms to a schema. In the following definitions our notation uses small letters for elements (nodes, edges, data types, etc.) and capital letters for sets of elements. Sets of sets are printed as bold capital letters. A typical example would be $n \in N \in \mathbf{N} \subseteq \wp(N)$, where N is any set and $\wp(N)$ is the power-set of N.

A. Graph Schema

Let T denote a set of simple or structured (complex) data types. A data type $t := (l, d) \in T$ has a name l and a definition d. Examples of simple (predefined) types are $(int, \mathbb{Z})$, $(char, ASCII)$, $(\%, [0..100])$ etc. It is also possible to define complex data types like an order line $(OrderLine, (posNo, partNo, partDescription, quantity))$. The components need to be defined in T as well, e. g., $(posNo, int > 0)$. Recursion is allowed as long as the defined structure has a finite number of components.

Definition 1 (Typed Graph Schema)**.** *A typed graph schema is a tuple* $TGS = (N_S, E_S, \rho, T, \tau, C)$ *where:*

- N_S *is the set of named (labeled) objects (nodes)* n *with properties of data type* $t := (l, d) \in T$*, where* l *is the label and* d *the data type definition.*
- E_S *is the set of named (labeled) edges* e *with a structured property* $p := (l, d) \in T$*, where* l *is the label and* d *the data type definition.*
- ρ *is a function that associates each edge* e *to a pair of object sets* (O, A)*, i. e.,* $\rho(e) := (O_e, A_e)$ *with* $O_e, A_e \in \wp(N_S)$*.* O_e *is called the* tail *and* A_e *is called the* head *of an edge* e*.*
- τ *is a function that assigns for each node* n *of an edge* e *a pair of positive integers* (i_n, k_n)*, i. e.,* $\tau_e(n) := (i_n, k_n)$ *with* $i_n \in \mathbb{N}_0$ *and* $k_n \in \mathbb{N}$*. The function* τ *defines the min-max multiplicity of an edge connection. If the min-value* i_n *is* 0 *then the connection is optional.*
- C *is a set of integrity constraints, which the graph database must obey. The constraint language may be freely chosen.*

The notation for defining data types T, which are used for node types N_S and edge types E_S, can be freely chosen. The integrity constraints C restrict the model beyond the structural limitations of the multiplicity τ of edge connections. Typical constraints af C are semantic restrictions of the content of an instance graph. This makes the expressiveness of the TGS at least as strong as the models to which it is compared in Section V.

B. Typed Graph Model

Definition 2 (Typed Graph Model)**.** *A typed graph Model is a tuple* $TGM = (N, E, TGS, \phi)$ *where:*

- N *is the set of named (labeled) nodes* n *with data types from* N_S *of schema TGS.*
- E *is the set of named (labeled) edges* e *with properties of types from* E_S *of schema TGS.*

- *TGS is a typed graph schema as defined in Subsection III-A.*
- *ϕ is a homomorphism that maps each node n and edge e of TGM to the corresponding type element of TGS, formally:*

$$\begin{aligned} \phi: \quad TGM &\rightarrow TGS \\ n &\mapsto \phi(n) := n_S (\in N_S) \\ e &\mapsto \phi(e) := e_S (\in E_S) \end{aligned}$$

The fact that ϕ maps each element (node or edge) to exactly one data type implies that each element of the graph model has a well defined data type. The homomorphism is structure preserving. This means that the cardinality of the edge types are enforced, too. Data type and constraint checking is applied for all nodes and edges before any insert, update, or delete action can be committed. If no single type can be defined, union type or *anyType* (sometimes called *variant*) may be applied. Usually this is an indication for a weak data model and it should be clear that this could affect data quality and processing.

As graphical representation for the TGS we adopt the UML-notation for nodes and include the properties as attributes including their data types. Labels are written in the top compartment of the UML-class. Edges of the TGS are represented by UML associations. For the label and properties of an edge we use the UML-association class, which has the same rendering as an ordinary class but its existence depends on an association (edge), which is indicated by a dotted line from the association class to the edge. This not only allows to label an edge but to define user defined edge types.

The correspondence between the UML notation and the TGS definition is shown in Table I.

TABLE I. TGS correspondence with UML notation

TGS	UML
$n \in N_S$	class
$e \in E_S$	association
$t = (l, d) \in T$	l = name of n resp. e; d = type of n resp. e
$\rho(e)$	all ends of e
$\tau_e(n)$	(min,max)-cardinality of e at n
C	constraints in [ ] or { }

The use of hyper-nodes $n \in N_S$ and hyper-edges $e \in E_S$ instead of simple nodes resp. edges allow to group nodes and edges to higher abstracted complex model aggregates. This is particularly useful to keep large models clearly represented and manageable. Large graph models may then be grouped into sub-graphs like in Junghanns et al. [18]. Each sub-graph can be rendered as a hyper-node. If the division is disjoint these hyper-nodes are connected via hyper-edges forming a higher abstraction level schema (see Figure 3 (b)).

C. Examples

Lets recall the example graph from Figure 1 and model its corresponding schema. We want to make clear that a user may write as many reviews as he likes, but only one for a particular performance. A rating needs to refer exactly to one performance and one user. This is reflected in Figure 2 by the

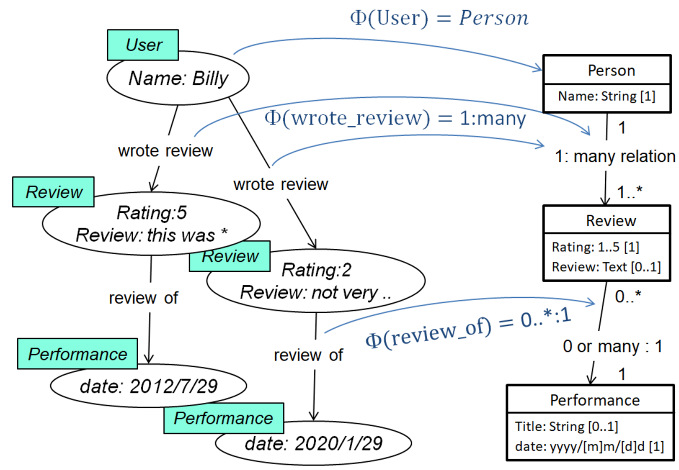


Figure 2. Example graph with schema in UML notation.

"1:many" and "0 or many:1" relationships. We use the UML-notation for the schema and keep the notation from Figure 1 for the instance graph for clarity.

The homomorphic mapping ϕ guaranties that the instance graph obeys the schema, i. e., type, cardinality, and constraint checking. Now, it is clear from the schema that a user must have at least one review. The review is existence dependent on the user and a performance. The "wrote review" edge is a 1:many relation and "review of" is an optional many:1 relation. This has the consequence that a review needs a person and a performance. But, a performance may exist without any review.

In the next example we present a commercial enterprise that sells products and parts to customers. The enterprise assembles products from parts and if the stock level is not sufficient it purchases parts from different suppliers. Figure 3 models this situation using UML rendering. It demonstrates the abstraction power of the TGM showing two schema abstraction levels. The upper part (a) shows the TGM on a detailed level. The properties are suppressed in the diagram for simplicity except for *Customer* and *CustOrder*. The schema is grouped into 3 disjoint sub-graphs depicted with dashed lines.

In the lower part (b) these sub-graphs are shown as hyper-nodes of the graph schema. This allows a simplified and more abstracted view of the model. Also, some aggregate properties (e. g., #orders) are shown to illustrate the modeling capabilities. The coloring of the (hyper-)edges helps the reader to identify which edges have been aggregated. The hyper-edges connecting these abstracted nodes must use the most general multiplicity of the multiple edges it combines. In the example the edge *orders/from* combines two edges, i. e., *orders* with 0..1 - 1 multiplicity and *from* with 0..* - 0..* multiplicity, which leads to the most general multiplicity.

IV. Properties of the Typed Graph Model

The TGM has some valuable properties and can be regarded as a **supermodel** because its semantic expressiveness proves to be at least as powerful as prevalent data models. In order to prove this statement we need the following definitions.

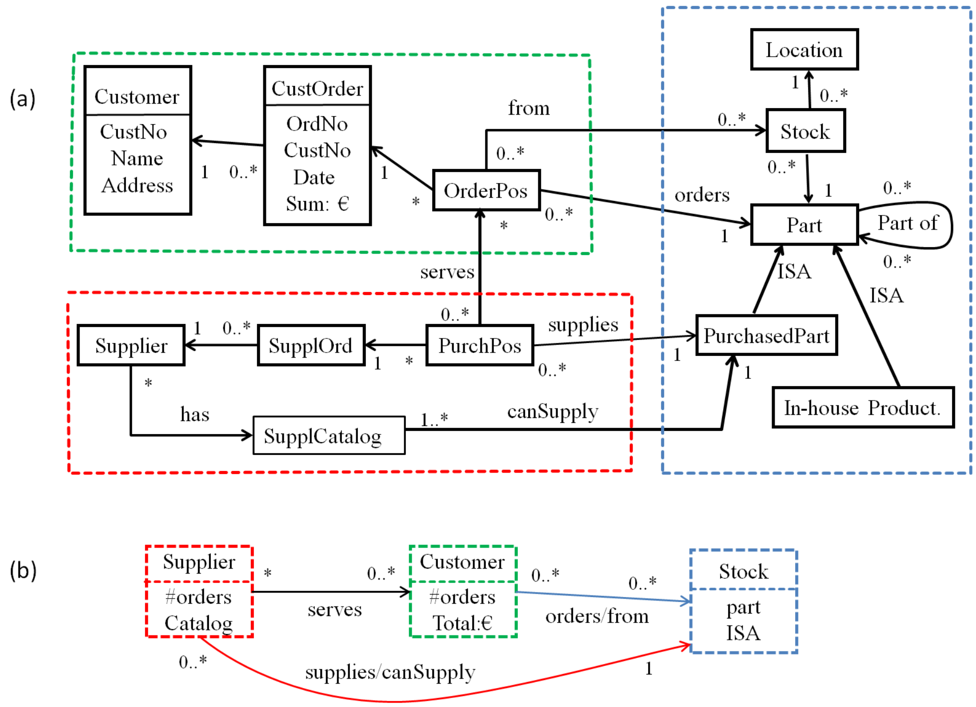


Figure 3. Example TGM of a commercial enterprise showing two levels of detail.

As we are only interested in databases that satisfy the constraints of its schema, we modify the definitions from Angles et al. [28] and restrict it to schema bound databases. More precisely, let M be a database model and S^M be a database schema. A database is then an ordered pair $D^M := (S^M, I^M)$, where I^M is an instance obeying all constraints of S^M.

Definition 3 (Schema Mapping)**.** *Let M_1 and M_2 be two data models. A schema mapping from M_1 to M_2 is a total function* **SM** *from the set of all database schemas in M_1, to the set of all database schemas in M_2.*

$$\mathbf{SM} : M_1 \rightarrow M_2$$
$$s_1 \mapsto s_2$$

where s_1 and s_2 are schemas of M_1 resp. M_2.

A. Some Property Definitions

Every data model allows to structure the data according to its modeling elements. These conceptual elements determine the representational power of the data model. A model M_2 subsumes the information capacity of M_1 if and only if every schema in M_1 can be translated to a schema in M_2 without loss of information. Two database models can be evaluated in terms of its information capacity considering the following properties:

Definition 4 (Computable Mapping)**.** *A schema mapping CM from $s_1 \in M_1$ to $s_2 \in M_2$ is computable if there exists an algorithm that translates schema $s_1$into s_2.*

If any schema mapping from M_1 to M_2 is computable then this implies that M_2 subsumes M_1. A computable mapping may still result in a schema that allows an invalid database instance.

Definition 5 (Semantics Preservation)**.** *A computable schema mapping SP is semantics preserving if for every valid database d_1 of schema s_1, there is a valid database d_2 obeying schema s_2 where s_2 is produced by the mapping SP, i. e., $s_2 = SP(s_1)$.*

This property guarantees that the result of the instance mapping will always be a valid database according to s_2.

Definition 6 (Information Preservation)**.** *A semantics preserving schema mapping IP is information preserving if there exists an inverse computable schema mapping IP^{-1} from M_2 to M_1 such that for every $s_1 \in M_1$ it holds $s_1 = IP^{-1}(IP(s_1))$. Such a schema mapping IP is alternatively called "schema translation".*

This definition indicates that, for some schema mapping IP, there exists an "inverse" mapping which allows recovering the original schema previously transformed. In general, the inverse mapping IP^{-1} is only a partial function because it is only defined on s_2 and may not be defined on all elements of M_2, i. e., the image $SM(M_1)$ can be a proper subset of M_2. Information preservation is an important property because it guarantees that a schema mapping results in a new schema capable to not lose any information. Moreover, it implies that the target database model M_2 subsumes the information capacity of the source database model M_1. If M_1 subsumes M_2 as well, then both data models are equivalent and IP^{-1} and IP are total functions. McBrien and Poulovassilis [29][30] describe equivalence-preserving mapping of schema constructs based on a Hypergraph Data Model (HDM). The ideas and problems with schema translation have been reviewed and revised in the light of model management by Bernstein and Melnik [31].

B. Schema Mapping by means of a Meta-model

When considering database mappings two types of mappings can be distinguished: (1) schema independent and (2) schema dependent. We are only interested in schema dependent mappings to always ensure high data fidelity. The above schema mapping of Definition 3 is model independent [32]. A meta-model that is general enough to capture all popular models in the literature would suffice. Hull [33] as well as Atzeni and Torlone [34][35] describe such a framework for heterogeneous data models. It was developed towards a model independent **supermodel** [36] and later implemented as a general tool, called MIDST [37]–[39]. It consists of the following meta-constructs: *lexical, abstract, aggregation, generalization, and function*. For example, the Entity-Relationship Model (ERM) involves (i) abstracts (the entities), (ii) aggregations of abstracts (relationships), and (iii) lexicals (attributes) with functions (to entities or relationships). This means that the ERM is a specialization of the supermodel, i. e., a schema in any (sub)model is also a schema in the supermodel, only the names of the model elements differ. Hull and Atzeni give more examples and claim that this supermodel subsumes Relational Model (RM), ERM, XML, Object Oriented Model (OOM), Object Relational (OR), and XSD.

C. Information Preserving Schema Translation to TGM

There are works that show information preserving mappings from ERM to Graph Database Schema [40], RM to RDF and OWL [41], and RM, CSV, XML JSON to RDF using RML, the RDF Mapping Language [42]. Angles et al. [28] have shown that a RDF database can be mapped to a property graph database. This includes information preservation for the schema translation and the instance mapping. Taking all together it seems possible to translate most popular data models to an enhanced graph data model, i. e., a PGM. It is evident that the TGM subsumes the PGM as any model element of PGM can be mapped 1:1 to the corresponding TGM model element. The more general approaches of Arenas et al. [43] and McBrien/Poulovassilis [30] provide criteria for model independent schema mappings that are information preserving. Hull [33] and Atzeni/Torlone [34][35] propose basic meta-constructs for the supermodel that covers all relevant data models. This puts us now in a position to state our main Theorem.

Theorem 1 (Information Preserving Schema Translation to TGM)**.** *Let M be any data model that can be subsumed by the **supermodel** of Hull and let T be the Typed Graph Model. For any schema $s \in M$ there exists an information preserving mapping (translation) $\mathbf{M}$:*

$$\begin{aligned}\mathbf{M} : M &\rightarrow T \\ s &\mapsto t\end{aligned}$$

where t is a TGS and s is a schema from M.

It would be easy to proof the theorem by contradiction. But such a proofs gives no constructive idea how a concrete mapping would look like. Therefore, we proof the theorem by constructing a generic mapping $\mathbf{M}$ taking the meta-constructs of the supermodel and assign uniquely model elements from T. These model element pairs allow us to map any schema $s \in M$ to a schema $t \in T$. Then it will be shown that we can construct an inverse mapping that leads to the original schema s again.

Proof of the Translation Theorem: Following Atzeni and Torlone we define the schema mapping $\mathbf{M}(M) = T$ by the following elementary 1:1 transformations for each model element $\sigma \in s$:

1) lexical $\rightarrow$ property
2) abstract $\rightarrow$ node
3) aggregation $\rightarrow$ edge with aggregation type
4) generalization $\rightarrow$ edge with generalization type
5) function $\rightarrow$ edge with a single target (multiplicity 1)

The above transformation can be chained or composed to form a directed acyclic graph where each transformation assignment represents a node and the directed edges represent the sequence of assignments. Each path from a leaf node ends at the root node representing the mapping $\mathbf{M}$. Given a schema s the mapping $\mathbf{M}$ translates the input s into a TGS $t \in T$. All transformation steps are 1:1 such that the resulting function $\mathbf{M}$ is a injective mapping that can be reversed. The inverse mapping $\mathbf{M^{-1}}$ uses just the opposite transformation assignments listed above. It should be noted that $\mathbf{M^{-1}}$ may be a partial function only, i. e., there might exist model elements in T (e. g., a composition edge type with existential dependency of its components) that may have no corresponding model element in the source model M.

Let σ w.l.o.g. be any model element from schema s. $\mathbf{M}$ translates σ into an element $\tau \in T$ in the following way:

$$\tau = \mathbf{M}(\sigma) = \mathbf{m_n}(\mathbf{m_{n-1}}(...(\mathbf{m_1}(\sigma)...))$$

where $\mathbf{m_i}$ are the elementary transformations from above. The data type σ_t of an element σ is carried over to the same data type of the translated element τ. Applying the inverse transformations $\mathbf{m_i^{-1}}$ in opposite order results in the identiy function $\mathbf{I}$ which proofs the information preservation property.

$$\begin{aligned}\sigma = \mathbf{m_1^{-1}}(...(\mathbf{m_{n-1}^{-1}}(\mathbf{m_n^{-1}}(\mathbf{m_n}(\mathbf{m_{n-1}}(...(\mathbf{m_1}(\sigma)...)) \\ = \mathbf{M^{-1}}(\mathbf{M}(\sigma)) = \mathbf{I}(\sigma))\end{aligned}$$

■

D. Example Translation from extended ERM to TGM

To demonstrate the translation algorithm we use an example from Hidders' GDM [8], which will be discussed and compared to the TGM in detail in the next section. At the moment, we concentrate on the translation process taking Hidders' example but using the ERM enhanced with generalization (IS-A) relationship. The original visualization of Hidders is depicted in Figure 5 (a). The extended Entity-Relationship (ER) diagram is shown in Figure 4 where the blue numbers next to the ER-symbols refer to the elementary transformations from Theorem 1.

The complete translation has to execute the elementary transformation steps from left to right (in order of the blue arcs) until all ERM elements have been converted. For instance, before the relationship *Contract* can be mapped to an edge of general aggregate type as indicated by transformation 3), the relationship attributes need to be mapped first. The *salary* is considered as a literal (lexical 1) that transforms directly

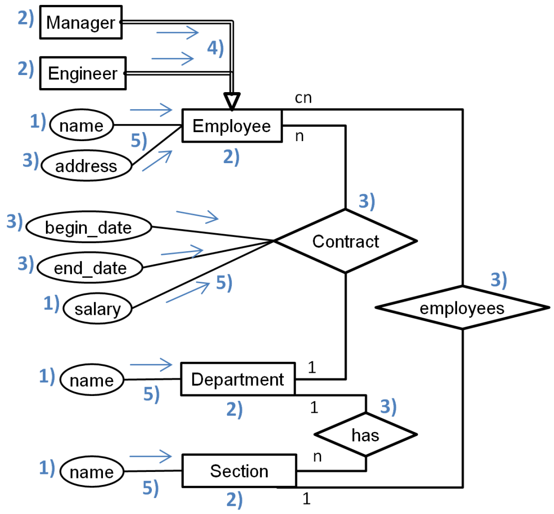


Figure 4. Hidders' example [8] transformed to a TGS.

to a *Contract* property. The *begin_date* and *end_date* are considered as aggregation model elements because of its date structure. All three attributes are single valued (indicated by the function 5) and mapped as properties of *Contract*.

It is also possible to specify more precisely the type of aggregation in the TGM by defining an edge type "Contract" that already includes the properties *begin_date*, *end_date*, and *salary*. This would result into the same semantics but with a user defined edge type. It should be pointed out that this possibility results from TGM's capability to support different abstraction levels.

The result of the translation is shown in the lower part of Figure 5. The visualization of the TGS uses the UML rendering listed in Table I.

V. Comparison with other Data Models

In the following, we compare our TGM to other models with respect to structural differences and schema support. We point out modeling restrictions of these models and show how such situations are modeled with TGM. Query and manipulation languages are beyond the scope of this paper.

A. Comparison with GDM of Jan Hidders

Jan Hidders' [8] model added labels and properties together with their data types to nodes and edges (relationships). Property names are modeled as edges in the schema. This allows to model labeled relationships with complex properties. Structured and base data types share the same graphical representation, which makes it difficult to distinguish both. The ISA-relationship is rendered as a double line arrow similar to the extended ERM. Hidders' model does not allow to restrict the cardinality of relationships. This restriction limits its modeling power compared to the TGM, which provides a min-max notation for the cardinality.

The example in Figure 5 is from the publication of Hidders [8]. The schema shows *Employee* and *Department* classes linked by a *Contract*. The relationship *Contract* is existence

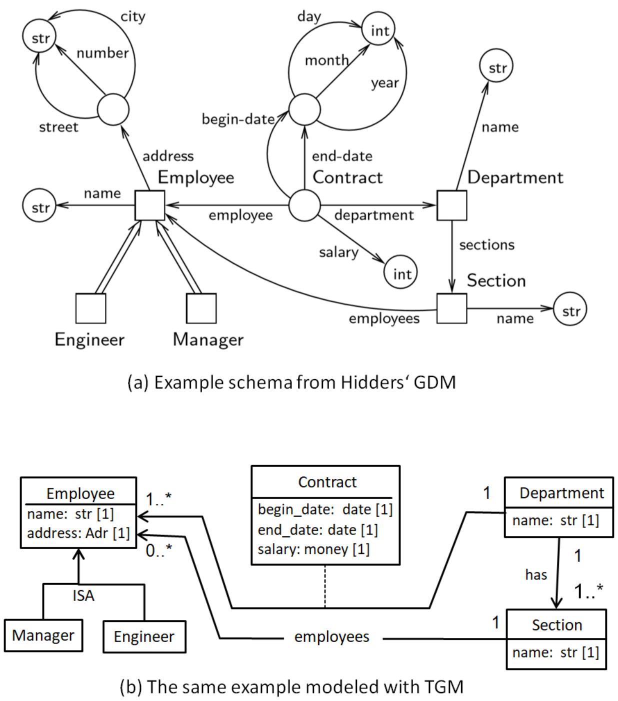


Figure 5. Comparison by example with Hidders' GDM.

dependent on the connected nodes. The properties of *Contract* are salary of type *int*, begin-date and end-date of structure-type $date = (day, month, year)$. In Hidders' model these dates are modeled on the element level using data type *int*. Hidders' schema elements, i. e., nodes (objects), edges (properties) and data types appear on the same visual level, which makes it difficult to read and obscures semantics. The modeling power of complex data types provide a clear advantage for the TGM.

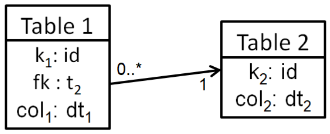
B. Comparison with the Relational Model (RM)

There is a 1:1 correspondence between attributes and properties and any relation can be modeled as a node with properties. The min-max notation for relationship multiplicity can model any link cardinality. The TGM can therefore easily represent tabular structures, foreign key constraints (many-to-one relationships), and join-tables as the building blocks of the RM. Beyond this, the TGM is able to directly model many-to-many relationships of any min-max multiplicity. This makes the TGM strictly stronger than the relational model.

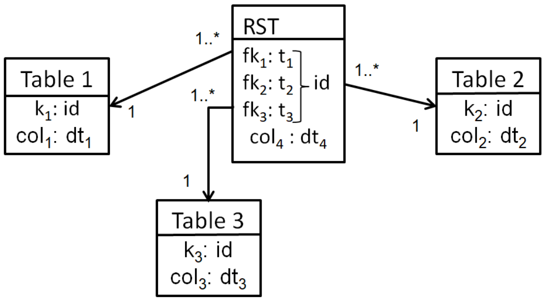
Figure 6 (a) shows a typical FK relationship between Table 1 and Table 2. In part (b) of Figure 6 three tables are liked via a join-table RST with foreign keys fk_1, fk_2, fk_3 representing a many-to-many ternary relationship with attribute col_4.

Another difference between TGM and RM is that foreign keys (FK) are not necessary because their function is taken over by an edge linking the Tab_1-node (Table 1 without FK) with the referenced node Table 2. This can be seen in Figure 7 (a).

A join-table in the RM is existence dependent on the tables it refers to by FKs. The FKs forming the primary key (PK) of the join table are not necessary in the TGM because of the same reason as mentioned above.

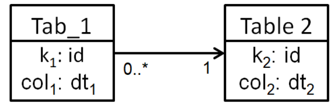


(a) 2 tables with FK relationship

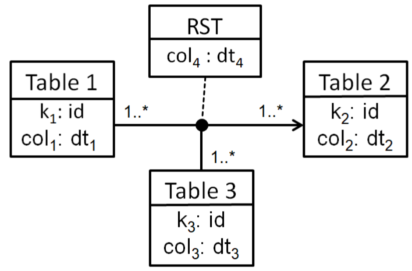


(b) 3 tables connected by a ternary join-table

Figure 6. Modeling a many-to-one relationship (FK) and a ternary join-table with the RM.



(a) 2 nodes (tables) with many-to-one link (relationship)



(b) 3 nodes (tables) connected by a ternary hyper-edge (join-table)

Figure 7. Modeling a many-to-one relationship (FK) and a ternary join-table with TGM.

In Figure 7 (b) the join-table RST maps directly to an hyper-edge labeled RST with property col_4. The hyper-edge and the join-table in the relational case represent a many-to-many connection linking 3 nodes that correspond to the respective tables in the RM. In the relational model it is not possible to restrict this ternary relationship to (*,1,1). For instance, if many connections of Table 1 should link to exactly *one* connection of Table 2 and *one* connection of Table 3. For the TGM this would be simple; only the cardinality at Table 2 and Table 3 would have to be changed from 1..* to 1.

To make the above ternary relationship example less abstract the RST could be an offer of products from Table 1 from one supplier of Table 3 to the client of Table 2. With this in mind it is clear that an offer depends on the product(s), one supplier, and one client.

The TGM can also represent non-normalized tables because the model supports complex structured data types having multivalued or array data. It is only necessary to define the appropriate data types in the set of available data types T.

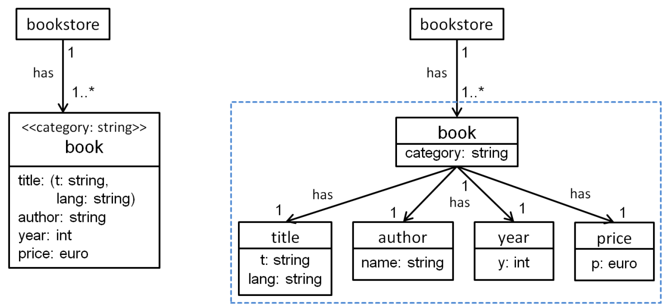
C. Comparison with XML Schema

XML documents represent hierarchical hypertext documents. The document structure can be defined by an XML schema. The hierarchy of XML-documents is directly supported by the TGM using directed edges. XLink provides references (arcs) between elements of internal or external XML-documents. Extended XLinks can connect to more than one element, but the references are always instance based, i. e., the target elements must be listed by URI. The TGM is more abstract and expressive allowing the definition of non-hierarchical references on the schema level.

As example for the comparison serves a bookstore offering an unlimited number of books. A simple XML-schema for the bookstore is given by w3schools.com. The schema defines books with elements like ”title”, ”author”, etc. and its corresponding data types. Some data types are not as precise as they could, e. g., the data type xs:double for the price element. We will replace xs:double in our TGM by the money-type *euro* to be more precise. Some elements have attributes attached like the language (”lang”) of a book title. The attribute minOccurs=”1” of xs:sequence requires the bookstore to have a least one book.

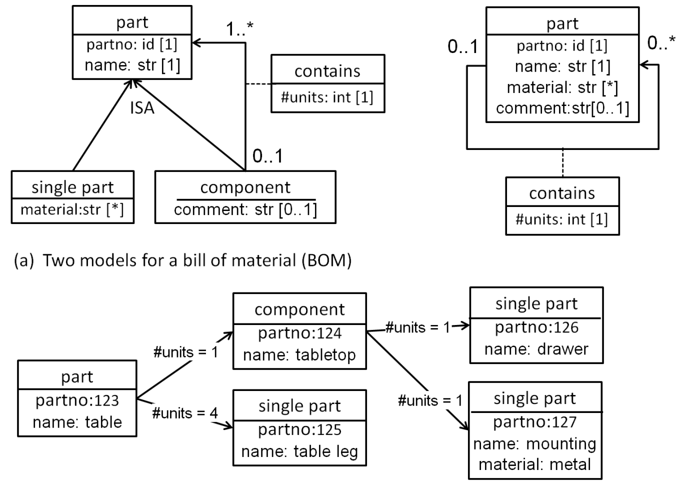
```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema ... >
  <xs:element name="bookstore" >
    <xs:complexType >
      <xs:sequence minOccurs="1"
          maxOccurs ="unbounded" >
        <xs:element name="book" >
          <xs:complexType>
            <xs:sequence>
              <xs:element name="title" >
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:string">
                    <xs:attribute name="lang"
                        type="xs:string" />
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
                </xs:element>
              <xs:element name="author"
                  type ="xs:string"/>
              <xs:element name="year"
                  type ="xs:integer"/>
              <xs:element name="price"
                  type ="xs:double"/>
            </xs:sequence>
            <xs:attribute name="category"
                type="xs:string"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

If we model the XML-elements as nodes in TGM then



(a) Compact representation (b) Detailed representation of each XML-element

Figure 8. Comparing an XML Schema example with TGM.



(a) Two models for a bill of material (BOM)

(b) An instance of a product (table) structure conforming to the above schemas

Figure 9. Comparison by example with the OOM.

XML-attributes and the element values should be represented as properties. The name of an XML-element is mapped to a node label. The order of the XML-elements cannot be represented with this approach and XML-element values can be distinguished from XML-attributes by convention only.

An alternative TGM model represents the complete book structure as one node. In this case the XML-elements and their attributes are modeled as structured properties of the book. The order of the elements and their associated attributes can be preserved. In fact, if XML Schema is used for specifying the data types N_S and E_S (see Subsection III-A) all the flexibility and semantics provided by XML Schema can be represented with the TGS. This argument shows that the TGM is at least as powerful as the XML model.

The example bookstore is depicted in Figure 8 where the left part (a) shows the compact version with the whole book modeled as one node and the right part (b) shows the version where each XML-element is modeled as node. We see from this example another possibility to use sub-graphs for higher abstracted models.

D. Comparison with the Object-Oriented Model

Because we already use the UML for rendering the TGM, it is easy to see that classes correspond one-to-one with typed hyper-nodes. Any methods are simply ignored as we only deal with the network structure of OOM. Any complex internal class structure can be directly modeled by appropriate data types $t \in T$. The type set T is defined beforehand but can contain any user defined structures. In contrast to the OOM the TGM allows different levels of abstraction in the modeling depending whether a structure is modeled by a detailed graph with simple types or a more compact graph using complex data types. This shows the same semantic expressiveness for structures, but a higher flexibility of the TGM. Considering the operations on data the OOM has the advantage to specify the allowed operations by methods.

The UML provides a rich set of association types, which need to be mapped to the label of the edges. Our TGM provides types not only for nodes but also for edges (called associations in UML). With this information it is possible to model different association types like aggregation, generalization, etc. Even user defined associations are possible, e. g., an aggregate could be further qualified as un-detachable or detachable composition or a loose containment. The arrow of the edge only indicates the reading direction of the association but does not limit the navigation of the TGM.

It is also possible to model recursive structures as the examples from Figure 9 illustrates. The bill of material (BOM) is an important example for a recursive structure used in production planning and control. It defines recursively a (compound) part with its components until a single part is reached. As example instance a *table* is given in Figure 9 (b) consisting of 4 table legs and a tabletop consisting of a drawer and a mounting.

If the edge of *contains* in Figure 9 (a) is followed against the arrow direction it is possible to find the component where an individual part is built-in (used). A complete *where-used list* for a generic (not an individual) part may be obtained with a small schema modification. Only the from-end of the *contains*-edge needs to change its multiplicity from 0..1 to 0..*. With this small modification all components can be identified where a generic part is used.

E. Comparison with RDF Schema

RDF is a data format for expressing statements about resources with emphasis on Web data. A *resource* can be anything, including data, objects (incl. people), and (abstract) concepts. RDF Schema uses a semantic extension (meta-constructs) of the basic RDF vocabulary to model RDF data. Simple statements about a resource are expressed in the form of *subject-predicate-object* triples. It is clear that the underlying structure of such RDF-triples is graph based where a node represents a *subject* or an *object* and an edge represents the *predicate*. This implies that RDF Schema triples can be expressed as property graph.

Angles et al. [28] describe a schema-dependent computable schema mapping from an RDF Schema to a Property Graph Schema (PGS) that is semantics and information preserving. They prove in their paper that ”the PGM subsumes the information capacity of the RDF data model”. The mapping is illustrated in their paper with an example RDF Schema depicted in Figure 10. Because of the asymmetrical definition

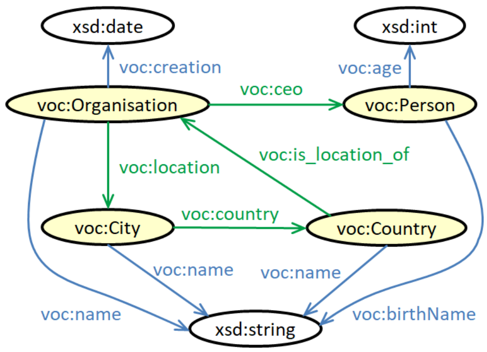


Figure 10. Graphical Illustration of the Example RDFS from Angles [28].

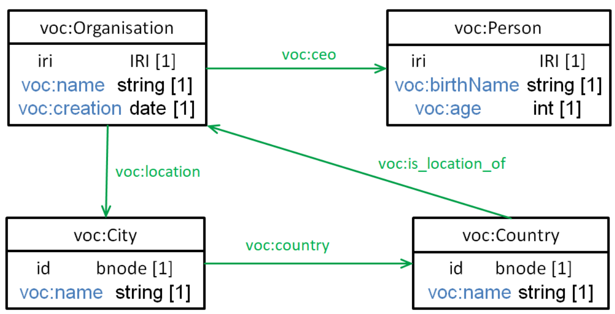


Figure 11. Resulting property graph schema of example Figure 10.

of subject (may not be a literal) and object (can be a literal) the mapping distinguishes objects that are also subjects and objects that are ”only objects”. The yellow colored subjects are mapped to nodes in the PGS and the ”only objects” (the data types xsd:date, xsd:int, and xsd:string in their example) are mapped to PGS properties. The green colored predicates are RDF predicates which are mapped to PGS edges, whereas the blue colored predicates are properties of the subject with a data type specified by an ”only object”. The resulting property graph schema has 4 nodes and 4 edges and is visualized in Figure 11 as TGS using the same coloring scheme as in the RDF schema.

When comparing the RDF Schema with the TGS we see that both share most characteristics. In particular, nodes and edges have unique labels. These labels serve as global identifiers, called International Resource Identifier (IRI) in the RDF syntax.

There are some limitations of the PGM that the TGM overcomes:

- Properties of RDF graphs support multi-value properties, whereas the PGM usually supports only single value properties [28]. The TGM, however, uses the appropriate data-type to model multi-value properties. This can be array, set, list, or bag data types.
- In RDF edges can have edges, i. e., edges hold meta-data. This is realized with (edge, predicate, object) triples. The simplest situation for this is if a predicate has a label. The TGM handles this and more complex situations by edge properties (see Figure 12 for an example) or hyper-edges, which is graphically rendered by an UML association class.
- RDF has three types of nodes (IRI, blank node, and literal) which need to be mapped to only one node type in the PGM. Because the TGM supports typed nodes any RDF node-type can be handled directly.
- The RDF model has some special semantics like reification and subclassing (rdfs:subClassOf and rdfs:subPropertyOf) that are not supported in the paper of Angles et al. [28]. The TGM also supports these special structures. This is achieved with the appropriate node or edge types.
- Each node or edge in an RDF graph contains one single IRI, a Literal or nothing (in case of a blank node), whereas each node or edge in a TGM could contain multiple properties, depending on its type. This includes multi-value properties and properties for hyper-edges as explained before.

With the TGM any kind of node type can be supported because of its capability to define user defined types for the nodes and edges. RDF allows to identify an RDF triple (statement) with an IRI. Such an aggregate construct is possible in TGM with a ternary edge type.

For a general schema mapping we have the following rules:

1) Abstract elements (subjects) are mapped to TGM nodes.
2) Aggregation (predicates) and generalization (rdfs:subClassOf, rdfs:subPropertyOf) elements are mapped to TGM edges with the appropriate type.
3) Functions (predicate cardinality is always 1) are mapped to TGM edge ends.
4) Lexical elements (literals, IRIs, ”only objects”) are mapped to TGM properties.

In order to illustrate the mapping to TGM we amend the example from Angles et al. [28] with more properties for the voc:ceo predicate. The predicate now holds the date when the CEO was appointed and the profit dependent bonus payments. For each voc:profit predicate a voc:bonus is associated. The profit is of type xsd:int and the bonus has a user defined percent data type with a range from 0 to 100. RDF auxiliary nodes (bnodes) are needed to add this information pair without any ambiguity. We need a bnode of type rdf:Alt to indicate that the bonuses are alternatives.

Figure 12 shows on the left side the amended RDFS example and the corresponding TGS on the right side. It immediately catches the eye that the TGS is far more readable for humans than the RDFS, which make it better suitable for data modeling. It is cumbersome to add properties to a predicate as it requires blank nodes (bnodes) in order to form legal and unambiguous RDF triples. For the corresponding TGM edge property we simply use an array of size 3 with a structured data type containing profit and the corresponding bonus in percent. The array *bonuses* can have a constraint XOR that clarifies the semantics as an alternative 1 out of 3. The comparison of both schemata clearly shows that even with the coloring the RDFS tends to quickly becomes confusing because properties are not aggregated to objects or predicates. Also data types are expressed on the same level as subjects, which makes the RDF less clear for data modeling purposes.

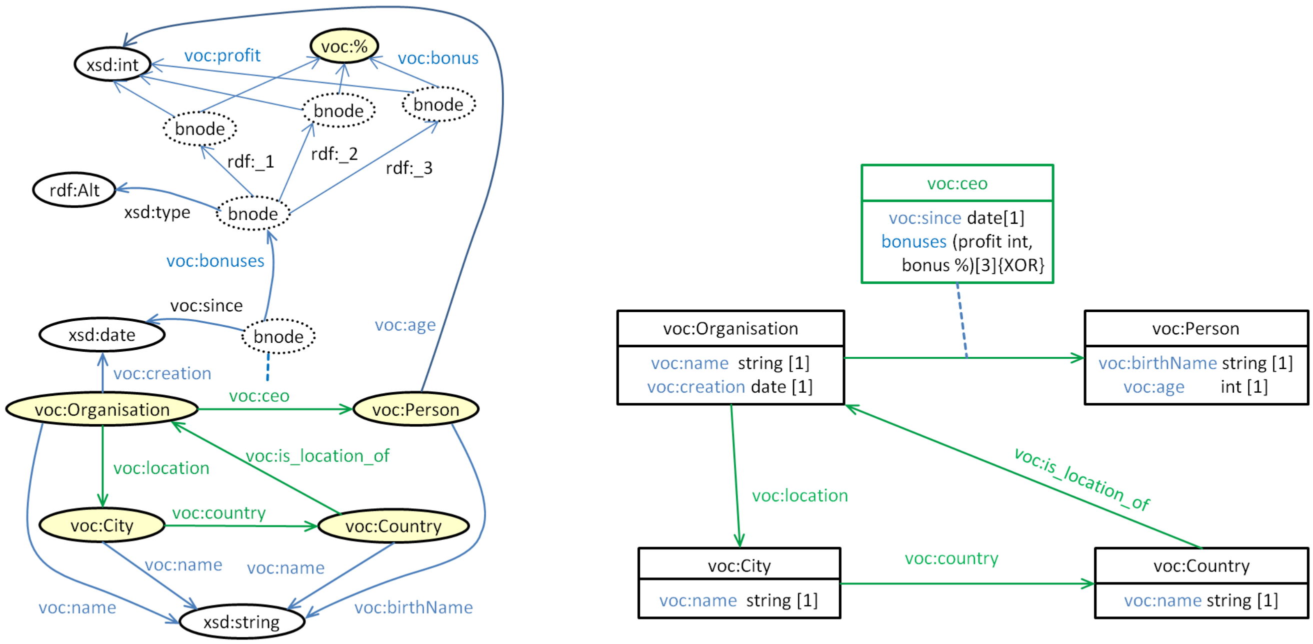


Figure 12. Example RDFS (left part) to TGS (right part) mapping with complex predicates.

VI. Conclusion and Future Work

In this paper we presented a structure definition of the TGM and an UML-like notation to visualize a graph database and its graph schema. Due to the TGS with predefined and user-defined data types, the TGM improves the formal data quality compared to other graph models. We have demonstrated the superior modeling power in comparison to other graph data models and prevalent data models, namely relational, object oriented, XML model, and RDFS. The model supports built in and user defined complex data types, which allow different abstraction levels. Another possibility for abstraction is to compress a sub-graph into a hyper-node reducing the visible complexity. This capability is especially useful for large and complex data models.

Because of its semantic modeling power the TGM could act as **supermodel** for model-management and serve as a unifying data model that supports data integration from various data sources with different data models. The main challenge for an automated data integration are incompatible data sources where the TGM as **supermodel** could help to solve quality issues and specify information preserving data translations. Details, like a score for the mapping quality and how much of the information was preserved by the mapping still need to be investigated. The development of a manipulation and query language for the TGM is future work. The idea is to combine elements of other graph languages with the dot-notation known from object-oriented languages to support navigation paths.

References

[1] F. Laux, "The Typed Graph Model", DBKDA 2020, pp. 13-19, ISBN: 978-1-61208-790-0, [Online] URL: https://www.thinkmind.org/index.php?view=article&articleid=dbkda_2020_1_30_50016 [retrieved: 2020-11-26].

[2] Neo4J - Homepage, [Online] URL: https://neo4j.com [retrieved: 2020-11-26].

[3] ArangoDB - Homepage - Graph and Beyond, [Online] URL: https://www.arangodb.com [retrieved: 2020-11-26].

[4] JanusGraph - Homepage, [Online] URL: https://janusgraph.org [retrieved: 2020-11-26].

[5] Amazon Neptune, [Online] URL: https://aws.amazon.com/de/neptune [retrieved: 2020-08-01].

[6] I. Robinson, J. Webber, and E. Eifrem, *Graph Databases*, 2^{nd} ed., O'Reilly Media, 2015.

[7] G. Daniel, G. Sunyé, and J. Cabot, "UMLtoGraphDB: Mapping Conceptual Schemas to Graph Databases", in Proceedings of Conceptual Modeling - 35^{th} International Conference ER, Gifu, Japan, pp. 430–444, 2016. [Online] URL: https://hal.archives-ouvertes.fr/hal-01344015/document [retrieved: 2020-11-26].

[8] J. Hidders, "Typing Graph-Manipulation Operations", In Proceedings of the 9^{th} International Conference on Database Theory" (ICDT), Siena, Italy, pp. 391–406, 2003.

[9] DBLP computer science bibliography, [Online] URL: https://dblp.uni-trier.de [retrieved: 2020-11-26].

[10] M. A. Rodriguez and P. Neubauer, "Construction from dots and lines", Bulletin of the American Society for Information Science and Technology, Vol. 36(No 6), pp. 35–41, ISSN:1550-8366, 2010. [Online] URL: https://arxiv.org/pdf/1006.2361.pdf [retrieved: 2020-11-26].

[11] N. Spyratos and T. Sugibuchi, "PROPER - A Graph Data Model Based on Property Graphs", ISIP 10th International Workshop, Communications in Computer and Information Science, vol.622,pp. 23–35, Springer, 2015.

[12] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang, and X. He, "Music Recommendation by Unified Hypergraph: Combining Social Media Information and Music Content", In Proceedings of the 18th International Conference on Multimedia (ACM Multimedia 2010), Firenze, Italy, pp. 391–400, 2010.

[13] A. Ghrab,O. Romero, S. Skhiri, A. Vaisman, and E. Zimányi, "GRAD: On Graph Database Modeling", Cornell University Library,arXiv:1602.00503, 2016. [Online] URL: https://arxiv.org/ftp/arxiv/papers/1602/1602.00503.pdf [retrieved: 2020-12-06].

[14] B. Amann and M. Scholl, "Gram: A Graph Data Model and Query Language", In Proceedings of the ACM Conference on Hypertext (ECHT '92), pp. 201–211, Milan, Italy, 1992.

[15] M. Gyssens, J. Paredaens, J. Van den Bussche, and D. Van Gucht,

"A Graph-Oriented Object Database Model", IEEE Transactions on Knowledge and Data Engineering, Vol 6 No 4., pp. 572–586, 1994.

[16] M. C. Pabón, C. Roncancio, and M. Millán, "Graph Data Transformations and Querying", In Proceedings of the 2014 International C* Conference on Computer Science & Software Engineering (C3S2E '14), Montreal Canada, Article No 20, pp. 1–6, 2014. [Online] URI: https://doi.org/10.1145/2641483.2641521 [retrieved: 2020-08-01].

[17] J. Pokorný, "Conceptual and Database Modelling of Graph Databases", In Proceedings of the 20^{th} International Database Engineering & Applications Symposium (IDEAS 2016), Montreal, Canada, pp. 370–377, 2016.

[18] M. Junghanns, A. Petermann, N. Teichmann, K. Gómez, E. Rahm, "Analyzing extended property graphs with Apache Flink", Proceedings of the 1^{st} ACM SIGMOD Workshop on Network Data Analytics (NDA@SIGMOD 2016), San Francisco, USA, pp. 3:1–3:8 2016.

[19] M. Levene and A. Poulovassilis, "The hypernode model and its associated query language", In Proceedings of the 5^{th} Jerusalem Conference on Information Technology, Jerusalem, pp. 520–530, 1990.

[20] R. Angles, "The Property Graph Database Model", Proceedings of the 12^{th} Alberto Mendelzon International Workshop on Foundations of Data Management, Cali, Colombia, CEUR WS Proc., 2018, [Online] URL: http://ceur-ws.org/Vol-2100/paper26.pdf [retrieved: 2020-12-06].

[21] D, Brickley and R.V. Guha (eds.), RDF Schema 1.1 W3C Recommendation, published 25 February 2014, [Online] URL: https://www.w3.org/TR/rdf-schema/ [retrieved: 2020-12-06].

[22] W3C OWL Working Group, OWL 2 Web Ontology Language Document Overview (Second Edition) W3C Recommendation, published 11 December 2012, [Online] URL: https://www.w3.org/TR/owl2-overview/ [retrieved: 2020-12-06].

[23] B. Motik, P. F. Patel-Schneider, and B. Parsia (eds.), OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition) W3C Recommendation, published 11 December 2012, [Online] URL: https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/ [retrieved: 2021-12-06].

[24] H. Chiba, R. Yamanaka, and S. Matsumoto, "G2GML: Graph to Graph Mapping Language for Bridging RDF and Property Graphs", In proceedings: J. Z. Pan et al. (eds) The Semantic Web 19th International Semantic Web Conference (ISWC 2020), pp. 160–175, Lecture Notes in Computer Science, vol 12507, Springer 2020, DOI: https://doi.org/10.1007/978-3-030-62466-8_11 [retrieved: 2020-12-06].

[25] N. N., What is openCypher? [Online] URL: https://opencypher.org/ [retrieved: 2021-12-06].

[26] A. Schätzle, M. Przyjaciel-Zablocki, T. Berberich, and G. Lausen, "S2X: Graph-parallel querying of RDF with GraphX", in Biomedical Data Management and Graph Online Querying, Big-O(Q) 2015, DMAH 2015, Lecture Notes in Computer Science, vol 9579, pp. 155–168, Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-41576-5_12.

[27] V. Nguyen, J. Leeka, O. Bodenreider, and A. Sheth, "A formal graph model for RDF and its implementation", 2016, arXiv:1606.00480. [Online]. Available: http://arxiv.org/abs/1606.00480 [retrieved: 2021-08-01].

[28] R. Angles, H. Thakkar, and D. Tomaszuk, "Mapping RDF Databases to Property Graph Databases", IEEE Access, vol. 8, pp. 86091–86110, 2020, DOI: 10.1109/ACCESS.2020.2993117 [Online] URL: https://doi.org/10.1109/ACCESS.2020.2993117 [retrieved: 2021-08-01].

[29] A. Poulovassilis and P. McBrien, "A General Formal Framework for Schema Transformation" Data & Knowledge Engineering 28(1), pp. 47–71, 1998, ISSN 0169-023X, DOI: https://doi.org/10.1016/S0169-023X(98)00013-5.

[30] P. McBrien and A. Poulovassilis, "A Uniform Approach to Inter-model Transformations", CAiSE 1999, pp. 333–348, Lecture Notes in Computer Science 1626, Springer 1999, ISBN 3-540-66157-3, DOI: https://doi.org/10.1007/3-540-48738-7_25.

[31] Ph. A. Bernstein and S. Melnik, "Model management 2.0: manipulating richer mappings", SIGMOD Conference 2007, pp. 1–12, DOI: 10.1145/1247480.1247482, [Online] URL: https://doi.org/10.1145/1247480.1247482 [retrieved: 2021-11-29].

[32] M. Lenzerini, "Data Integration: A Theoretical Perspective", ACM PODS 2002, pp. 233–246, [Online] URL: https://doi.org/10.1145/543613.543644 [retrieved: 2021-11-29].

[33] R. Hull, "Managing Semantic Heterogeneity in Databases: A Theoretical Perspective", PODS '97: Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 51–61, ACM Press, 1997, DOI: 10.1145/263661.263668.

[34] P. Atzeni and R. Torlone, "Schema Translation between Heterogeneous Data Models in a Lattice Framework", DS-6: Proceedings of the Sixth IFIP TC-2 Working Conference on Data Semantics: Database Applications Semantics, pp. 345–364, 1995.

[35] P. Atzeni and R. Torlone, "Management of Multiple Models in an Extensible Database Design Tool", EDBT 1996, pp. 79–95.

[36] P. Atzeni, P. Cappellari, and Ph. Bernstein, "Model-Independent Schema and Data Translation", EDBT 2006, pp. 368–385.

[37] P. Atzeni, P. Cappellari, and G. Gianforme, "MIDST: model independent schema and data translation", SIGMOD Conference 2007, pp. 1134–1136.

[38] P. Atzeni, L. Bellomarini, F. Bugiotti, and G. Gianforme,"A runtime approach to model-independent schema and data translation", EDBT 2009, pp. 275–286.

[39] P. Atzeni, P. Cappellari, R. Torlone, Ph. Bernstein, and G. Gianforme, "Model-independent schema translation", VLDB Journal 17(6), pp. 1347–1370, 2008.

[40] N. Roy-Hubara, L. Rokach, B. Shapira, and P. Shoval, "Modeling Graph Database Schema", IT Professional, Volume: 19, Issue: 6, pp. 34–43, IEEE publ., 2017, DOI: 10.1109/MITP.2017.4241458.

[41] J. Sequeda, M. Arenas, and D. Miranker, "On directly mapping relational databases to RDF and OWL", WWW '12: Proceedings of the 21st international conference on World Wide Web, pp. 649–658, 2012, DOI: 10.1145/2187836.2187924.

[42] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. Van de Walle, "RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data", LDOW, 23rd International World Wide Web Conference, 2014, [online] URL: http://ceur-ws.org/Vol-1184/ldow2014_paper_01.pdf [retrieved: 2021-12-06].

[43] M. Arenas, J. Pérez, J. Reutter, and C. Riveros, "Foundations of schema mapping management", ACM PODS 2010, pp. 227–238, DOI: 10.1145/1807085.1807116, [Online] URL: https://doi.org/10.1145/1807085.1807116 [retrieved 2021-11-29].

Development and Evaluation of Educational Materials on Human-Centered Design

Jun Iio
Faculty of Global Informatics,
Chuo University,
Shinjuku-ku, Tokyo 162-8478, Japan
e-mail: iiojun@tamacc.chuo-u.ac.jp

Ayano Ohsaki
Faculty of Data Science,
Musashino University,
Nishitokyo-shi, Tokyo 202-8585, Japan
e-mail: a-ohsaki@musashino-u.ac.jp

Rika Waida
CX Studio,
JVC KENWOOD Design corporation,
Setagaya-ku Tokyo 158-0097, Japan
e-mail: waida-rika@jvckenwood.com

Akira Kambayashi
Japan R&D Center,
HP Japan Inc.,
Tsurumi-ku, Yokohama, 230-0051, Japan
e-mail: kambayashi.akira@hp.com

Abstract — **Human-Centered Design (HCD) is the design principle that focuses on the users of services, systems, or products. The idea of HCD was proposed more than two decades ago, and it has been widely adopted in the information technology and design industries. However, entry-level educational materials are needed to increase the concepts among consumers and students who study engineering and industrial design. The Human-Centered Design Organization is a specific non-profit organization that promotes the idea of HCD in the Japanese industry. It has a working group whose members have been tasked with developing the required entry-level educational materials on HCD and promoting them to the industry. This paper describes some of its activities. As per the HCD cycle itself, we distinguish between the development and the promotion of the materials. In addition, we explain the relevance of the educational materials the working group delivered by illustrating the results of the questionnaire administered after the introductory seminar to those who are not familiar with HCD. Their efforts have been of great value to the people who must teach HCD concepts to newcomers.**

Keywords - human-centered design; educational materials; HCD cycle.

I. INTRODUCTION

Human-Centered Design (HCD) is the concept of a design process where the developers design their services, systems, or products focusing on their users. That is, HCD is considered a user-oriented design process. HCD was proposed more than two decades ago, and it was standardized by the International Organization for Standard (ISO) as ISO 13407 in 1999. Also, it was integrated into ISO 9241 in 2010 (ISO 9241-210:2010), adding the concept of User Experience (UX). Subsequently, it was updated to ISO 9241-210:2019 in 2019 [2].

In conventional industrial design, designers tend to focus on their design convenience. However, it often leads to the users' inconvenience. HCD solves such problems by prioritizing users' requirements rather than developers' options.

In Japan, a non-profit organization, the Human-Centered Design Organization (HCD-Net), was established in 2004 [3]. HCD-Net aims to assemble knowledge on HCD and to promote methods and skills regarding HCD. With their long-term efforts, the concept of HCD has been widely adopted among experienced engineers, especially in the information technology and design industries. However, it is still not popular among consumers. Surprisingly, even students learning engineering and industrial design are not familiar with the HCD concepts [4]. Therefore, entry-level educational materials are needed for training newcomers to perform HCD processes appropriately.

Although there are many training services, educational materials, books, and seminars for the higher-level training on HCD activities, unfortunately, we have few items that can be used as the educational material for introducing basic knowledge of HCD.

Several Working Groups (WGs) were established in HCD-Net to fill the gap between entry-level and high-level education due to the lack of educational materials. The members of these WGs have been actively working to achieve their goals. "The fostering teachers WG" was established in June 2016, and it meets monthly for face-to-face discussions. In addition to the meetings, several events have been held by the WG, and the work has been actively progressing [1], [5]-[8].

The rest of the paper is structured as follows. In Section II, we present the basic idea of the HCD process. In Section III, literature reviews are described. In Section IV, the WG's strategies are illustrated. Then, in Section V, we discuss how the HCD process worked in the WG's activities and the value of the educational materials delivered as their work. In Section VI, the evaluation of the educational materials that the WG provided is explained. Finally, conclusions, future work, and the acknowledgment close the article.

II. The HCD Process

Before explaining the WG's activity further, we describe the basic idea of the HCD process better to understand the character of the WG's work.

The HCD standard is a process standard, i.e., the standard defines several processes to realize an efficient design from a user's viewpoint. The general phases of the HCD process can be explained with the following steps: (quoted from [9]).

- *Specify context of use:* Identify who the primary users of the product, why they will use the product, what are their requirements and under what environment they will use it.
- *Specify Requirements:* Once the context is specified, it is the time to identify the granular requirements of the product. This is an important process which can further facilitate the designers to create storyboards and set important goals to make the product successful.
- *Create Design solutions and development:* Based on product goals and requirements, start an iterative process of product design and development.
- *Evaluate Product:* Product designers do usability testing to get users' feedback of the product. Product evaluation is a crucial step in product development which gives critical feedback of the product. The important point is that this cyclical process must be repeated several times to satisfy the service level of the users' requirements.

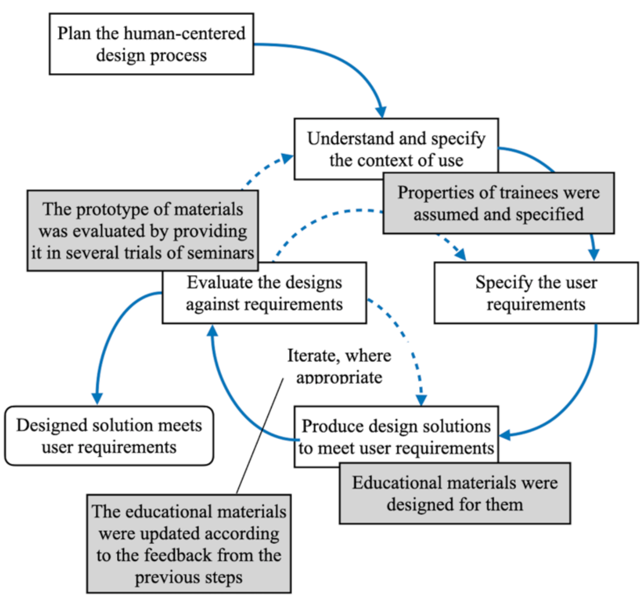


Figure 1. The WG's activities, along with the concept of HCD defined in the ISO 9241-210.

WG members oversaw creating the entry-level materials and training the trainers who can teach the basic concepts of HCD using their materials.

The fact that their activities themselves were based on the concept of HCD should be noted; that is, the designing process of their products was as follows: 1. Properties of trainees were assumed and specified. 2. Educational materials were designed for them. 3. The prototype of materials was evaluated by providing it in several trials of seminars. 4. After that, the educational materials were updated according to the feedback from the previous steps. Figure 1 illustrates the processes defined by ISO 9241-210 and the cases adopted to WG's activities in each step, respectively.

III. Literature Review

As the concept of HCD is more widely recognized, HCD education is gathering more and more interest from engineers in various fields. Instructors in this field are paying particular attention to how to teach UX concepts. Some case studies in universities and professional training colleges have been reported. However, it remains unclear how to help newcomers understand HCD in the entire business field.

Ito *et al.* [10] reported their implementation of the e-learning course on the basics of HCD. They were working for a computer-electronics manufacturer, and their e-learning course was intended to prevent miscommunication regarding user interfaces among the employees. It was an excellent example of HCD education conducted in the enterprise.

Gonzalez *et al.* [11] surveyed 140 students of the Human Factors and Ergonomics Society (HFES) and analyzed 40 UX job postings. The results show a discrepancy between the skills the UX industry expects students to have and the skills HFES promotes for a career in UX. They recommended a focus on increasing HFES's relevance to students interested in future UX careers. Vorvoreanu *et al.* [12] also reported on the UX education for undergraduate students at university.

The concept of "design thinking" is a similar idea to UX design. Wrigley *et al.* [13] focused on surveying the design thinking education provided as Massive Open Online Courses (MOOCs). MOOCs are open to the public, and most can be participated in for free via the Internet. Therefore, anyone who wants to learn about design thinking can acquire knowledge by accessing the courses presented on their websites.

Dirin and Nieminen [14] studied the relevance of User-Centered Design (UCD) education to a mobile application development course implemented in a university. They analyzed the feedback from students and concluded that UCD education had a significant role in the development and improvement of students' capabilities on consulting and user study research.

We can find many other cases where HCD or UCD processes were introduced to education programs in various fields; Adam *et al.* [15], Organ *et al.* [16], and Carter *et al.* [17] reported cases in health and medical education, Harvey *et al.* [18] reported an issue in fashion education, Wilson *et al.* [19] discussed the possibility of applying the UCD approach to the training environment for aircraft maintenance personnel. Bowie and Cassim [20] argued for the HCD methodology in contemporary communication design education. These papers reveal the presence of a potential need for HCD education in various domains.

Additionally, there are some studies on designing or evaluating a curriculum by incorporating HCD processes similar to our approach in creating their educational materials. Altay [21] pointed out a similarity between the learner-centered approach in education and the user-centered approach in design disciplines. Altay illustrates this by adopting a user-centered approach within the human factors course as learner-centered instructional methods. Reich-Stiebert *et al.* [22] explored robot design education employing the HCD approach. They investigated students' preferences regarding the design of educational robots and evaluated the course according to the results. Chen *et al.* [23] reported on evaluating the curriculum using a method of creating student personas in the field of resource engineering education.

IV. The WG's Strategies

The WG's activities' starting point was the textbook published as the first of the HCD book series. Based on the contents of the book, the WG considered two strategies; one was to develop presentation slides and guidebooks as the educational materials, and another was to foster trainers who could provide seminars to newcomers who were not familiar with HCD.

Under these strategies, the WG created two prototypes of the educational materials for engineers and salespeople. Furthermore, some simulated seminars were conducted to acquire feedback and opinions to brush up on the materials.

A. Educational Materials for Engineers

The first target was newbie engineers who were not familiar with the concept of HCD. The WG published a beta-version of the presentation slides in June 2017, after several discussions by the WG members. After collecting some feedback, the materials were updated, and version 1.0 of the document was published in May 2018.

The presentation materials have 42 slides, which are intended for conducting a seminar of approximately one-and-a-half hours. An overview of the contents is as follows:

- Case studies
- The concept of human-centered design
- Usability
- Introducing the HCD cycle
- Appendix (good practices)

Figure 2 shows some examples of the document. The upper left of the figure is the cover page, the upper right shows a sample from the case studies, the lower-left illustrates the HCD cycle, and the lower right is the cover of the appendix.

As seen from the small icon at the corner of the cover page (see the upper left of Figure 2, the materials are published under the license of Creative Commons (CC BY-NC-SA 3.0). Therefore, everyone can share, redistribute, modify, and create deliverables based on this product, if they follow the conditions defined by the CC license. This licensing strategy is beneficial for future trainers, whom the WG also wants to encourage because those educators can modify educational materials as they like.

Figure 2. Examples of the presentation slide for training engineers.

B. Educational Materials for Salespeople

After creating the entry-level educational materials on HCD for engineers, the WG started a discussion on another version of the educational materials. The members of the WG considered that the people in the front office who had contact with their customers had to know the HCD concepts. Especially in the case of business to business (B2B), such businesses require customers' understanding and cooperation. Hence, the WG decided on salespeople as the next target for education on the idea of HCD.

At the beginning of the preparation work, the WG invited some salespeople and producers who were using HCD processes and worked directly with their customers in their daily business activities. The WG members had several interviews to learn about their thought processes, how they worked with their customers, etc. Also, they invited salespeople who did not know the HCD to attend an entry-level HCD lecture so that the discussions could be fruitful for both sides.

Although the base materials were those for engineers, minor modifications were made to the original ones. There were two significant changes; one is that the thoughts of the customer-orientation investigation were introduced instead of the case studies. The other was that the discussion on the positioning of the HCD was added before the conclusions. The latter part also mentions User Experiences (UX) because UX is also a key topic for discussing HCD-related issues with customers.

The overview of the contents for salespeople is as follows:

- Considering the view of the customer-oriented
- The concept of human-centered design
- Usability
- Introducing the HCD cycle
- Positioning of the HCD
- Appendix (good practices)

The education materials of the HCD for salespeople were released in May 2019 (version 1.0).

C. Guidebooks

In addition to providing the presentation slides, the WG also supplies a guidebook on how to conduct efficient training on HCD. Generally, it is not easy to run seminars along with the presentation materials when they were created by other individuals. Therefore, guidebooks for training courses for trainers using two versions (for engineers and salespeople) of educational materials are also provided.

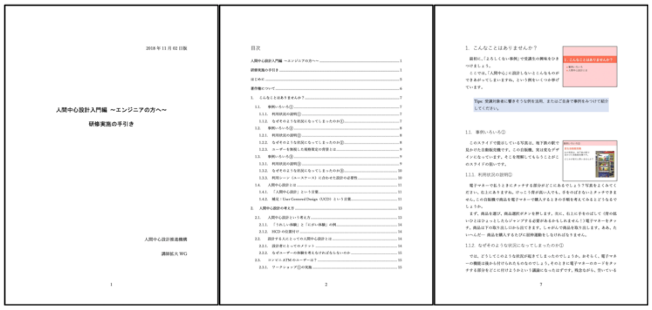


Figure 3. Examples of the presentation slide for training engineers.

Figure 3 shows some examples of the guidebook for the educational materials for training engineers. The left figure is the cover page, the middle shows the table of contents, and the right shows one of the instructional pages.

As shown on the right in Figure 3, the instructions are described for all presentation slides. The guidebook helps novice trainers by giving some additional information on how to teach the topics, etc.

All the educational materials (presentation slides) and complementary guidebooks are uploaded to the server hosted by HCD-Net. These can be downloaded from [24] (for engineers) and [25] (for salespeople).

D. Simulated Seminars (Trial Events)

To evaluate the prototype of the educational materials and lectures, the WG held five simulated seminars. Table 1 shows a list of trial events officially announced by HCD-Net.

TABLE I. THE LIST OF TRIAL SEMINARS

ID	Target	Version	Date	Participants
1	Engineer	Alpha	Mar 4, 2017	18 pros and beginners
2	Engineer	Beta	Aug 29, 2017	10 pros and beginners
3	Engineer	Ver. 1.0	May 25, 2018	21 pros and beginners
4	Salesperson	Beta	Jan 19, 2019	18 pros and beginners
5	Engineers & Salesperson	Mofidied	Dec 19, 2019	26 (mainly) beginners

The target of the first three seminars (ID 1, 2, 3) was the version for engineers. Lectures based on alpha, beta, and version 1.0 were examined in each trial, respectively. The next one (ID 4) was for salespeople. At that time, the beta version of lectures for salespeople was confirmed. We asked for HCD professionals to participate in those trials (from ID 1 to ID 4). Therefore, we could hear various opinions from not only beginners but also from professionals. Furthermore, the trial seminars were helpful for those professionals because most of them oversaw human resource development. They were motivated to take the contents of the entry-level workshops back to their companies.

V. DISCUSSIONS

As described in the previous section, the WG created the educational materials and guidebooks according to the HCD processes. This section discusses the compliance with such methods and the values of the WG's products.

A. How the HCD Process Worked in the WG's Activity

Looking back, in this section, we consider how the four steps in the HCD cycle were applied to the WG's activities. We recall that the HCD cycle has four phases: specifying the context of use, specifying requirements, creating design solutions and development, and evaluating products.

1) Specify Context of Use: As we described in the introduction of this paper, our study aimed to create entry-level educational materials and encourage the instructors who present the training in their organizations. Considering the situations and experiences of each WG member [5][6], the WG decided on engineers as the first target group of trainees and then salespeople as the second target group of trainees.

2) Specify Requirements: Because the educational materials are designed to accompany the textbook's explanations, the critical part of the WG's work was to decide which components should be selected. Furthermore, the course time was considered very short. At the beginning of the WG's discussion, it assumed that the entry-level education on HCD would be conducted in one or two hours. Therefore, the members tried not to make the contents of the materials too complicated. Also, the members discussed what the participants of the lectures would consider necessary for their studies and their future careers. That is one of the essential points of the WG's activity in the view of the HCD concept.

3) Create Design Solutions and Development: The WG's process for making the materials was iterative, requiring at least two cycles.

The first cycle started with the prototype of the educational materials for engineers. The product was firstly published in its beta version. The WG then collected feedback and comments at the trial seminars (see Section 4.4). After that, the materials were published as version 1.0, and currently, it has been updated to version 1.1.

The second cycle was based on the first one. The prototype of materials for salespeople was started from the latest version for engineers and then updated according to the WG's interviews and feedback from trial seminars. It was published as the beta version, and updated to version 1.0, as well.

4) Evaluate Product: Evaluation by the potential users is a critical process in the HCD cycle. In the WG's activities, the members also considered it the principal procedure. As described previously, the WG remained focused on the evaluation process with the trial events run to evaluate the materials during the design phase.

The WG's leading work in 2019 was the evaluation, improvement, and investigation of which organizations were utilizing their materials. Several new members, who were users of the materials, joined the WG in 2019. The educational materials and the guidebooks were updated according to the interviews conducted with them and feedback from questionnaires.

B. *Value of the Educational Materials*

The WG aimed to prepare entry-level educational materials and foster instructors who can teach newcomers unfamiliar with the concept of HCD. Therefore, it tried to fill the gap between newbies and experienced engineers, designers, and salespeople by providing the materials.

A review from Amazon's sales listing of the textbook states that: "It is not easy to understand only by reading. It would be worth reading if some lectures were provided using this book as its textbook." We had to agree with this comment. Hence, our decision to provide lectures on entry-level HCD knowledge with these materials and guidebooks.

During the work conducted in 2019, the WG collected several opinions and impressions of the products from the new members. All of them mentioned that it was helpful, but there was still some room for improvement. As the materials were provided under the CC license, the users could modify the contents to become suitable for their courses.

VI. Evaluations

We conducted a seminar that introduced the concept of HCD to the office staff who were unfamiliar with it to evaluate the education materials. To perform a subjective evaluation, we asked the participants to fill a questionnaire after the seminar.

The 56 participants answered the following questions.

- How to create awareness for the seminar, their profile, and the knowledge levels of the HCD before participating in the workshop.
- How much did you understand the concept of the HCD?
- How satisfied are you with the seminar?
- The consciousness to learn the HCD more deeply.

A. *The Way to Find the Seminar*

Figure 4 shows the way to find the seminar. It tells us that the HCD-Net actively reached the candidates who should participate in the workshop. Viral communications (friends' comments in the several SNSs, recommendations from friends and colleagues) are unignorable to disseminate holding events.

As it was an introductory course, it was necessary to disseminate broadly, so "word-of-mouth," such as recommendations from companies and acquaintances, should be used.

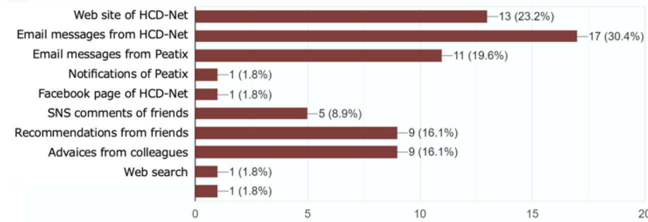
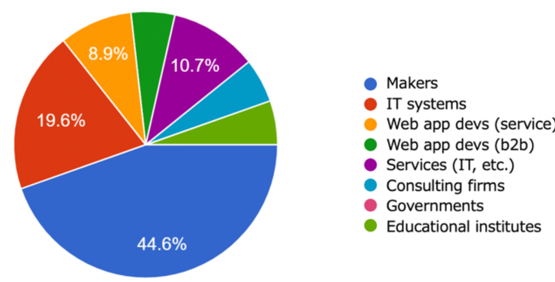


Figure 4. The way to find the seminar.

B. *The Properties of the Participants*

Figures 5 and 6 illustrate the characteristics of the participants of the seminar. The former shows the occupations, and the latter shows the frequency distributions of their generation, which indicates that all working ages are interested in the HCD seminar.



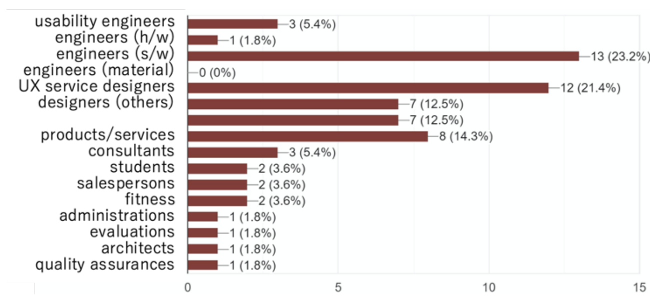


Figure 5. The occupations of the participants.

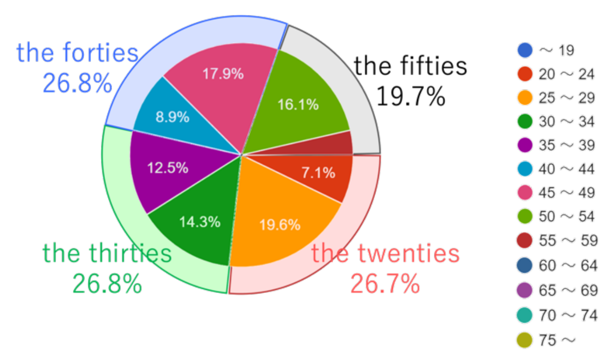


Figure 6. The frequency distribution of participants' generation.

C. Knowledge Levels of the HCD before Participating in the Seminar

The participants' knowledge levels are shown in Figure 7. Approximately one-third, i.e., 32.2% of participants, were estimated as beginners—the seminar's primary target. Further, 26.7% of participants were practitioners and not directly targeted at this introductory course. However, the results imply that they were interested in online seminars and teaching methods, which would also be meaningful.

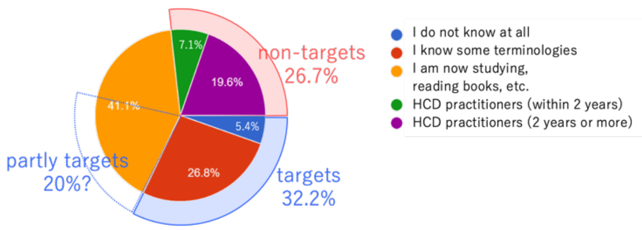


Figure 7. The knowledge levels of the HCD before participating in the seminar.

D. How Much Did You Understand the Concept of the HCD?

Figure 8 illustrates the participants' understanding levels by the categories: the basic knowledge of HCD, users, usability, HCD cycle, and the relation among HCD, design thinking, and UX.

For the basic concept of teaching the knowledge of HCD, the seminar should be designed to present HCD issues comprehensively; the results shown in Figure 8 imply that it achieved the goal successfully.

In addition, the answers to the question asking did you understand the relation between HCD, design thinking, and UX showed an excellent score. It was considered that the seminar and educational materials focused on the issues and that it was effective.

E. How Much Did You Satisfy the Seminar?

The satisfaction levels of participants by categories are indicated in Figure 9. Apart from the mini work, almost all factors affected the impression of the participants. Significantly, the fact that they were satisfied with the seminar structure and its presentation materials indicates the WG's deliverables were meaningful and relevant to teach the HCD concept to beginners.

F. The Consciousness to Learn the HCD More Deeply

Figure 10 shows the participants' awareness to learn HCD in depth. The answers varied from level one—which meant they did not have any motivation to learn more—to level five—who had some zeal to learn HCD in depth.

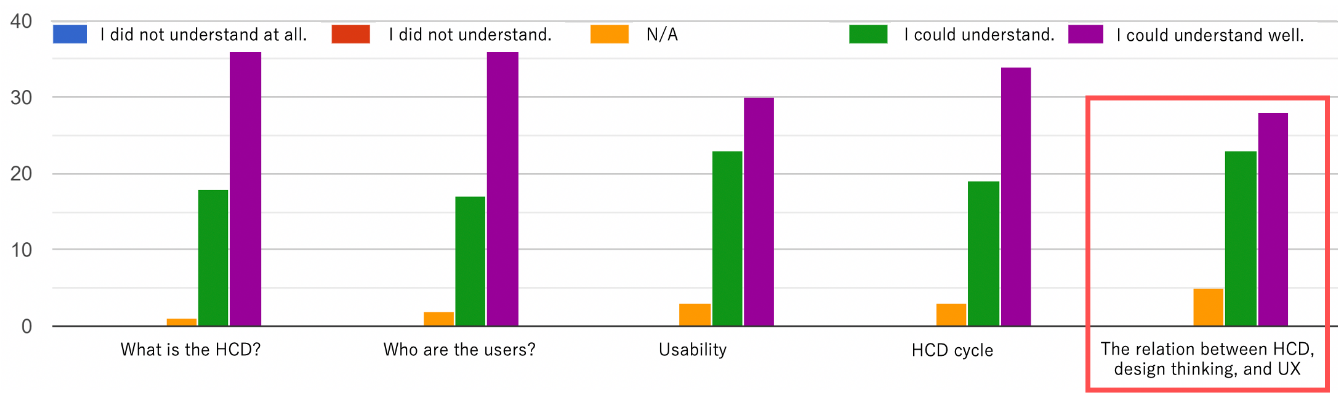


Figure 8. The understanding levels of the participants.

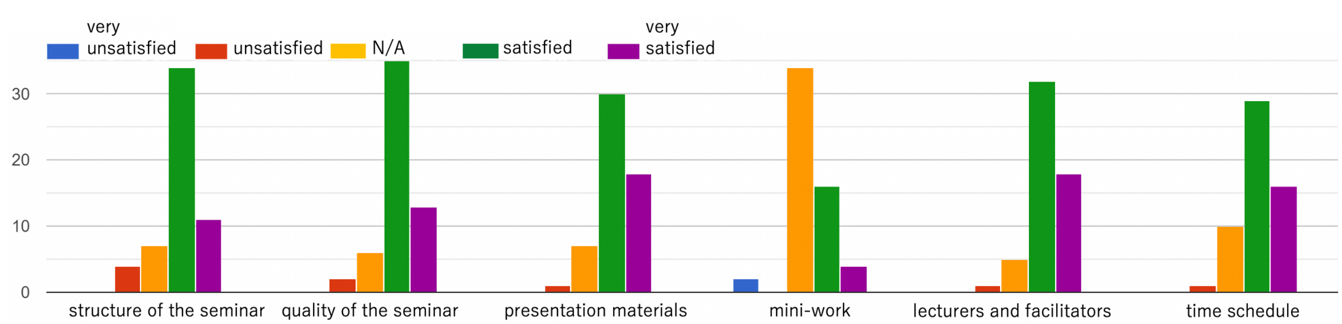


Figure 9. The satisfaction levels of the participants.

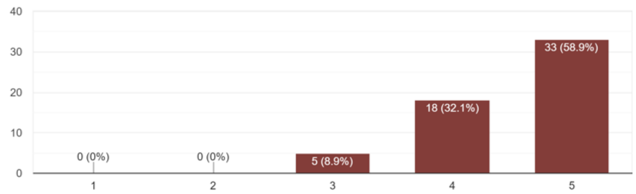


Figure 10. Do you want to learn more about HCD? (from 1 not at all, to 5 very likely)

Figure 10 indicates that the introductory seminar to the HCD was completed, and the participants got a positive impression to learn something deeper about the HCD issues. It also implied that the educational materials the WG delivered had some values to promote the relevance of the concept of the HCD.

VII. Conclusions and Future Work

The human resource development WG members, which was set up in HCD-Net to implement the entry-level educational materials on HCD and foster lecturers who can train newcomers in each organization, have been working actively during recent years. This paper described their activities and provided an overview of their results. The most significant feature of their work was that their outcomes, that is, the HCD educational materials themselves, were designed according to HCD processes.

The educational materials they created are intended for two different target groups: engineers and salespeople. Firstly, the training materials for engineers were designed. After that, based on the prototype, the revised ones for salespeople were created. Guidebooks for conducting the training were also created to accompany the educational materials to make it easier for the lecturers to present them.

The WG's main activities in 2019 were conducting interviews with the users, delivering questionnaires to them, and improving the educational materials according to the feedback, as described in the last part of Section 4.1. However, a more in-depth analysis of the feedbacks remains to be done as part of their future work.

Several evaluations were performed as part of the HCD cycles in the development processes. We conducted a series of simulated seminars and interviews with users of the prototype versions. After the first publication of those materials, an introductory workshop was held for those unfamiliar with the concept of the HCD. At the end of the seminar, we asked the participants to fill the questionnaire. We analyzed the answers subjectively to evaluate and validate the relevance of the educational materials.

However, the lectures using these educational materials should be more widespread if we let the HCD concepts penetrate all the industries. More and more promotions will be needed, and they remain our future work. Furthermore, more evaluations to improve educational materials should be conducted. It will be an ongoing task as part of the WG's future activities.

Acknowledgment

The authors thank all WG members and the participants of trial seminars and evaluations, the board members of the HCD-Net, and collaborators for the activities concerning the development of the education materials.

References

[1] J. Iio, A. Osaki, and R. Waida, “Development and Promotion of Educational Materials on Human-Centered Design, ” *The Thirteenth International Conference on Advances in Computer-Human Interactions (ACHI2020),* pp. 447-451, Valencia, Spain & Online, 2020.

[2] “ISO 9241-210:2019 Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems,” https://www.iso.org/standard/77520.html,[retrieved: 19th Nov 2020]

[3] Z. Liu, “User Experience in Asia,” *Journal of Usability Studies,* vol. 9, no. 2, pp. 42–50, 2014.

[4] B. B. Hong, E. Bohemia, R. Neubauer, and L. Santamaria, “Design for Users: The Global Studio,” *The 20th International Conference on Engineering and Product Design Education (E&PDE18),* London, UK, 6–7th September 2018.

[5] H. Yasu *et al.,* “Framework of Education to Promote HCD among Organizations, — Learning from Case Studies —,” *Bulletin of Human Centered Design Organization,* vol. 12, no. 1, pp. 13–19, 2016.

[6] H. Yasu *et al.,* “Framework of Education to Promote HCD among Organizations — Proposal and Evaluation on Action Plans for Each Trainee —,” *Bulletin of Human Centered Design Organization,* vol. 13, no. 1, pp. 19–24, 2017.

[7] R. Waida *et al.,* “Teaching Materials of HCD Introductory Course for Practitioners – Activities of Making the Beta Version,” *Bulletin of Human Centered Design Organization,* vol. 14, no. 1, pp. 24–28, 2018.

[8] A. Kambayashi *et al.,* “Teaching Materials of HCD Introductory Course for Practitioners: Activities of Official Version for Engineers and the Beta Version for People in Contact with Customers,” *Bulletin of Human Centered Design Organization,* vol. 15, no. 1, pp. 9–14, 2019.

[9] “User-centered design,” From Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/User-centered_design, [retrieved: 19th Nov 2021]

[10] J. Ito, A. Ikegami, and T. Hirayama, “Practice of Promoting HCD, Education by a Consumer Electronics Manufacturer,” *M. Kurosu (Ed.): Human Centered Design, HCII 2009,* LNCS 5619, pp. 594–600, 2009.

[11] C. A. Gonzalez, M. Ghazizadeh, and M. Smith, “Perspectives on the Training of Human Factors Students for the User Experience Industry,” *Proceedings of the Human Factors and Ergonomics Society Annual Meeting,* vol. 58, no. 1, pp. 1807–1811, 2014.

[12] M. Vorvoreanu, C. M. Gray, P. Parsons, and N. Rasche, “Advancing UX Education: A Model for Integrated Studio Pedagogy,” *Proceedings of the Computer Human Interaction, CHI 2017,* May 6–11, Denver, CO, USA, 2017, DOI: http://dx.doi.org/10.1145/3025453.3025726

[13] C. Wrigley, G. Mosely, and M. Tomitsch, “Design Thinking Education: A Comparison of Massive Open Online Courses,” *The Journal of Design, Economics, and Innovation,* vol. 4, no. 3, pp. 275–292, 2018.

[14] A. Dirin and M. Nieminen, “Relevance of UCD Education to Software Development – *Recommendation for Curriculum Design.” Proceedings of the 8th International Conference on Computer Supported Education (CSEDU 2016),* no. 2, pp. 112–120, 2016.

[15] M. Adam, S. A. McMahon, C. Prober, and T. Bärnighausen, "Human-Centered Design of Video-Based Health Education: An Iterative, Collaborative, Community-Based Approach," *Journal of Medical Internet Research,* vol. 21, no. 1:e12128, 2019. DOI: 10.2196/12128

[16] D. Organ et al., "A systematic review of user-centred design practices in illicit substance use interventions for higher education students," *European Conference on Information Systems 2018: Beyond Digitization – Facets of Socio-Technical Change,* Portsmouth, UK, 23–28 June. 2018.

[17] J. Carter, Y. J. Bababekov, and M. D. Majmudar, "Training for our digital future: a human-centered design approach to graduate medical education for aspiring clinician-innovators." *npj Digital Medicine,* vol. 1 no. 1, 2018. https://doi.org/10.1038/s41746-018-0034-4

[18] N. Harvey, P. Ankiew, and F. van As, "Fashion design education: effects of users as design core and inspirational source." *Proceedings PATT 37: Developing a knowledge economy through technology and engineering education,* pp. 203–211, Malta, 3–6 June. 2019.

[19] C. Wilson, W. Bennett Jr., S. Guarino, K. Bove, and T. L. Cain, "Applying a User-Centered Design Approach to Developing Game-Based Training Environments for Aircraft Maintainers," *End-User Considerations in Educational Technology Design,* pp. 217–238, 2018. DOI: 10.4018/978-1-5225-2639-1.ch01

[20] A. Bowie and F. Cassim, "Linking classroom and community: A theoretical alignment of service learning and a human-centered design methodology in contemporary communication design education," *Education as Change,* vol. 20, no. 1, pp. 126–148, 2016.

[21] B. Altay, "User-centered design through learner-centered instruction," *Teaching in Higher Education,* vol. 19, no. 2, pp. 138–155, 2014. DOI: 10.1080/13562517.2013.827646

[22] N. Reich-Stiebert, F. Eyssel, and C. Hohnemann, "Exploring University Students' Preferences for Educational Robot Design by Means of a User-Centered Design Approach," *International Journal of Social Robotics,* 2019. https://doi.org/10.1007/s12369-019-00554-7

[23] K. C. Chen et al., "Creating a Project-based Curriculum in Materials Engineering," *Journal of Materials Education,* vol. 31, no. 2, pp. 37– 44, 2009.

[24] Human Centered Design Organization, "HCD training materials for engineers ver. 1.1," https://www.hcdnet.org/hcd/column/materials01/hcd-1177.html [retrieved: 19th Nov 2021]

[25] Human Centered Design Organization, "HCD training materials for salespeople," https://www.hcdnet.org/hcd/column/materials01/hcd-1307.html [retrieved: 19th Nov 2021]

Towards Expanding Conceptual-Level Inheritance Patterns for Evolvable Systems

Marek Suchánek
Faculty of Information Technology
Czech Technical University in Prague
Prague, Czech Republic
ORCID: 0000-0001-7525-9218
marek.suchanek@fit.cvut.cz

Robert Pergl
Faculty of Information Technology
Czech Technical University in Prague
Prague, Czech Republic
ORCID: 0000-0003-2980-4400
robert.pergl@fit.cvut.cz

***Abstract*—Inheritance as a relation for expressing generalisations and specialisations or taxonomies is natural for human perception and essential for conceptual modelling. However, it causes evolvability problems in software implementations. Each inheritance relation represents a tight coupling between entities with uncontrolled propagation of changes. Such a coupling leads to a combinatorial effect or even a combinatorial explosion in the case of complex hierarchies with multiple inheritances. This extended paper uses our analysis of multiple inheritance and method resolution order in the Python programming language to design and demonstrate code generation techniques with the use of inheritance patterns. The analysis is based on the design of inheritance implementation patterns from our previous work. Our design shows how can be conceptual-level inheritance implemented efficiently in software systems. The proposed design is demonstrated with Python (sometimes called "executable pseudocode"). However, it can be implemented with other object-oriented programming languages, even those that do not support multiple inheritance. The resulting design and prototype of expanders for inheritance patterns are ready for application in practice and further use within different technology stacks.**

***Index Terms*—Multiple Inheritance; Python 3; Evolvability; Method Resolution Order; Composition Over Inheritance.**

I. Introduction

This paper extends the previous conference paper [1] where inheritance in Python has been analysed, and conceptual-level inheritance patterns implementation in Python has been introduced. It provides more detailed information about the design and implementation by both adding new sections and extending the existing ones. Moreover, it includes code generation design and prototype previously outlined as future work.

Inheritance in the software engineering discipline is a commonly used technique for both system analysis and software development. During analysis, where we want to capture a specific domain, inheritance serves for refining more generic entities into more specific ones, e.g., an employee as a specialisation of a person. It is natural to have multiple inheritance, e.g., a wooden chair is a seatable physical object, furniture, and flammable object. On the other hand, in Object-Oriented Programming (OOP) software implementations, inheritance is typically used or even misused to re-use methods and attributes purely. In OOP, it causes so-called ripple effects violating evolvability of software [2]–[4].

The Python programming language is the most popular general-purpose and multi-paradigm programming language [5]. It is dynamically typed but allows multiple inheritance and also type hints [6]. Sometimes, Python is also called "executable pseudocode" thanks to its easy-to-read indentation-based syntax and versatility [7]. It is widely used for prototyping due to its flexibility and then rewritten into enterprise-ready implementation (e.g., using Java). For the implementation of conceptual-level inheritance in OOP, there are several patterns suggested in our previous work [3]. Although the patterns are compared and evaluated, implementation examples and empirical proofs are not yet provided.

Whenever there is a repeated pattern in software implementation, it strives for either re-use via generalisation, code generation or a combination of both. The code generation techniques instantiate patterns using code templates. The produces code can be then used in a more extensive code base and further customised. Normalised Systems use a code generation technique called expansions (the templates are called "Expanders"). One of the core advantages is the produces code's evolvability as it supports extensions using features and custom code insertions. Moreover, the design and provided tooling is verified by large-scale and real-world use cases. We strive to implement Expanders for the proposed inheritance patterns in Python. It should prove that the patterns can be used to implement conceptual-level inheritance without combinatorial effects with utilizing code generation.

In this extended paper, we design the conceptual-level inheritance patterns in Python using its specific constructs to allow easy use of the patterns instead of traditional OOP inheritance that causes ripple effects. The prototype implementation is designed to compare and demonstrate the additional complexity versus complexity caused by combinatorial effects. Then, with the patterns prepared, we design and prototype the expanders. First, Section II outlines the Design Science Research methodology that is followed within this research. Section III acquaints the reader with terminology and the overall context. In Section IV, we analyse how traditional inheritance works in Python and then describe the design and implementation of each pattern. Subsequently, Section V describes the patterns expansion design, and Section VI briefly describes the implemented prototype and results. Finally, Section VII summarises and evaluates the expanders in contrast to traditional inheritance, describes the new experience with developing expanders for Python, and suggests future research.

II. Methodology

This research follows the Design Science Research (DSR) methodology [8]. It designs the artefacts based on a three-cycle view [9]. The knowledge base represents the NS theory and expanders, previous work on inheritance, and conceptual-level and software-level inheritance. On the other side, the environment is the Normalized Systems development, where we want to deliver the artefacts to allow modelling with inheritance. However, without causing combinatorial effects in the implementation – those are the requirements according to DSR.

The artefacts need to be evaluated and improved in design cycle iterations. The evaluation considers the measure of eliminations of combinatorial effects as well as usability (clarity, versatility, and extensibility). In this paper, we present the final artefacts of this procedure; however, we also mention key improvements done during the iterations.

We set the following research goals that the designed artefact will address:

- G1 – Design and implementation of the conceptual-level inheritance patterns [3].
- G2 – Design a code generation for the proposed implementation.
- G3 – Prove that the designed code generation is feasible and versatile (i.e., can produce usable source code with the possibility to extend it).

Whereas G1 and G2 can be considered as traditional design artefacts, G3 is related to evaluation and actually demonstrating the value of artefacts supporting G1 and G2. Although we use Python in our research, the design must be done language-independent. If someone designs implementation of the patterns in other languages following our steps, the same outcomes should be achieved.

The structure of this paper follows the workflow for achieving the set goals in the natural order. First, we design the known patterns for implementation for G1. That gives us reference implementations that we strive to code-generate for G2, i.e., remove the burden of manual coding of complex structures for implementation where conceptual-level inheritance is natural and straightforward. Finally, with a designed code generation technique, we can prove and demonstrate the feasibility and versatility for achieving G3.

III. Related Work and Terminology

This section briefly introduces the related research and terminology required for our approach and provides an overview of the knowledge base in terms of DSR. It focuses on Normalized Systems and Expanders, inheritance on conceptual-level, and Python programming language that we selected for prototyping due to its versatility and other capabilities..

A. Normalized Systems Theory

Normalized Systems Theory [2] (NST) explains how to design systems that are evolvable using the fine-grained modular structure and elimination of combinatorial effects, i.e., size of change impact is proportional to the size of the system. The book [2] also describes how to build such software systems based on four elementary principles: Separation of Concerns, Data Version Transparency, Action Version Transparency, and Separation of States. Violation of any of these principles leads to combinatorial effects. A code generation techniques producing skeletons from the NS model and custom code fragments are applied to make the development of evolvable information systems possible and efficient.

The theory [2] states that the traditional OOP inheritance inherently causes combinatorial effects. Without multiple inheritance, it even leads to the so-called "combinatorial explosion", as you need a new class for each and every combination of all related classes to make an instance that inherits different things from multiple classes, e.g., a class `JuniorBelgianEmployeeInsuredPerson`. But even with multiple inheritance, the generalisation/specialisation relation is special and carries potential obstacles to evolvability. First, the coupling between subclasses and superclasses with the propagation of non-private attributes and methods is evident. Also, persisting the objects in traditional databases is challenging [2] [3] [10].

B. Normalized Systems Expanders

The code templates, together with mapping from NS models, are called Expanders. It allows the generation of any textual files from NS models. For software development, expanders to produce enterprise information systems in Java are used. However, one can also develop expanders to produce technical documentation of the system or graphical representation using SVG. The expanded code base is expected to be enhanced by adding custom code fragments called "craftings" to implement functionality that cannot capture using elements [11].

To avoid overwriting a system's craftings upon re-generation (or so-called rejuvenation), they should be harvested. The harvesting procedure basically goes through the code base and stores both insertions and extensions in the designed location. Then, when a system is rejuvenated, it takes the selected expanders, NS model of Elements, technical details and harvested craftings to generate the codebase. There might be several reasons for re-generation: a change in the model, updated expanders, or a different underlying framework (cross-cutting concern) [2], [11], [12].

Finally, the expanders can also be variable using *features*. It aims to include pieces of code to multiple expanders. Then, a feature can be enabled by an option from the NS model (e.g., by specifying a particular data option for a data element) or another condition. It typically adds some logic to multiple artefacts at once to extract cross-cutting concerns from the expanders (decoupling). As expanders are very variable and can become complex, there is also a prepared way to test them easily [11].

C. Conceptual-Level Inheritance

Inheritance in terms of generalisation and specialisation relation is ontologically aligned with real-world modelling. It is tightly related to ontological refinements, where some concept is further specified in higher detail. It forms a taxonomy – a classification of things or concepts. For example, an employee is a special type of a person, or every bird is an animal. In conceptual modelling, inheritance is widely used to capture taxonomies and refine concepts under certain conditions. Although it is named differently in various languages, e.g., is-a hierarchy, generalisation, inheritance, all usually work with the ontological refinements. As shown in [13] different views on inheritance can be made with respect to implementation, where it can be (mis)used for reuse of classes without a relevant conceptual sense [3] [14].

D. Object-Oriented Programming and Inheritance

When talking about inheritance in OOP, it is crucial to distinguish between class-based and prototype-based style. In prototype-based languages, objects inherit directly from other objects through a prototype property. Basically, it is based on cloning and refining objects using specially prepared objects called prototypes [15]. On the other hand, a more traditional and widespread class-based programming creates a new object through a class's constructor function that reserves memory for the object and eventually initialises its attributes. In both cases, inheritance is used for polymorphism by substituting superclass instance by subclass instance with eventually different behaviour [13] [16].

Both single and multiple inheritance can be used for reuse of source code. In [13], a clear explanation between essential and accidental (i.e., purely for reuse) use of inheritance is made. Moreover, [17] shows how multiple inheritance leverages reuse of code in OOP, including its consequences. According to [18], Python programs use widely (multiple) inheritance and it is comparable to use of inheritance in Java programs of the similar sample set.

E. The Python Programming Language

Python is a high-level and general-purpose programming language that supports (among others) the object-oriented paradigm with multiple inheritance. It allows redefinition of almost all language constructs including operators, implicit conversions, class declarations, or descriptors for accessing and changing attributes of objects and classes. Both methods and constants for such redefinitions start and end with a double underscore and are commonly called "magic", e.g., magic method `__add__` for addition operator. The syntax is clean as it uses indentation for code blocks and limits the use of brackets. Python can be used for all kinds of application from simple utilities and microservices to complex web application and data analysis [6] [18].

Often, Python is used for prototyping, and then the production-ready system is built in different technologies such as Java EE or .NET for enterprise applications and C/C++ or Rust for space/time optimisation. Another essential aspect that makes Python a suitable language for prototyping is its dynamic type system that allows duck typing [19], however static typing is supported using annotations since version 3.5. A Python application can be then checked using type checkers or linters similarly to compilers, while preserving a flexibility of dynamic typing [6] [20].

"The diamond problem" related to multiple inheritance is solved using "Method Resolution Order" (MRO) that is based on the C3 superclass linearisation algorithm. Normally, a class in Python has method `mro` that lists the linearised superclasses. It can also be redefined using metaclasses, i.e., classes that have classes as its instances. By default, a class is a subclass of class `object` and an instance of metaclass `type`. Class `object` has no superclass and it is an instance of `type`. Class `type` is a subclass of `object` and is an instance of itself [6] [16].

IV. Python Inheritance Analysis

In this section, we analyse how conceptual-level inheritance implementation patterns proposed in [3] can be used in Python. We discuss the implementation options with respect to evolvability and ease of use, i.e., the impact of the pattern on the potential code base. The proposed implementation of the patterns fulfils our goal G1.

For demonstration, we use a conceptual model depicted in Figure 1 using OntoUML [14]. We use monospaced names of class and object names in the following text, e.g., `Person`. We strive to design the implementation of such a model where object `marek` is an instance of multiple classes with minimal development effort but also minimal combinatorial effects. Our model also contains the potential diamond problem, i.e., class `AlivePerson` inherits from `Locatable` via `Insurable` but also via `LivingBeing` and `Person`. Overriding is also included using derived attributes, as we avoid methods for the sake of clarity; however, it would work equivalently.

Note that Figure 1 is a conceptual model and not an implementation model. For example, one may object that Man and Woman can be just an attribute and enumeration. That would be true only for traditional programming. In the conceptual model, that approach is not correct as Man and Woman are types of Person, not property. In programming, when considering evolvability, the reason is that enumeration does not allow evolution in a sense where items have own properties and change over time. Therefore, in Normalized Systems, there is no such concept of enumeration.

A. Traditional OOP Inheritance

The first of the patterns uses a default implementation of inheritance in the underlying programming language. In case of Python, multiple inheritance with MRO allows creating subclasses for combinations given by the conceptual model. We immediately run into the combinatorial effect. First, we need to implement classes according to the model with inheritance and call the initializer of superclass(es) in the initializer (i.e.,

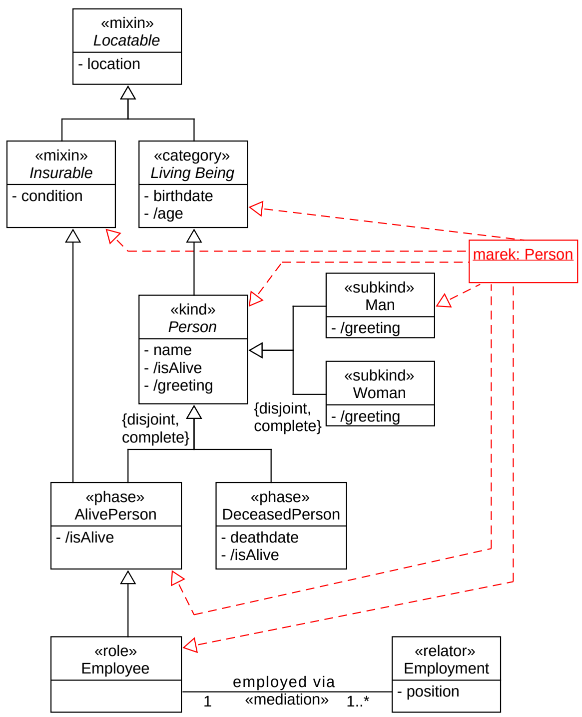


Fig. 1. Diagram of OntoUML example model with instance

```python
class LivingBeing(Locatable):

    def __init__(self, birthdate, location):
        super().__init__(location)
        self.birthdate = birthdate

    @property
    def age(self):
        # computation of age
        return result

class Man(Person):

    @property
    def greeting(self):
        return f'Mr. {self.name}'

class AlivePerson(Person, Insurable):

    def __init__(self, name, birthdate, location,
    ↪   condition):
        Person.__init__(self, name, birthdate,
        ↪   location)
        Insurable.__init__(self, condition)

    @property
    def is_alive(self):
        return True

class EmployeeMan(Employee, Man):
    pass    # Employee __init__ inherited

marek = EmployeeMan("Marek", ...)
```

Fig. 2. Part of the traditional inheritance implementation

`__init__` method). In case of single inheritance, it can be easily resolved using the built-in `super` function, but in case of multiple inheritance, all superclasses must be named again as call of the function `super` returns only the first matching according to MRO as shown in Figure 2. Also notice that all arguments of the initializer must be propagated and repeated. A possible optimization would be to use variadic `*args` and `**kwargs`, but in exchange for readability and checks with respect to number (and type) of arguments passed. Another interesting fact in our example is that `EmployeeMan` does not need to define the initializer, as it inherits the one from `Employee` and `Man` inherits it from `Person`. If `Man` has its own attributes, then `EmployeeMan` would have the initializer similarly to `AlivePerson`.

After having the model classes implemented, extra classes must be generated as an object can be instance of only one class. For example, `marek` is instance of such class `EmployeeMan`. For our simple case, number of extra classes is six – `Man` and `Woman` combined with `AlivePerson`, `DeceasedPerson`, and `Employee`. Adding a single new subclass of `Person`, e.g., `DisabledPerson`, would result in doubling the number and therefore a combinatorial explosion. The second point where a combinatorial effect resides is the order of superclasses (*bases* or *base classes* in Python), which influences MRO. For instance, if `Person` and `Insurable` define the same method – in our case the one from `Person` – it would be resolved for execution according to order in list `EmployeeMan.mro()`. On the attribute level, each change propagates to all subclasses, i.e., it is again a combinatorial effect. This can be avoided using the mentioned `**kwargs` and their enforcing, as shown in Figure 3. Knowledge of superclasses for initialization can be then used to automatically call the initializer of all the superclasses. We implement this in helper function `init_bases`, where superclasses are iterated a initialized in the reverse order to follow the MRO, i.e., the initializer of first listed superclass is used as the last one to eventually override effects of others.

With implementation shown in Figure 3, all classes with initializers can be easily generated automatically from the model with a single exception. The order of classes – i.e., if `AlivePerson` should be a subclass of `Person` and then `Insurable` or vice versa – is not captured in the model, but it is crucial for MRO. The order of superclasses has to be encoded in the model, or alternatively all permutations must be generated, which would result in a significantly higher number of classes that are not necessarily needed. Navigation is done naturally thanks to MRO and Python itself, for example, `marek.greeting` or `marek.location`.

B. The Union Pattern

The Union pattern basically merges an inheritance hierarchy into a single class. In our case, the "core" class of hierarchy can be naturally selected as `Person`. All subclasses are uniquely merged into `Person` and `Person` merges also all superclasses as shown in Figure 4. For example, if there is another subclass of `Insurable`, it would not be merged into `Person`. On the other hand, for example, a new subclass of

```python
def init_bases(obj, cls, **kwargs):
    for base in reversed(cls.__bases__):
        if base is not object:
            base.__init__(obj, **kwargs)

class Person(LivingBeing):

    def __init__(self, *, name, **kwargs):
        init_bases(self, Person, **kwargs)
        self.name = name

class AlivePerson(Person, Insurable):

    def __init__(self, **kwargs):
        init_bases(self, AlivePerson, **kwargs)

# ...

class EmployeeMan(Employee, Man):

    def __init__(self, **kwargs):
        init_bases(self, EmployeeMan, **kwargs)

marek = EmployeeMan(name="Marek", ...)
```

Fig. 3. Implementation of initializers and extra classes with use of keyword arguments and helper function for model-driven development

Man would be merged. This pattern is inspired closely by the "single-table inheritance" used in relational databases, but it immediately runs into problems once behaviour should be implemented.

According to the pattern, each decision on generalisation set of subclasses must be captured in the class that unions the hierarchy. In our case, we need three discriminators – for Man and Woman, for AlivePerson and DeceasedPerson, and for Employee. Value of each discriminator described what subclass(es) are "virtually" instantiated. All of these generalisation sets are disjoint and complete with the exception of the one with Employee that is not complete (i.e., not all alive persons must be employees). If there is a non-disjoint generalisation set, it would be solved using enumeration of all possibilities for the discriminator. For example, if Man/Woman is not disjoint nor complete, there would be four possible options (no, just man, just woman, both) instead of current two (just man or woman).

To allow polymorphism without branching and checking the discriminator value and taking a decision on behaviour with combinatorial effect, we use directly classes for delegation as values for discriminators, similarly to the well-known "State pattern". With this implementation, it incorporates separation of concerns and improves re-usability. It is crucial that all attributes, i.e., data, are encapsulated in the single object that is passed during the calls. Figure 4 shows Delegation descriptor for secure delegation of behaviour to separate classes that even do not need to be instantiated; therefore, static methods are used, and an instance of the union class is passed.

It is essential to point out that this solution may reduce the number of classes, but only of purely data classes without behaviour. Union classes can be then easily generated from a conceptual model. The detection of the "core" class is a matter of the model – if OntoUML is used, naturally all identity providers (e.g., with stereotype Kind) are suitable. In modelling languages that have no such explicit indication, a special flag has to be encoded in the model. Classes encapsulating behaviour can also be easily generated from the model and related to data class using the explained Delegation descriptor. There is one problem with this pattern implementation – it does not support isinstance checks. When avoiding inheritance, the only possible solution lies in special metaclass that would override __instancecheck__. This would also require to forbid instantiation of behaviour classes, so it is unambiguous if the object is an instance of a data or a behaviour class.

```python
class Man:

    @staticmethod
    def greeting(person):
        return f'Mr. {person.name}'

class Delegation:

    def __init__(self, discriminator, attr):
        self.discriminator = discriminator
        self.attr = attr

    def __get__(self, instance, owner):
        d = getattr(instance, self.discriminator)
        a = getattr(d, self.attr) if d else None
        return a(instance) if callable(a) else a

class Person:

    greeting = Delegation('_d_man_woman',
    ↪ 'greeting')
    is_alive = Delegation('_d_alive_deceased',
    ↪ 'is_alive')
    age = Delegation('_x_living_being', 'age')

    def __init__(self, name, birthdate, location,
    ↪ condition):
        self.location = location
        self.condition = condition
        self.birthdate = birthdate
        self.name = name
        # optional-subclass attributes
        self.employment = None
        self.deathdate = None
        # discriminators
        self._d_man_woman = None
        self._d_alive_deceased = None
        self._d_employee = None
        # superclasses with behaviour
        self._x_living_being = LivingBeing

    def d_set_man(self):
        self._d_man_woman = Man

    def d_set_employee(self, employment):
        self._d_employee = Employee
        self.employment = employment

# ...
```

Fig. 4. Part of union pattern implementation

C. Composition Pattern

The composition pattern follows the well-known precept from OOP – "composition over inheritance". Similarly to union pattern, a "core" class per hierarchy in the model must be identified. Classes are then connected using association *is-a* instead of inheritance. The Union pattern basically merges an inheritance hierarchy into a single class. For the original subclass, it is required to have a link to its superclass(es), but the other direction is optional unless the generalization set is complete or the superclass is abstract.

The final implementation of this pattern is based on the improved traditional OOP inheritance. Instead of inheritance, i.e., specification of superclasses, all superclasses from the conceptual model are instantiated during the object initialization. During this step, a bidirectional link must be made to allow navigation from both superclass and subclass instances. The "core" class must be again chosen to allow creation of composed object using multiple subclasses, e.g., an instance of `Person` that is also an `Employee` and a `Man`.

```
class Delegation:

    def __init__(self, p_name, a_name):
        self.p_name = p_name
        self.a_name = a_name

    def __get__(self, instance, owner):
        p = getattr(instance, f'{self.p_name}')
        a = getattr(p, self.a_name) if p else None
        return a(instance) if callable(a) else a

    def __set__(self, instance, value):
        p = getattr(instance, f'{self.p_name}')
        setattr(p, self.a_name, value)

class LivingBeing:

    location = Delegation('_p_locatable',
    ↪  'location')

    def __init__(self, *, birthdate, _c_person=None,
    ↪  _p_locatable=None, **kwargs):
        self._p_locatable = _p_locatable or
        ↪  Locatable(_c_living_being=self,
        ↪  **kwargs)
        self._c_person = _c_person
        self.birthdate = birthdate

# ...
```

Fig. 5. Part of composition pattern implementation

The example in Figure 5 shows that we also incorporated a *Delegation* descriptor. Although it results in repetition when defining where to delegate, it clearly describes the origin of a method or an attribute, and it can be generated easily. With the fact that these parts can be generated, combinatorial effects related to renaming or other changes of methods and attributes used for delegation are mitigated. The diamond problem is solved directly by passing child and parent class objects as optional arguments during initialisations. It could also be solved using metaclasses, but as this code can be generated, it allows higher flexibility and eventual overriding.

As the built-in MRO is not used, the resolution must be made manually on the model level similarly to the Union pattern, i.e., to decide what overrides and what is overridden. By replacing inheritance with bidirectional links, we managed to significantly limit combinatorial effects, but in exchange for the price in requiring additional logic and moving the MRO into the model itself. Unfortunately, this implementation needs also to incorporate model-consistency checks, as we do not enforce multiplicity in child-parent links according to the pattern design.

D. Generalisation Set Pattern

This pattern enhances the Composition pattern by adding particular constructs that encapsulate logic regarding generalisation sets. Inheritance relation is not transformed into *is-a* association but into connection via a special entity that handles related rules, such as complete or disjoint constraints and cardinality. As we present in Figure 6 this helps to remove shortcomings of the Composition pattern and its difficult links and composed-object instantiation. Instead of multiple child links, there is just one per Generalisation Set (GS), and parent links are changed accordingly. An object of `GS` class maintains the inheritance and ensures the bi-directionality of links.

```
class Delegation:

    # ...

    def __get__(self, instance, owner):
        gs = getattr(instance, f'{self.gs_name}')
        p = getattr(gs, f'{self.p_name}')
        a = getattr(p, self.a_name) if p else None
        return a(instance) if callable(a) else a

# ...


class GS_ManWoman:

    _gs_name = '_gs_man_woman'

    disjoint = True
    complete = True

    def __init__(self, person, man=None,
    ↪  woman=None):
        self.person = person
        self.man = man
        self.woman = woman
        self.update_links()

    def update_links(self):
        setattr(self.person, self._gs_name, self)
        if self.man is not None:
            setattr(self.man, self._gs_name, self)
        if self.woman is not None:
            setattr(self.woman, self._gs_name, self)

# ...
```

Fig. 6. Generalisation Set implementation example

Introduction of an intermediate object to encapsulate inheritance and related constraints adds complexity in two aspects. First, the diamond problems must be still treated by sharing superclass objects in the hierarchy for eventual reuse. Second, the delegation must operate with the intermediary object when accessing the target child (or parent) object. However, solutions to these issues can be also generated directly from the model and in principle – despite their complexity – they do not hinder evolvability.

Finally, this solution (if entirely generated from a model) is the most suitable, since it limits combinatorial effects and allows to efficiently check consistency with the model in terms of inheritance and generalisation set constraints. Although in some cases, the `GS` object is not adding any value (e.g., a single child and a single parent case), implementing a combination of a generalisation set and composition patterns would make the software code harder to understand. Unity in implementation of conceptual-level inheritance is crucial here.

V. Inheritance Expanders Design

In this section, we describe the common design of the inheritance patterns code generation using Normalized Systems Expanders and related tooling. We do so for all of the patterns and again using Python programming language.

A. Capturing Inheritance in Models

The input models for expanders describe all the entities with their properties and relations. Although creating a custom meta-model for using expanders is possible, that is not part of our goals. Therefore, we use the NS meta-model of so-called Elements. Our focus is on Data Elements, which can be seen as counterparts to classes or entities from structural conceptual modelling. For data elements, we are interested in their fields – link fields representing relationships between data elements and value fields that are attributes with a data type.

Due to the evolvability issues, there is no way how to model inheritance directly with the current NS meta-model. Nevertheless, it is required to capture this somehow in a model, e.g., that a data element Person is a specialisation of a data element Living Being. For the same reason, there is no way to model an abstract class (or entity) as it makes no sense without inheritance. Another complication comes with generalisation sets that are groups of inheritance relations.

NS meta-model provides a way to encode additional information using so-called options flexibly. Various options are based on the construct to which they are attached, e.g., field options are related to a certain field. We use these options to encode inheritance-related information in NS models:

- **Abstract Element** – Indication that a data element is abstract, i.e., must be further specialised. It is captured using a single data option on a data element – if present, it is an abstract data element.
- **Inheritance** – Indication that a link between two data elements forms an inheritance relation. There might be a field option stating that the target is a generalisation or specialisation on a link field.
- **Generalisation Set** – Instead of creating groups of relations and then stating if those are disjoint, complete, or both, we capture these atomically. On a link field with an inheritance indication, a field option might add if the relation is disjoint with other such fields (a field is uniquely identified by the name and the data element name).

B. Code Templates and Mapping

With the inheritance-related information encoded in NS models, the expanders can be designed. First, it is required to prepare a mapping that queries the information from a model and supplies it to corresponding code templates. Then, a code template creates a file based on these supplied inputs. The high-level design is shown in Figure 7.

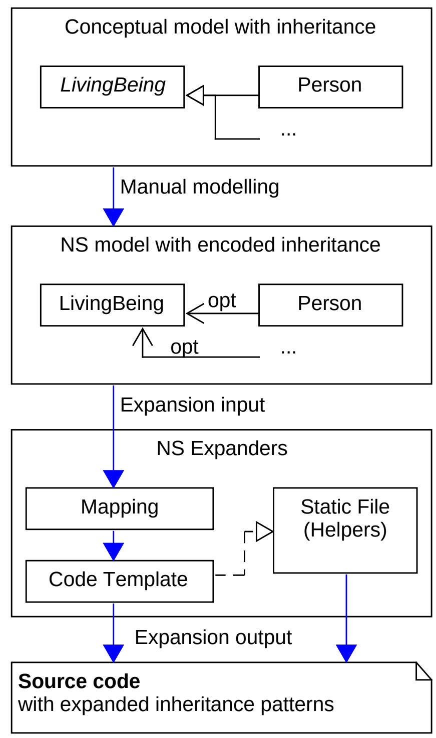


Fig. 7. High-level design of inheritance patterns expansion

For all data elements in a component, the template needs all data options, fields, and fields options. Other model parts are out of our scope. Data elements are turned into classes, with fields turned into instance attributes based on a specific inheritance pattern used. The patterns designed in the previous section are a basis for this step. All model-specific names are variable and queried from an input model.

There might be some additional static constructs, e.g., `Delegation` class or `init_bases` function, that requires no inputs from a model. Those common definitions are separated into static files, which are just copied. Code templates may rely on such definitions – import them from a known location and use them. That is depicted by the dashed arrow in Figure 7.

As our purpose is to generate class hierarchies in Python based on the design presented in the previous section, there will be one expander per pattern. An expander, in our case, produces a single Python file with all classes, including hierarchies for a given input model. It simplifies both generation and manipulation with the result as we are comparing different inheritance patterns. Moreover, it is natural for Python to have all data classes of a single component inside one file in contrast to Java with file-per-class.

C. Anchors in Expanders

Anchors are an essential part of expanders. We follow the best practices in using anchors. It delimits generated fragments based on a particular entity, e.g., when a class is formed from a data element, it is delimited by anchor comments. Similarly, it is done for a block of fields assignments or method parameters that are generated from a model.

Another type of anchor is related to custom code. Wherever it may be necessary to add custom code, there should be such an anchor delimiter. Then, expanded code can be easily enhanced with insertions that can be later harvested and eventually re-injected upon rejuvenation. These anchors are added on module, class, and method levels. To maintain code readability after generation, there are multiple anchors for each level, e.g., at the top of a module for imports, before class declarations, and finally, at the end after all class declarations.

D. Features in Expanders

Similarly to anchors, expanders may also contain so-called features. It can be used as optional parts of expanders or for potential future extensions of expanders. We follow the same principles and add anchors so that future extensions may add code fragments to any part of the output source code. That will simplify the future prototyping of enhancements and adding optionally generated code. However, we did not identify any practical use of features for this work as we aim to generate specific inheritance patterns described in the previous section.

VI. Expanding Inheritance Patterns

The design described above outlines the common design for code generation of all inheritance patterns as part of G2. In this section, we provide additional details in the design specific to a certain pattern. It also addresses G3 as we implement and demonstrate the expanders. The implementation also serves evaluation purposes according to the DSR methodology.

An expander takes a model of NS elements together with technical details (configuration) as inputs. A mapping (part of expander) defines what is extracted from the model and eventually transformed (e.g., new derived variables). Finally, a code template (part of expander) is filled based on the mapping with data originating in the input model and files are produced and stored according to the expander and project configuration. The input model can be seen in an XML representation, but that is just a projection for serialization; from our perspective, the syntax does not matter; we care about semantics.

A. Traditional Inheritance Expanders

The first presented pattern uses the traditional inheritance. In the first stage, the inheritance is done using the multiple inheritance with MRO that is standard in Python. That will serve as the reference for other patterns which should "work the same" but without using the inheritance directly for evolvability reasons. Moreover, at this stage, the expander is the most straightforward to design:

1) All data elements become classes.
2) For each data element, all superclasses must be resolved by checking link fields for the specialisation option.
3) For each data element, all value fields and non-inheritance link fields are instance attributes (using initialisation `__init__`).
4) For each data element, all calculated fields become a method with property decorator and anchors for implementing the calculation.

The second stage is to generate classes representing all possible combinations of classes (e.g., `EmployeeMan` from Figure 2). It clearly shows the potential of a combinatorial explosion. As an optimisation, we exclude the unnecessary combinations with disjoint classes for generalisation sets:

1) For each class in a single hierarchy, list all combinations for its specialisations.
2) Remove combinators where any two classes are disjoint.
3) Create classes from the list, with the corresponding superclasses.
4) List all properties for the class and remove duplicates.
5) Create initialisation that calls correct initialisation of the parents.

When using the proposed '`init_bases`' function, the two list steps are significantly reduced. Only own fields are initialised directly and other are passed to this function. It makes both expanders and resulting code simpler and shorter.

B. Union Pattern Expanders

Expansion with the union pattern has the opposite procedure when compared to the previous one. It requires resolving the "folding" of class hierarchy into a single one. It must solve several edge cases such as hierarchies with multiple ultimate parents or the creation of discriminator attributes. Then all fields are merged into a single union class based on the previous expander design.

1) Find a root in a hierarchy; if there is not a single root, add an artificial one.
2) Create data element for each hierarchy.
3) Create a discriminator instance attribute for every disjoint generalisation set in the hierarchy.
4) Add all value, link, and calculated fields as for traditional inheritance into the single class.

The logic about solving the potential issue with the hierarchy and folding it to a single class must be robust, and it is resulting in quite complex code in expanders and mapping. Moreover, the resulting classes are also very complex and hard to use, as we discuss the results further in the next section. We did not use the `Delegation` feature here as it adds complexity for methods, but at this point, there are no generated methods – only attributes.

C. Composition Pattern Expanders

The composition pattern that follows the "composition over inheritance" rule with the use of delegation can be expanded more easily than the union pattern. It creates a class per data element as for traditional inheritance, but instead of using inheritance, it uses the `Delegation` descriptor. For parents and children, there are additional special attributes just as shown in Figure 5. The steps are as follows:

1) All data elements become classes.
2) For each data element, all superclasses are represented as `_p_`-prefixed attributes.
3) For each data element, all subclasses are represented as `_c_`-prefixed attributes.
4) For each data element, all value fields and non-inheritance link fields are instance attributes (using initialisation `__init__`).
5) For each data element, all calculated fields become a method with property decorator and anchors for implementing the calculation.
6) For each data element, all inherited fields are added using `Delegation` description on the class level.

Steps 1, 4, and 5 are the same as for the traditional inheritance expanders. It shows that only the inheritance part is replaced by delegation and necessary preparations (of the linking attributes). Also, note that this pattern does not take into account disjoint nor complete constraints of generalisation sets.

D. Generalisation Set Pattern Expanders

As explained, the generalisation set pattern can be seen as an extension to the composition pattern. However, in terms of expansion, there is a significant overhead needed to find the generalisation sets in the input model. It must group the inheritance relations where disjoint, and union constraints through options are used. Based on these groups, additional `GS` classes can be generated as shown in Figure 6.

It creates classes from data elements as for composition pattern but uses `GS` class for `_p_`-prefixed and `_c_`-prefixed where it exists. The added complexity in both expanders and resulting code turned to be very significant. One of the main issues was the different handling of single-class generalisation sets and multiple-class generation sets. Therefore, we create `GS` classes also for single-class sets to keep consistency across source code.

E. Expanders Implementation

With the design described in the previous section and steps for each pattern expander, the implementation did not require any additional knowledge. The steps are represented in "code" of mappings that queries and transforms the information from an input model to desired form, e.g., a list of data elements in a component or a list of superclasses for a data element. Although the expressiveness of XML mappings in expanders is limited, it was entirely sufficient for our implementation. Figure 8 shows a basic mapping common for all of our expanders.

```xml
<mapping>
  <value name="componentName"
  ↪   eval="component.name"/>

  <list name="dataElements"
  ↪   eval="component.dataElements"
  ↪   param="dataElement">
    <value name="name" eval="dataElement.name"/>

    <list name="valueFields"
    ↪   eval="dataElement.fields" param="field"
    ↪   filter="field.valueField neq null">
      <value name="name" eval="field.name"/>
      <value name="type" eval="field.valueField.
      ↪   valueFieldType.name"/>
    </list>

    <list name="linkFields"
    ↪   eval="dataElement.fields" param="field"
    ↪   filter="field.linkField neq null">
      <value name="name" eval="field.name"/>
      <value name="targetElement"
      ↪   eval="field.linkField.
      ↪   targetElement.name"/>
      <value name="lnType" eval="field.linkField.
      ↪   linkFieldType.name"/>
    </list>
   </list>

   <!-- ... -->
</mapping>
```

Fig. 8. Fragment of a basic mapping

Whereas the mapping file queries and transforms the information to be supplied into a code template, the code template itself may contain a more complex transformation. It uses the String Templates well-known from Java that are capable of using conditions, loops, functions, and many other constructs. For example, filtering the non-inheritance link fields has been done in templates when generating the corresponding code (instance attributes). In summary, the implementation based on the design described above turned to be straightforward without any unexpected issues. A fundamental code template fragment from traditional inheritance expanders is shown in Figure 9.

F. Expanders Comparison

Both the design of the expanders and their implementation provided valuable knowledge about the patterns and verification of the theoretical comparison made in the previous work. In terms of complexity (computed based on logic and mathematical operations needed in mappings and templates), the composition pattern shows the best results even when compared to the traditional inheritance. The leading cause has been identified in omitting the disjoint and complete constraints.

The resulting code varies a lot based on the input model and hierarchies. Generally, traditional inheritance suffers from

```
delimiters "$", "$"

base() ::= <<
# Generated traditional inheritance patterns for
# component "$componentName$"
import Delegation from .helpers
@anchor:imports
# anchor:custom-imports:start
# anchor:custom-imports:end

# anchor:dataElements:start
$dataElements:dataElementEntry()$
# anchor:dataElements:end

# anchor:custom-moduleEnd:start
# anchor:custom-moduleEnd:end
>>

dataElementEntry(de) ::= <<

class $de.name$(
   # anchor:de-$de.name$-bases:start
   $deSuperclasses(de)$
   # anchor:de-$de.name$-bases:end
):

    $deInitializer(dataElement)$

    @anchor:methods
    # anchor:custom-$de.name$-methods:start
    # anchor:custom-$de.name$-methods:end

    ...


>>
```

Fig. 9. Fragment of a code template

a combinatorial explosion. Union pattern encapsulates the explosion inside a single class which is unmaintainable when it "folds" a more complex hierarchy with several generalisation sets. It may be helpful for smaller and mostly linear hierarchies. The code with generalisation sets pattern is the most complex in terms of additional classes and navigation between subclasses and superclasses.

VII. Evaluation and Discussion

In this section, we summarize and evaluate achievements of our research. Based on our observations and implementation of inheritance using patterns, we evaluate inheritance in Python. The patterns and its key aspects are compared in Table I. Then, we also describe the future steps that we plan to do as follow-up research and projects based on outputs described in this paper.

A. Initial Prototype Implementation

We demonstrated our implementation of all four previously designed patterns as set in the goal G1. Mostly, results and related usability options are consistent with the design. The more we minimize or constrain combinatorial effect, the more complex and hard-to-use (in terms of working with final objects) the implementation gets. We were able to simplify use of objects for the price of repetition and use of special constructs for delegation. Contrary to the original patterns design, we were not able to efficiently combine multiple patterns together based on various types of inheritance used in the model. As a result, our implementation of the most complex generalisation set pattern is suggested as a prototype of how inheritance may be implemented if one wants to avoid combinatorial effects while still needing to capture inheritance in a generic way for models of any size and complexity.

B. Expanders Prototype

To also fulfil the goals G2 and G3, we designed code generation for all of the inheritance patterns (in Python programming language) and proved its feasibility by implementing them with the use of Normalized Systems Expanders. It revealed several implementation issues and helped to refine the design based on the DSR design cycle. It verified the hypothesis that union and generalisation set patterns create over-complex source code that is hard to maintain and extend. The most suitable composition pattern keeps flexibility, and additional constraints can be supplied using custom insertions and extra logic where needed.

The expanders clearly mitigate the burden of additional complexity brought by inheritance implementation patterns as that is something generated and managed only in the code templates. A change in the inheritance can be easily projected into code by re-generating. The burden that remains is related to custom code, where a programmer must use the navigation between superclasses and subclasses that may not be natural, but the patterns were designed to make this as effortless as possible.

C. Evolvability of Python Inheritance

During the implementation of the patterns, it became obvious that even high flexibility of programming language and allowed multiple inheritance do not help in terms of coupling and combinatorial effects caused by using class-based inheritance. With a simple real-world conceptual model, we were able to show how the combinatorial explosion endangers the evolvability of software implementation. MRO algorithm used in Python does not help with limiting combinatorial effects. Rather it is the opposite since order in which superclasses are enumerated significantly influences implementation behaviour. Also, it makes harder to combine overriding from two superclasses, for example, both class `A` and `B` implement methods `foo` and `bar` but subclass `C` cannot inherit one from `A` and other from `B` (solution is to override both and call it from subclass manually).

On the other hand, the flexibility of Python proved to be useful while we were implementing the patterns. Thanks to magic methods, descriptors, and metaclasses, the final implementations allow creating easy-to-use and inheritance-free objects even though underlying complex relations with constraints are needed as shown in the examples. Notwithstanding, such possibilities of Python are similar to constructs and methods in other languages (e.g., reflection). While trying to implement the patterns efficiently, we concluded that generating implementation from a model is crucial for evolvability

TABLE I. COMPARISON OF THE INHERITANCE IMPLEMENTATION PROTOTYPES

Implementation	Classes*	Extra constructs	CE-handling	Issues
Traditional	$N + 2^N$	0	none	initialization, order of superclasses, uncontrolled change propagation
Traditional + init_bases	$N + 2^N$	init_bases function	shared initialization	shared attributes across hierarchy, order of superclasses, uncontrolled change propagation
Union pattern	2	Delegation class	shared class (merged)	Separation of Concerns violated, maintainability, discriminators
Composition pattern	N	Delegation class	shared initialization, delegation	manual handling of GS constraints, added complexity (for humans)
GS pattern	$N + 1$	Delegation class, GS helpers	shared initialization, delegation	added complexity (for humans)

(*) per single hierarchy of N classes, worst case (all combinations needed)

regardless of what technologies are used, as a lot of repetition is needed.

D. Future Work – Inheritance Modelling for NS

As explained and practically demonstrated, we had to encode the inheritance relations in an NS model using fields options. A software analyst must create a link between data elements and then mark it as an inheritance in terms of modelling. In visualisations of such a model, it still looks like a regular relation. A potential next step would be to propose an extension to the NS Elements meta-model to directly model inheritance relations. That would require enhancement in the meta-model as well as incorporation in existing expanders.

E. Future Work – Production-Ready Technologies

The goal of the paper was to use Python to show reference patterns implementation prototypes. The next step may be to leverage the lessons learned to formulate a single transformation description using production-ready technology stack, e.g., Java EE. This final transformation of conceptual-level inheritance should allow simple extensibility and customizations. Moreover, it should cover all possibilities with respect to the underlying modelling language. This language does not have to be OntoUML used in this paper; however, it must be expressive enough to capture all the necessary details for a correct implementation – for instance, hierarchy “core” classes.

F. Future Work – Inheritance in UI/UX

Another step or aspect of implementing conceptual-level inheritance lies in creating a related user interface compatible with data and behaviour encapsulation. After there is MDD-based generation of code from a model for backend service, it should also be considered how to generate UI for related frontend. To elevate user experience, special relations made for implementing inheritance should look different than others. Moreover, the user should be able to easily create an object with a selection of possible subclasses closer to the model rather than to the implementation on the backend.

VIII. Conclusions

This extended paper proposes code generation design for conceptual-level inheritance patterns that minimise the issues caused by combinatorial effects. We analysed and demonstrated the evolvability of inheritance in Python using already-designed implementation patterns. Due to Python’s flexibility and ability to redefine core constructs, we managed to implement the conceptual-level inheritance implementation patterns while maintaining code readability and maintainability. Further, we designed code generation of those constructs from models using Normalized Systems Expanders and existing tooling. It proved the feasibility of implementing the patterns in larger codebases. Although we used Python for this work, it can be used as a basis for implementation in other object-oriented languages. Finally, future work has been outlined. The following challenges are identified in the generation of user interface fragments, e.g., forms or details, for entities with inheritance.

Acknowledgment

This research was supported by the grant of Czech Technical University in Prague No. SGS20/209/OHK3/3T/18.

References

[1] Marek Suchánek and Robert Pergl, “Evolvability Analysis of Multiple Inheritance and Method Resolution Order in Python,” in *PATTERNS 2020, The Twelfth International Conference on Pervasive Patterns and Applications*, H. Mannaert, I. Pu, and J. Daykin, Eds. IARIA, 2020, pp. 19–24, Accessed: 24 Nov 2021. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=patterns_2020_2_10_79

[2] Herwig Mannaert, Jan Verelst, and Peter De Bruyn, *Normalized Systems Theory: From Foundations for Evolvable Software Toward a General Theory for Evolvable Design*. Kermt (Belgium): Koppa, 2016.

[3] Marek Suchánek and Robert Pergl, “Evolvability Evaluation of Conceptual-Level Inheritance Implementation Patterns,” in *PATTERNS 2019, The Eleventh International Conference on Pervasive Patterns and Applications*, vol. 2019. Venice, Italy: IARIA, May 2019, pp. 1–6, Accessed: 24 Nov 2021. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=patterns_2019_1_10_78001

[4] Antero Taivalsaari, “On the Notion of Inheritance,” *ACM Computing Surveys*, vol. 28, no. 3, pp. 438–479, September 1996.

[5] Pierre Carbonnelle, “PYPL: PopularitY of Programming Language,” Feb 2020, accessed: 15 Nov 2021. [Online]. Available: http://pypl.github.io/PYPL.html

[6] Python Software Foundation, “Python 3.8.0 Documentation,” 2019, [online]. Accessed: 12 Nov 2021. [Online]. Available: https://docs.python.org/3.8/#

[7] David Hilley, "Python: Executable Pseudocode," [retrieved: Aug, 2020]. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.211.7674&rep=rep1&type=pdf

[8] A. R. Hevner, "Design science research," in *Computing Handbook, Third Edition: Information Systems and Information Technology*, H. Topi and A. Tucker, Eds. CRC Press, 2014, pp. 22: 1–23.

[9] ——, "The three cycle view of design science," *Scand. J. Inf. Syst.*, vol. 19, no. 2, p. 4, 2007. [Online]. Available: http://aisel.aisnet.org/sjis/vol19/iss2/4

[10] Andrei Alexandrescu, *Modern C++ Design: Generic Programming and Design Patterns Applied*, ser. C++ in-depth series. Addison-Wesley, 2001.

[11] NSX. Prime Radiant Online. Accessed: 2 Nov 2021. [Online]. http://primeradiant.stars-end.net/foundation/.

[12] H. Mannaert, P. De Bruyn, and J. Verelst, "On the interconnection of cross-cutting concerns within hierarchical modular architectures," *IEEE Transactions on Engineering Management*, 2020.

[13] Antero Taivalsaari, "On the Notion of Inheritance," *ACM Computing Surveys*, vol. 28, no. 3, pp. 438–479, September 1996.

[14] Giancarlo Guizzardi, *Ontological Foundations for Structural Conceptual Models*. Centre for Telematics and Information Technology, 2005.

[15] Alan Borning, "Classes versus prototypes in object-oriented languages." in *FJCC*, 1986, pp. 36–40.

[16] John Hunt, "Class Inheritance," in *A Beginners Guide to Python 3 Programming*. Springer, 2019, pp. 211–232.

[17] Fawzi Albalooshi and Amjad Mahmood, "A Comparative Study on the Effect of Multiple Inheritance Mechanism in Java, C++, and Python on Complexity and Reusability of Code," *International Joutnal of Advanced Computer Science and Applications*, vol. 8, no. 6, pp. 109–116, 2017.

[18] Matteo Orru et al., "How Do Python Programs Use Inheritance? A Replication Study," in *2015 Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 2015, pp. 309–315.

[19] Ravi Chugh, Patrick M Rondon, and Ranjit Jhala, "Nested refinements: a logic for duck typing," *ACM SIGPLAN Notices*, vol. 47, no. 1, pp. 231–244, 2012.

[20] John Hunt, *Advanced Guide to Python 3 Programming*, ser. Undergraduate Topics in Computer Science. Springer International Publishing, 2019.

Analysing a Systematic Literature Review Combined with an Undergraduate Survey on Misconceptions about Software Engineering

Carolin Gold-Veerkamp
University of Applied Sciences Aschaffenburg
Aschaffenburg, Germany
Email: carolin.gold-veerkamp@th-ab.de

***Abstract*—From a constructivist perspective, learning is an active, cognitive process in which individuals construct their own knowledge by connecting new concepts with previous knowledge, skills, and experience that serve as points of departure. The purpose of this study is to identify and analyse misconceptions in Software Engineering to use these insights for higher education. Therefore, a systematic literature review as a secondary data accumulation as well as a primary data acquisition covering undergraduates are conducted, analysed, and compared. Concerning the SLR, out of 2,158 publications found, only 20 evidence-based misconceptions/beliefs/myths from 3 papers could be found. The undergraduate survey resulted in 69 misconceptions covering 13 interviews. Additionally, both approaches have been compared.**

Keywords*—*Software Engineering; Higher Education; Misconceptions; Systematic Literature Review; Undergraduate Survey.

I. Introduction

From a constructivist point of view, learning is to be understood as an active, individual, situated, social, and cognitive psychological process. Each individual has to build up their own knowledge by combining new concepts based on previous knowledge, existing competencies, previous experience, as well as conceptions and putting it into a network-like relationship. This means, learners form conceptions and models to explain phenomena, processes, and artifacts before they are confronted with them in institutional learning. These possibly alternative – from scientific or expert perspective – conceptions have a twofold significant impact on the learning process. On one hand, they can serve as the basis for learning, on the other, they can also contradict the educational content and thus hinder the learning process.

To be able to do so in Software Engineering (SE) education, we have to take a step back and clarify which misconceptions undergraduates bring to university. Thus, this contribution is based upon two pillars, a Systematic Literature Review (SLR) as a secondary data analysis [1] as well as a primary data acquisition [2] that both provide information about SE misconceptions. These shall be presented, compared and contrasted.

As a consequence, the goal is to achieve sustainable learning (in higher education), a purely technical structuring of the learning content is therefore insufficient. Furthermore, didactics should do justice to the learners' "points of departure" [3, p. 6]. The Model of Educational Reconstruction, which is epistemologically based on the constructivist position, calls for precisely this consideration [3] [4]. The model comprises the triad of *content clarification*, *learners' conceptions*, and *didactic design*; it considers the scientific concepts and the student conceptions as equivalents.

This contribution is structured on the basis of these two pillars: The first part covers the SLR (Sections II-IV) starting with terminological aspects, as a plurality of terms evolved, and related work in other disciplines. The SLR process is explained in Section II, complemented by the results (Section III) as well as a short discussion (Section IV). The second part explaines the survey process (Sections V-IV) covering the approach in Section V, the results (Section VI) and again a short discussion in Section VII. The third part (Section VIII) combines both pillars by analysing and comparing the MCs found; the aim is not to conduct a one-to-one matching, but to look on context-related proximity.

A. Terminological Aspects

Due to many different ways of looking at the research object '*(mis-)conceptions*' as well as the critical examination of the terminology, an abundance of terms has developed. The different understandings have led to a plurality of terms with multiple connotations. The abundance of technical terms has risen so much in the course of research (especially in natural sciences didactics) that it is now almost impossible to survey. The fact that the terms cannot be clearly distinguished from each other often leads to a more or less synonymous use and thus to an undifferentiated mix. As a result of the dissatisfaction with this situation, researchers have again constructed and defined additional terms, which expands the existing term dilemma.

In addition to [5] [6], also others include collections of terms. This list (merely referenced by several publications) gives an impression of the broad spectrum:

- *Preconceptions* [5]–[8]
- *(Students') conceptions* [5] [6]
- *Alternative conceptions* [5]
- *Naïve conceptions* [6]
- *Naïve theories* [5] [6]
- *Naïve beliefs* [5]
- *Beliefs* [9]
- *Alternative beliefs* [5]
- *Alternative frameworks* [5] [6] [8]
- *Intuitive theories/science* [8]
- *Prior knowledge* [8]
- *Misconceptions*, the "standard term" as [5, p. 119] state – despite the negative connotation [6] [8].

In spite of the heterogeneity of terms, opposed opinions and discussions on the different types of expression, it can be stated consensually that individuals each develop different conceptions of certain concepts, which should and must be used as a starting point in teaching. These conceptions can, but do not have to be in line with modern scientific theories [6] and therefore may act as learning obstacles [10], often referred to as *misconceptions* (MC).

B. Related Work on Misconceptions in Didactics

The research and publications about misconceptions in natural sciences in the context of school are immense, as a bibliography by Duit [11] proves. When looking at the catalogue, encompassing over 8,300 publications and summarizing them per decade (Figure 1), it is obvious that since the mid-1970s international researchers have been investigating the field.

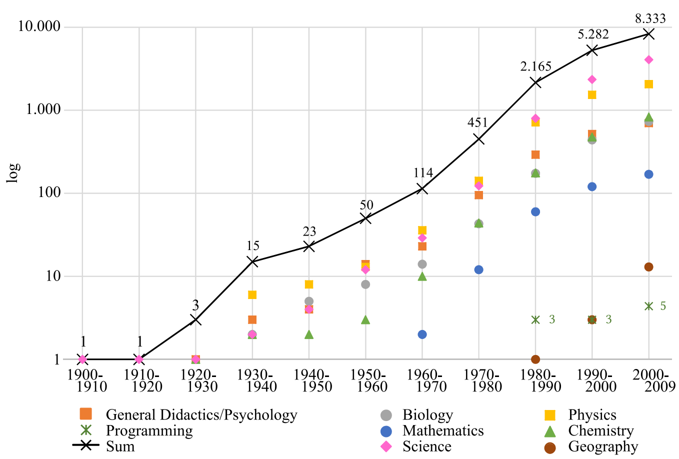


Fig. 1. Diagram of Accumulated Number of Publications per Decade, Sorted by Discipline Listed in [11]; esp. Focused on 'Programming'

Out of these, merely five publications [12]–[16] can be assigned to 'programming' as nearest to SE, but also to science and/or maths; i.e., they are equivocal.

Moreover, in the last few years, several papers on misconception research in computer sciences appeared concerning:

- ... programming [17]–[20] and object-oriented programming in particular [21] [22].
- ... artifacts, e.g., computers, smartphones, and so on [23, and others].
- ... the Internet [24].

All publications listed have in common that they do not particularly deal with SE and are contextualised within school education. This results in research needs for SE in university.

II. Method: SLR on Misconceptions in SE

In order to be able to present the state of research on (undergraduate) conceptions in SE, there is a need for a SLR, which summarizes all available information about this phenomenon thoroughly and impartially [25, p. 7]. Conducting a SLR is a quantitative methodology of secondary data collection for the synthesis of research results from primary studies. The guidelines – used here – that Kitchenham and Charters have drawn up for SE are derived from several approaches in medicine and the social sciences [25, p. vi].

The following explanations, which describe the three-phase process of the SLR – carried out as a computer-based, automated search – are divided into the initial planning in Section II-A, the actual practical implementation (Section II-B) and the subsequent presentation and use of the results (Section III), see also [25] [26].

A. Phase 1: Initial Planning

The planning of the SLR contains some parameters that require previous definition in order to minimize bias. The SLR is determined as follows:

1) Research Question(s): To what extent does research on misconceptions in SE already exist? Which misconceptions in SE are known/documented?

2) Search Strategy:

a) Language Selection: At this point, the language radius, which is one of the inclusion criteria, should be anticipated. The reason for this is the following definition of the Search Query (SQ). Since research on conceptions is international, publications in German and English are considered.

b) Queries and Synonyms: Regardless of the various connotations (Section I-A), the search should encompass the previous research on MCs in SE as broadly as possible. Therefore, the search query is based on the numerous synonymously used English terms shown in Table I. (Indicating wildcards, i.e., placeholders, by an asterisk (*).)

TABLE I
Development of the English Search Query Using Synonyms

<table>
<tr><th rowspan="2">Synonyms</th><th colspan="2">Substrings</th></tr>
<tr><th>Noun</th><th>Adjective</th></tr>
<tr><td>preconceptions</td><td>preconception*</td><td>–</td></tr>
<tr><td>students' conceptions</td><td rowspan="3">conception*</td><td>–</td></tr>
<tr><td>alternative conceptions</td><td>alternative</td></tr>
<tr><td>naïve conceptions</td><td rowspan="3">naïve</td></tr>
<tr><td>naïve theories</td><td>–</td></tr>
<tr><td>naïve beliefs</td><td rowspan="3">belief*</td></tr>
<tr><td>beliefs</td><td>–</td></tr>
<tr><td>alternative beliefs</td><td rowspan="2">alternative</td></tr>
<tr><td>alternative frameworks</td><td>–</td></tr>
<tr><td>intuitive theories</td><td>–</td><td rowspan="2">intuitive</td></tr>
<tr><td>intuitive science</td><td>–</td></tr>
<tr><td>prior knowledge</td><td>–</td><td>prior</td></tr>
<tr><td>misconceptions</td><td>misconception*</td><td>–</td></tr>
</table>

In contrast to *preconception*, *conception*, *belief* and *misconception*, the terms *theory*, *framework* and *science* (plus plurals) are only included in combination with the respective adjectives (Table I), since they are often used as technical terms in SE and unspecific for answering the research question. Same applies to the terms *student*, *knowledge* and *science*, because of the usage of pedagogical databases. These are combined with the disciplinary focus on SE, resulting in Search Query 1; including wildcards (*) and search for exact phrases (quotation marks). (The equivalent German SQ is not attached here.)

c) Database: Electronic literature databases are selected based on Kitchenham et al. [27] in combination with [28]. Kitchenham et al. have already dealt intensively with SLRs in the area of SE and set up a list of important English-language journals and conferences, which they themselves use for their literature research (see Table II).

(“software engineer*” OR “software development*” OR “software process*”)
AND
(“preconception*” OR “conception*” OR “belief*” OR “misconception*” OR
“naïve theor*” OR “alternative framework*” OR
“intuitive theor*” OR “intuitive science” OR “prior knowledge”)

Query 1. English Search Query

TABLE II
SELECTION OF ELECTRONIC DATABASES FOR SLR BASED ON [27] [28]

Source	IEEE	ACM	SD	SC	SL	ERIC	WoS	GS	arXiv	dblp
Information and Software Technology			X	X				X		
Journal of Systems and Software			X	X				X		
IEEE Transactions on SE	X							X		
IEEE Software	X							X		
Communications of the ACM		X								
ACM Computer Surveys		X								
ACM Transactions on SE Methodologies		X								
Software Practice and Experience								X		
Empirical SE Journal					X					
IEEE Proc. Software (now: IET Software)	X							X		
Proc. Int. Conference on SE	X	X						X		
Proc. Int. Symp. of Software Metrics	X	X						X		
Proc. Int. Symp. on Empirical SE	X	X						X		

These are used as a basis to identify databases that include these compilations, namely: IEEE-Xplore [29], ACM-Digital Library [30], SpringerLink (SL) [31], Scopus (SC) [32], and Science Direct (SD) [33]. This selection is supplemented by further search engines from the educational context (ERIC [34], Web of Science (WoS) [35]) and the metadata database GoogleScholar (GS) [36]. In addition to the proposed ones, arXiv [37], an open access repository for electronic preprints from numerous areas – including computer science –, and the dblp [38], which is co-founded by the German federal government, are used.

3) Selection Strategy: The selection is controlled on the basis of the following predefined Inclusion (IC) and Exclusion Criteria (EC).

IC.1 The publication is written in English or German language.
IC.2 It is explicitly about the discipline SE.
IC.3 MCs in SE are explicitly mentioned.

EC.1 The contribution is an abstract, workshop, poster, or similar, as these do not provide in-depth information.

4) Quality Assessment: The gathered publications have to be qualified against predefined Quality Criteria (QC):

QC.1 *Traceability*: How do the authors know this misconception? It is scientifically important to be able to track where the information comes from.
QC.2 *Validation*: Has it been confirmed that it is a misconception? How did the authors validate the conception to be “at odds with modern scientific theories” [6, p. 2]? If not done, there is no indication that it is really a *mis*conception.
QC.3 *Occurence in the population*: Does this misconception exist in the population? Did the authors test the misconception in a specific target group? Otherwise, the existence of the misconception is not empirically proven at all or limited to individual subjects (e.g., through interviews).

B. Phase 2: Conducting the SLR

The process of conducting the SLR is shown in Figure 2 as *Phase 2* of the overall process.

1) Stage 1: Conducting the Automated Search: For the search – if possible – use of extended/advanced search functions, wildcards (e.g., “misconception*”), and Boolean operators is made in order to be able to exploit the predefined syntax of the query (see String 1). Nevertheless, the string must be adapted to the options of the search engine. Care is taken to ensure no semantic changes take place.

The SQ is limited to document title and abstract, as recommended by [39, p. 2050] as well as others. (Deviations from this definition, due to the search options of the individual databases, are documented accordingly in the evaluation in Section III). The reason for this is that both metadata are already indicators of the relevance of a publication. Note: At this point, IC.3 is not completely applicable, since MCs are not specifically mentioned in title & abstract, but it is checked, whether the contribution is explicitly about misconceptions.

2) Stage 2: Applying the In-/Exclusion Criteria: The relevance of a publication is determined in a two-stage process (see Figure 2, *Stage 2*). First of all, the title and abstract are examined and evaluated on the basis of the predefined criteria. These provide enough information to decide whether a publication encompasses insights of interest; in doubt they were included. The papers included are then rechecked regarding the in-/exclusion criteria; this time considering the full text.

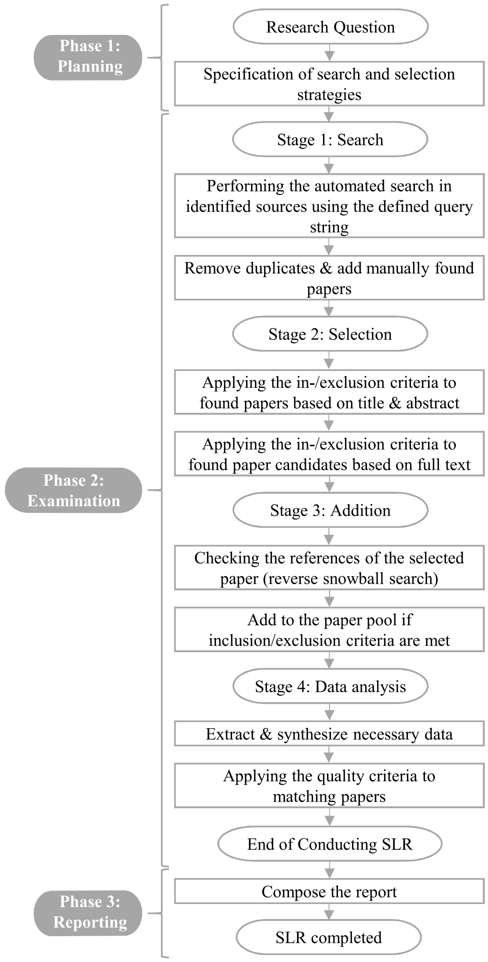


Fig. 2. Flowchart of the SLR process (based on [25] [26] [39])

3) Stage 3: Backward Snowballing: Once *Stage 2* is completed, "the references of the selected papers [are] reviewed and any missing candidate papers [are] assessed against the inclusion/exclusion criteria" [39, p. 2052] as well; this is referred to as 'backward snowballing'.

4) Stage 4: Data Analysis: To assess the quality of the methods and results in the gathered publications, quality criteria have to be predefined against which to assess the data extracted and synthesized.

III. Phase 3: Results of the SLR

The results of the coarse search based on the selection criteria (Section II-A3) applied to titles and abstracts (Section III-A) and the detailed search using full texts (Section III-B) are presented. Additionally, the results of the analysis of the MCs found in the selected publications is shown in Section III-C, which is based on the QCs (Section II-A4).

A. Results: Coarse Search

The automated search has been completed between April, 30th and May, 1st 2020. Since the search was not limited to a date range, the review process timewise included every research found, covering papers as of 1970. Table III illustrates the number of matches ($n = 2,158$) initially received through the SQs. Excluding data sets that contained entire proceedings/compilations instead of contributions as well as duplicates, results in $n = 1,481$. Finally, after applying the inclusion and exclusion criteria to title and abstract, $n = 128$ papers/articles can be identified as potentially relevant to our interest. Therefore, only these are considered in the next step, in which the full text of these publications is considered.

Duplicates could be localized both internally – within the results of the same SQ, within the same database, or overlaps between English and German SQs – and externally – between the results of different search engines. The number of duplicates can be seen in Table IV including multiple mentioning, as papers might be found in multiple databases. (Therefore, the sums are not equivalent with the numbers of duplicates in Table III.)

TABLE IV
Number of Duplicates

	IEEE	ACM	SD	SC	SL	ERIC	WoS	GS	arXiv	dblp
IEEE	15	32		222		1	2	15	5	12
ACM		54		111				9	4	7
SD			18	60				4		2
SC				53		7	4	31	10	21
SL					0					
ERIC						3		3		
WoS							0	3		4
GS								17		19
arXiv									8	2
dblp										4

TABLE III
Summary of SLR Results After Applying In-/Exclusion Criteria on Title & Abstract

		Search Engines										Sum
		IEEE	ACM	SD	SC	SL	ERIC	WoS	GS	arXiv	dblp	
Results of English SQ		250	410	93	847	0	29	4	87	257	41	2,018
Results of German SQ		16	54	7	46	0	3	0	7	2	5	140
Sum of Search Results		266	464	100	893	0	32	4	94	259	46	**2,158**
No Papers (e.g., Proc.)		2	2	0	53	0	2	0	34	6	0	99
Duplicates		15	85	18	383	0	10	4	16	18	29	578
Balance without Duplicates		249	377	82	457	0	20	0	44	235	17	1,481
IC.1a:	English	249	352	81	442	0	20	0	40	231	10	1,425
IC.1b:	German	0	0	0	5	0	0	0	1	0	0	6
IC.2:	SE Discipline-Specific	223	253	65	381	0	18	0	34	162	7	1,143
IC.3:	Misconceptions	30	60	4	43	0	8	0	6	5	2	158
EC.1:	Contribution Type	0	1	0	0	0	1	0	0	0	0	2
EC:	No Information	0	0	0	1	0	0	0	2	0	0	3
Paper Candidates		**29**	**40**	**4**	**37**	**0**	**6**	**0**	**5**	**5**	**2**	**128**

TABLE V
SUMMARY OF SLR RESULTS AFTER APPLYING IN-/EXCLUSION CRITERIA ON FULL TEXTS

	Search Engines										Sum
	IEEE	ACM	SD	SC	SL	ERIC	WoS	GS	arXiv	dblp	
Paper Candidates (see Table III)	29	40	4	37	0	6	0	5	5	2	128
IC.1a: English	29	40	4	37	0	6	0	5	5	2	128
IC.1b: German	0	0	0	0	0	0	0	0	0	0	0
IC.2: SE Discipline	29	40	4	37	0	6	0	4	5	2	127
IC.3: Mention Misconceptions	5	3	1	2	0	0	0	1	3	0	15
Papers Found	**5**	**3**	**1**	**2**	**0**	**0**	**0**	**1**	**3**	**0**	**15**
Backward Snowballing	27	5	0	2	0	0	0	1	4	0	39
Already Included in SLR	2	1	0	1	0	0	0	0	2	0	6
After Applying Selection Criteria	0	0	0	1	0	0	0	0	2	0	3
Result	**5**	**3**	**1**	**3**	**0**	**0**	**0**	**1**	**5**	**0**	**18**

B. Results: Full Text Search

Proceeding further, the predefined inclusion and exclusion criteria are then applied to the paper candidates based on the full text of the contributions. This results in $n = 15$ papers that match the criteria (see Table V). Papers are excluded that cover the topic 'misconception', but did not explicitly mention at least one statement the respective authors refer to as an MC concerning the topic SE (cf. IC.3). The subsequent backward snowball search – based on the adequate papers found – reveals some additional publications that have been checked against the inclusion/exclusion criteria listed as well. Summing up, a total of $n = 18$ papers are found (see Table V) that are of interest to the research question of this SLR.

Through the selection process in *Stage 2* and Backward Snowballing in *Stage 3* as a whole, we double-checked the contributions by assessing each paper. As Kitchenham et al. suggest, publications are included if we cannot make a consensual decision [39, p. 2052].

The matching papers found ($n = 18$, shown as the result in Table V) are listed below:

- IEEE: [42]–[46]
- ACM: [47]–[49]
- Science Direct: [50]
- SCOPUS: [51] (cites and covers the myths of the primary source [52] and 7 new statements) [52] [53]
- Google Scholar: [54]
- arXiv: [55] [56] (is included in [41] and thus not considered further) [40] (is the basis for [41]) [41]; and thus considered together, covering 21 MCs in total) [57]

C. Results: Misconceptions Found

Within the publications named, a total of 167 individual statements (see Table VI; without cross-references) are declared as misconceptions by the respective authors. The MCs gathered should be evaluated by assessing the quality of the publications in order to determine the capacity of the findings, using the quality criteria from Section II-A4.

The coding of the subcategories of the quality criteria was not determined in advance, but developed during the analysis based on and close to the available data; i.e., the publications themselves. The following subcategories are considered as high-quality (see grey marking in Table VI):

QC.1 *Traceability*: A primary *study* as well as the *reference* to quotable publication(s), in which the MC(s) were found is defined as satisfying scientific claims. In contrast, no indication is insufficient.

QC.2 *Validation*: The conception has to be *empirically confirmed* as "at odds with modern scientific theories" [6, p. 2] to be a *mis*conception. Whereas, a rejection, an explanation by the author(s) or reference(s) that the statement given is supposed to be a misconception is no sufficient evidence for validation. This is also due to the fact that MCs exist in all ages, from primary

TABLE VI
SUMMARY OF MISCONCEPTIONS FOUND IN THE FULL TEXTS USING THE QUALITY CRITERIA

		Papers Found																	Sum
		[38]	[39]	[40]	[41]	[42]	[43]	[44]	[45]	[46]	[47]	[48]	[49]	[50]	[51]	[53]	[54]	[55]	17
Misconceptions explicitly named		16	12	12	7	6	4	5	12	4	7	7	10	36	4	10	21	4	167
QC.1:	- Study						4							36		21 (in [40])			51
Traceability	- Reference(s)	15	6			6													37
	- No Indication	1	6	12	7			5	12	4	7	7	10		4			4	79
QC.2:	- Empirically Confirmed								12							8 (in [41])			20
Validation	- Empirically Rejected															2 (in [41])			2
	- Reference(s)	6	6		5														17
	- Only based on Explanation						4			4	7	7	10		4				36
	- No Indication	10	6	12	2	6		5						36		11		4	92
QC.3:	- Practitioners	16				6	4									21 (in [40])			37
Occurrence	- Undergraduates		12	12					12										36
	- No Indication				7			5		4	7	7	10	36	4	10/21 (in [41])		4	94
Intersection (of rows marked)									12							8			20

TABLE VII
LIST OF MISCONCEPTIONS MATCHING THE QUALITY CRITERIA

Topic(s): Project	Process Models	Team Skills	Reuqirements	Implementation	Defects	Documentation	Misconception	Reference(s)
	X						A defined software process is only important when you are working with people who are less skilled.	[49, (1)]
		X					A good software developer will often choose to work alone on a project in order to get it done faster.	[49, (2)]
	X	X					When you have a team of good programmers who work well together, a software process will usually get in the way.	[49, (3)]
				X			My code should take advantage of the implementation details in other code.	[49, (4)]
			X				It is expected that clients will describe their requirements accurately before a team begins programming.	[49, (5)]
				X			As a software developer, most of my time will be spent designing and implementing new algorithms and data structures.	[49, (6)]
X							Most of the time when I start a new programming task in industry, I will be working on a new project.	[49, (7)]
				X			Developers do not need to know the high-level context of the system; this allows them to concentrate on their task.	[49, (8)]
					X		A software project is successful only if it ships with very few known defects.	[49, (9)]
						X	Software engineering is about producing lots of documentation on the requirements and implementation of the project.	[49, (10)]
	X	X	X				Process, requirements, and team-management are important to business majors, not software developers.	[49, (11)]
				X			The majority of the cost of a successful software project will be the initial implementation effort.	[49, (12)]
					X		A file with a complex code change process tends to be buggy.	[40, (S2)], [41, (B1)]
					X		A file that is changed by more developers is more bug-prone.	[40, (S14)], [41, (B2)]
					X		A file with more added lines is more bug-prone.	[40, (S4)], [41, (B3)]
					X		Recently changed files tend to be buggy.	[40, (S7)], [41, (B4)]
					X		Recently bug-fixed files tend to be buggy.	[40, (S10)], [41, (B6)]
					X		A file with more fixed bugs tends to be more bug-prone.	[40, (S11)], [41, (B7)]
					X		A file with more commits is more bug-prone.	[40, (S12)], [41, (B8)]
					X		A file with more removed lines is more bug-prone.	[40, (S13)], [41, (B9)]

level to university and even experts and professors can hold them themselves [58, p. 9, 11].

QC.3 *Occurrence in the population*: Practitioners MCs are included, as it is very likely that students have them as well, if they can be encountered in professionally experienced. However, no indication of occurrence in the population can initially only be interpreted as a presumption.

The intersection of the QCs results in $n = 20$ misconceptions (Table VI). Yet, the papers [40] and [41] only deal with the topic 'defect prediction', the authors of [49] look at SE covering the software life cycle more holistically; see thematic structuring in Table VII.

Note: [49] would actually not be included in the intersection, as it is not explained where the MCs come from (QC.1). But the authors validated them (QC.2) and tested their occurrence concerning students (QC.3). Thus, the MCs listed are hypotheses, that have been empirically confirmed; thus, nevertheless, they are included in the intersection.

IV. DISCUSSION

Several aspects regarding the SLR should be remarked upon concerning the validity of the methodology as well as the results per se.

A. Methodology: Threats to Validity

First, one significant limitation is the broad number of synonyms for 'misconception'; it is almost impossible – despite all efforts – to ensure that all relevant papers are found.

Second, we used the four-eyes principle to proceed and discussed to achieve consensus, but enclosed papers causing persistent disagreement. However, this is not an ideal process, affecting reliability of assessment and evidence of results.

Third, a limitation is that own publications could turn out to be matches in the SLR, which must be handled objectively. This can result in a systematic error. It is therefore noted that authors of this paper also authored the publication [54].

B. Discussion of Results

Regarding the results of the SLR, it is noted that the cut of 2,158 publications to merely 3 [40] [41] [49] of interest identified is immense. As a result, it could be assumed that the search (engines/query) or the selection (in-/exclusion/quality criteria) are inadequate. However, this contradicts that ...

- ... SE didactics are still developing.
- ... the consideration of another database (Section I-B, [11]) also indicates that little research is available to date.
- ... other authors report the same for the adjacent field of computer sciences: "At present, hardly any empirical data concerning the issue of expectations and prior knowledge [...] in informatics [...] are available" [59, p. 143].

V. APPROACH: UNDERGRADUATE SURVEY ON MISCONCEPTIONS IN SE

This section covers the methodology to detect Misconceptions through GTM (Sec. V-A) and downstream to construct Misconception items for the catalogue (Sec. V-B).

A. Detection of Misconceptions through GTM

After presenting the process of conducting the SLR, the following sections explain the the second pillar of this contribution – the survey carried out: its design, the process as well as the results.

1) Design of the Undergraduate Survey: Due to several reasons and criteria, the decision was made to operationalise Straussian GTM and apply it in a tailored implementation [60] to detect undergraduates Misconceptions in SE as a step to build upon. The main causes are the explorative nature due to the research gap detected (see Section III; [1], interpretative reconstructing from the statements, and openness towards the data to grasp the undergraduates' understanding. The Grounded Theory (GT) Methodology can be described as an iterative process that is constantly enriched by new data. This was done after a careful interpretation why a qualitative, explorative and data-driven research approach and GTM, in particular, seems to be a promising research approach regarding the research goal and the research already available. Furthermore, a review of the three main variants of GT (Glaserian [61]/Straussian [62]/Constructivist GT [63]) has been done to determine relevant points of similarity and difference, to define the best solution possible for this application scenario. This has also been done since Stol et al. demonstrated the "method slurring" [64] that GTM is suffering from through a critical examination of the usage of GTM in SE [60].

The central research question of the GT study therefore is [65]: What Misconceptions do undergraduates have concerning Software Engineering?

2) Data Elicitation and Analyses: The study was carried out as follows with reference to key concepts of GTM:

a) Theoretical Sensitivity: This concept [66] has an ambivalent significance to the research process. On the one hand, contextual knowledge is essential to enable the researchers to localize concepts in and abstract them from the data. On the other hand, the assumptions and results may harbour the risk of a bias. Due to the academic background of the researchers[1] and due to a previously conducted small literature review on (Mis-)conceptions research in SE (in order to detect and formulate the research gap; see [65]), there is a high theoretical sensitivity on the present study. However, maybe in contrast to other research approaches, the SLR shown in Sections II-IV was not conducted in advance (cf. Section V-A2b). In order to deal with this tension field, this information has been disclosed here. This transparency as well as a reflection/discussion on this possible problem source is a way of disarming it, using traceability and verifiability of the findings.

b) Data Collection: Undergraduates' statements about SE have been elicited using interviews as a form of primary data acquisition. To recruit the individual interviewees, project members from different Universities of Applied Sciences – professors as well as research associates – set up the contact. The 16 interviews (14 male/2 female) conducted took place between June 14, 2017 and December 12, 2018.

c) Theoretical Sampling: Since GTM follows simultaneous data collection and analysis, the sample selection is driven by *Theoretical Sampling*. In this study, different criteria have been used to implement minimum and maximum contrast, which evolved through the research progress: University or place of study, degree program, and study progress.

As seen from Table VIII, the interviews have been conducted in waves of small cohorts (N = 3...6), which – per wave – represent intra-similar data (minimum contrasting). Due to the study purpose, it is not always possible and useful to progressively collect data, but at certain times several interviews are conducted en bloc, because of a specific point in the semester that should be captured; e.g., in the beginning, the focus was on the target group entering the module(s) in SE. Interviewees were recruited through announcements made in courses by lecturers (colleagues at the different Universities of Applied Science). Participation was voluntary.

[1]The researchers received Master degrees in computer sciences as well as engineering. Furthermore, they have worked in discipline-based educational research for several years as research associates.

TABLE VIII
UNDERGRADUATE INTERVIEW PARTNERS IN THE SAMPLE

Wave	N	University	Degree Program[a] CS	MT	EE&IT	Study Progress
1	4	C	X			2nd/3rd Semester[b]
2	3	A		X		2nd/3rd Semester[b]
3	3	C	X			Freshman
4	5	B			X	Freshman
	1			X		
Sum	16		7	4	5	

[a] CS = Computer Sciences, MT = Mechatronics, EE&IT = Electrical Engineering and Information Technology
[b] The undergraduates have been asked in advance to module(s) dealing with SE, which take place in the 3rd/4th semester, depending on the degree program, but have already gained some programming experience and other basics.

Despite their divergent characteristics, they stated similar statements that could be reconstructed as Misconceptions, concerning a strong focus on programming, testing, and greatly simplifying requirements engineering. This gave rise to the question: Is the reason for this possibly to be found in the previous programming training they took part in? Therefore, the decision was made to interview freshman students as well and contrast their statements to the already established code system (encompassing Misconceptions).

By contrasting the individual's data with interviews of similar and dissimilar interviewees, the theoretical epistemological value could be maximized. This constant comparison led to a theoretical saturation, which was reached when, despite additional data, no new information emerged from the data.

d) Guided Interviews: Data acquisition took place using guided interviews that encompassed questions like:

- *Have you already gained experience in the field of "Software Engineering" outside of your studies?*
- *How is software actually created? / How do you imagine software being developed? / What does this process look like?*
- *What happens in the development process until the software is finished?*
- *When is a Software Engineer's job done?*
- *How are the tasks divided into a project in terms of time?*
- *Which people are involved?*
- *What skills should a "good" typical Software Engineer have? Why?*

The undergraduates have been interviewed individually. The interviews lasted between 15 and 44 minutes (average = 23.75; SD = 7.05) depending on the undergraduates' openness towards the interview situation, readiness to leave their personal comfort zones, willingness to speak, and the number of further inquiries by the interviewer to meet the scope of the interview guideline and get a perception of the interviewees understanding. At the beginning of each interview, undergraduates were explicitly assured that this is not a test, but rather a feedback to get to know their conceptions and ideas in order to be able to adapt the courses. The interviews were recorded for documentation purposes.

e) Transcription: Transcribing the audio-recordings resulted in research generated documents – the basis for the analyses. This was done using defined transcription rules following [67]. The transcripts cover 131 pages (M = 8.19; SD = 2.65) and 1,832 paragraphs (change of speaker) in total.

f) Coding Procedure: During the first step, the open coding, information is "broken down analytically" [68, p. 127]. The aim is to grasp the essence of the data, to find relevant information and capture it in codes – initially rather descriptively –, but also by using inductive reasoning to abstract and conceptualise. In contrast to other coding paradigms, where coding schemes are for example predefined, in GTM the codes emerge from the text. As coding and Constant Comparison evolve, codes, concepts, and memos can be abstracted and conceptualised on and on through axial and selective coding. This also comprises finding, arranging, and relating concepts and key categories (see Coding Paradigm [62]).

Through this process, Misconceptions of SE could be concluded through the use of GTM as an interpretative data analysis from the statements made by the undergraduates in the interviews using a Computer-Assisted Qualitative Data Analysis Software (in this case: MAXQDA [69]).

g) Investigator Triangulation: Multiple observers or interviewers are used to detect or minimize biases by the researcher. This can be done collaboratively or independently. How this is done in research practice and documented in publications has been inspected in a literature review by Archibald [70]. This revealed that out of 186 articles only 166 use the words "investigator" and "triangulation" incidentally in one sentence, but do not explain its usage [70].

In the research described in this paper, three researchers act as investigators, who are all research associates from the field of engineering and informatics[1] in order to be able to identify Misconceptions in SE; the disciplinary background is of prime importance (ref. Theoretical Sensitivity in Section V-A2). Two of them already conducted qualitative and quantitative research. Through the analysis and interpretation a consensus procedure for findings is applied. This is done in order to minimize the (constructivist [58, 11]) bias of a single coder and objectify[2] the results as well as maximize the richness of data and the reliability.

B. Construction of Misconception Items for the Catalogue

As seen elsewhere, concerning disparate [71], [72] as well as SE related disciplines [19], [22], [73] (and likewise: [20], [24]), Misconceptions are often formulated and documented as clauses of statement, e.g.:

- Physics: Driving a car at high speed consumes less fuel as it takes less time to drive [71, p. 17].
- Biology: Growing means to expand/enlarge [72, p. 371].
- Programming: A variable can hold multiple values at a time [19, p. 153].

Therefore, the concepts found encompassing Misconceptions in SE have been used to phrase declarative sentences as seen above (cf. Tables IX and X, 2nd column each). Care was taken to ensure that the formulations are as close as possible to the statements made by the undergraduates in the interviews (i.e., grounded in the data).

VI. Results

First it has to be noted that the catalogue was translated from German, which means additional transfer and therefore room for interpretation, although that is precisely why care was taken to stay as close as possible to the German formulations. However, one has to be aware that this is not only a result of the translation work, but also insists on a single language. This means that every person individually interprets every single word. From a constructivist perspective, everyone construes every word individually against the background of one's own knowledge, experiences, volition, motivation, and interest slightly differently from one another. Ergo, no identical or congruent statement will arrive at every single individual, regardless of which wording is used.

Tables IX and X (see next two pages), presenting the 69 items that have been reconstructed from the interviews with undergraduates (cf. Section V-A) and conclusively formulated accordingly (cf. Section V-B).

VII. Discussion

A. Implementation of GTM

Due to constructivism as stated before, not only undergraduates or pupils might hold Misconceptions concerning various topics and disciplines, but also researchers, teachers, lecturers, and professors. As there is – to the best of our knowledge – no possibility to avoid this, the approach of investigator triangulation was chosen in order to minimize the (constructivist) bias of a single coder and objectify[3] the results as well as maximize the richness of data and the reliability.

The established catalogue makes no claim to completeness, as also the interview guide developed and used is non-exhaustive. It should be noted that there will never be a complete list of misconceptions, since on the one hand there will always be something overlooked, which is human and impossible due to constructivism, and on the other hand – also following the constructivist paradigm – the conceptions of a single person is so extremely individual that a qualitative explorative investigation of "every" individual would be required. Given the fact that research on conceptions in Software Engineering has so far represented a research gap (cf. Sections II-IV; [1]), this study should embody the starting point on the path the natural sciences have already taken over the past decades in Discipline-Based Educational Research.

Additionally, the interviews have been conducted before a module in SE to get to know undergraduates (mis-)conceptions

[2]Of course, the one objective truth or reality can never be established, but in using multiple investigators/perspectives, an alternative to validation can be pursued.

[3]Of course, the one objective truth or reality can never be established, but in using multiple investigators/perspectives, an alternative to validation can be pursued.

TABLE IX
CATALOGUE OF UNDERGRADUATES' MISCONCEPTIONS ABOUT SOFTWARE ENGINEERING (PART I)

#	Items	Reference(s) for Comparison
A	**Software Development Process**	
A1	The phase of conception of a software project consists of organisation and project planning.	
A2	At the start of a software project, a basic source code is needed as a foundation.	
A3	The development of software is made up of the problem definition and its programming.	[49, p. 33; items 6 & 12]
A4	Software Engineering consists of programming and testing the software on its target hardware.	[49, p. 33; items 6 & 12]
A5	A FMEA (Failure Mode and Effects Analysis) is done after implementation.	
A6	Iterations and Optimization are seldom part of the Software Engineering process.	[53, p. 32; all items]
A7	The design phase in a software project is completed quickly.	
A8	Operation and Maintenance are not part of Software Engineering.	
A9	The process of software development is generally linear and sequential.	[44, p. 11; item 4L], [53, p. 32; all items]
A10	Both the development idea and the exact procedure for creating software are specified externally.	
B	**Requirements**	
B1	The customer requirements are fully present for the Software Engineer at the start of the project.	[49, p. 33; item 5]
B2	The work of the people employed in Software Engineering begins only when the specific task is available in detail.	[49, p. 33; item 5]
B3	The software project starts only when the complete assignment is available.	[49, p. 33; item 5]
B4	Usually customers know their requirements precisely.	[49, p. 33; item 5]
B5	As a rule, the requirements of business customers are unambiguous.	[49, p. 33; item 5]
B6	The customer's wishes are incorporated into a new product, but the basic product idea comes from within.	
B7	The requirements elicitation is the quickest part of a project.	
B8	Software normally only has to be implemented according to the conceptions of the Software Engineer.	
B9	The Programmer decides about additional functionalities of the software at his/her own discretion.	
C	**Implementation**	
C1	In a software project the implementation is the longest phase.	[44, p. 11; item 4B], [49, p. 33; items 6 & 12]
C2	A Software Engineer programmes on his/her own.	[49, p. 33; item 2]
C3	The implementation of a software project starts with the programming.	
C4	Software development is generally aimed at a specific hardware.	
D	**Testing and Bug Fixing**	
D1	The test phase can be concluded when no more bugs are found.	[44, p. 11; item 4M], [49, p. 33; item 9]
D2	In Software Engineering the work is successfully completed when there are no more defects, according to the customer's assessment.	
D3	The Software Engineer's job is successfully done when there are no more errors in the source code.	[44, p. 11; item 4M], [49, p. 33; item 9]
D4	Good software means everything works.	
D5	The source code of good software has to be free of errors.	[44, p. 11; item 4M], [49, p. 33; item 9]
D6	A Software Engineer does not conduct software tests.	
D7	Software is tested by trial and error.	
D8	A Software tester corrects the found errors him-/herself.	[47, p. 229; item 2], [49, p. 33; item 6]
E	**Scope of Delivery**	
E1	The source code is in general part of the scope of delivery.	
E2	Software development is completed when the artifacts are handed over to the customer.	

TABLE X
CATALOGUE OF UNDERGRADUATES' MISCONCEPTIONS ABOUT SOFTWARE ENGINEERING (PART II)

#	Items	Reference(s) for Comparison
F	**Comparison: Small – Large Software Development Companies**	
F1	Small software companies normally have less experience than big companies.	
F2	As a rule, small software companies work faster than big companies.	
F3	Big software companies work faster than small ones.	
F4	In big software companies software is optimized faster than in small companies.	
F5	In comparison to small software companies, employees in big companies are more skilled.	
F6	Large software companies invest more work into a software product.	
F7	Software developed by small companies does not have to run as stable as software from large companies.	
F8	In big software development companies more time is spent on testing a software.	
F9	Primarily large software development companies conduct market research.	
F10	Large companies generally distribute mass-produced products.	
F11	Software product development in large companies encompasses more steps than in small companies.	
G	**Comparison: Mass Products – Individual Commissioned Work**	
G1	When developing a mass product, no requirements elicitation is carried out, but the acceptance rate is evaluated retrospectively.	
G2	A commissioned work does not have to be as stable as a mass-produced product.	
G3	A mass product offers more functionalities than a commissioned work.	
G4	When developing a mass product, it has to be designed more intuitively than a commissioned work.	
G5	Mass products are generally developed independent of an operating system.	
G6	Mass products are typically more configurable than product from individual contracts.	
G7	When developing a mass product, what the software developers themselves consider appropriate is implemented.	
G8	The GUI of a mass-produced product is designed differently compared to that of commissioned work.	
G9	When developing a commissioned work, it is intended for a person with in-depth knowledge of the product.	
H	**Comparison: New Development – Extending Existing Software**	
H1	Developing a new software is more work intensive than to extend an existing one.	
H2	When developing a new software, you can start the programming immediately.	
H3	An extension is more complex if the basis is not an in-house development.	
H4	When software is well documented, it is easier to expand than to begin a new development.	
H5	When software is expanded its intended application cannot be changed anymore.	
H6	Today everything is an extension. Nothing is developed from scratch.	
I	**Working Life of a Software Engineer**	
I1	The main task of a Software Engineer is programming.	[43, p. 4; item A], [44, p. 11; item 4B], [49, p. 33; items 6 & 12]
I2	The job description of a Software Engineer does not differ from that of a computer scientist.	[43, p. 4; item A]
J	**Customer Contact**	
J1	Customer contact is normally not performed by Software Engineers.	[43, p. 4; item D], [49, p. 33; item 11]
J2	In larger companies customer contact is in general not undertaken by Software Engineers.	[43, p. 4; item D], [49, p. 33; item 11]
J3	Customer dialogue is not part of Software Engineering itself.	[43, p. 4; item D], [49, p. 33; item 11]
K	**Overview**	
K1	An overview of the complete software development process is not required for working as a Software Engineer.	[49, p. 33; items 8 & 11]
K2	For working as a Software Engineer, it is sufficient to be familiar with your assigned task.	[49, p. 33; items 8 & 11]
K3	A Software Engineer does not need to have an overview of the finances of the project.	[49, p. 33; items 8 & 11]
L	**Miscellaneous**	
L1	Compared to nontechnical users, engineers can handle software even if it is not designed intuitively.	
L2	Programmers normally only have contact among each other.	

as they enter. Therefore, the interviews undertaken covered questions on a relatively coarse-granular level (see Sec. V-A).

B. Validaion of Results

The Misconceptions found have also been validated by 13 experts, which is explained and preseted in detail in the publication [2]. This is done for mainly two reasons:

- First, all individuals are subject to constructivism, so researchers and lecturers have to be aware that not only students/undergraduates/learners, but also researchers and lecturers might have Misconceptions – even concerning our own field of expertise (see "Investigator Triangulation" in Sec. V-A2) as stated by Duit [58, p. 11], who is himself a professor emeritus for the didactics of physics.
- Furthermore, the line between "technically adequate" and "inadequate" can be drawn relatively clearly in relation to Misconceptions in the natural sciences, since these are based on natural laws and rules. To a large extent, this can also be transferred to Computer Sciences fields, such as programming, since language definitions also specify "correct"/"incorrect" (e.g., syntactically speaking).

  In Software Engineering, on the other hand, this line does not seem to be so clear and easy to draw [74, p. 8 f.] [75, p. 12].

  First, SE does not follow any nomothetic premises, unlike the natural sciences, which aim to transfer their knowledge into regularities and thus try to explain the world through natural laws and rules – often mathematically [74, p. 8 f.] [75, p. 12]. There are simply no formulas or similar, as in the natural sciences and mathematics, which when applied lead to proven and guaranteed correct results [75, p. 12].

  Second, neither in SE nor in many other areas of computer science (e.g., programming), *one correct* (sample[4]) solution exists, but a whole solution space [76, p. 437], which is available. On the one hand, this is due to the human factor, which, for example, has a decisive influence on the result of a (large) software development project. On the other hand, there is also the intrinsic plurality of possible solutions that result from the implementation alternatives, so that subsequently in SE a solution path will probably never be reproduced identically twice [75, p. 12 ff.].

VIII. Analysis and Comparison

To combine the two pillars explained in detail beforehand, this third part is intended to analyse and compare the findings of both elicitations. It has to be noted that the aim is not to conduct a one-to-one matching, but to look on context-related proximities, to search for same or similar underlying ideas and thinking patterns (see Section VIII-A) as well as – partly based upon this – have a look at the thinking patterns and make considerations concerning a root cause analysis on MCs in SE (see Section VIII-B).

[4]"Sample solution" based on the didactic teaching context.

A. Comparison and Contrasting

Besides the MCs found in the interviews, Tables IX and X also encompass a third column covering references of MCs found through the SLR, which have a similar meaning or same underlying concept. For example, B2 (Table IX: *The work of the people employed in Software Engineering generally begins only when the specific task is available in detail.* is relatively similar to "It is expected that clients will describe their requirements accurately before a team begins programming." [49, p. 33; item 5]. The comparison of the survey results and the SLR results has been done on a meta level, since neither the same wording nor the exactly identical meaning is found when contrasting the item lists. This compasion has been done on the basis of the 69 items constructed from the MCs found through the interview study on the one site and the MCs identified in the SLR before the application of the quality criteria on the other site. This has been done, since the quality criteria have been established and used to filter for evidence-based MCs, which does not mean that these might be MCs although they do not meet the criteria.

The comparison shows several overlaps – especially concerning some items that also have overlaps within the SLR; e.g., A9, C1, I1, J1-3.

Finally, it should be noted: The fact that several items in Tables IX and X have not been "found" in the SLR and vice versa does not imply that the interviewees do not hold them. This may be due to the various reasons:

- Oral articulateness possibilities (cf. discipline-specific language and terminology)
- Open ended questions in die interviews
- Undergraduates are new to SE
- Qualitiative interviews assert no claim for completeness of results

B. Further Analyses

What can also be seen is that even larger clusters/categories can be formed through further abstraction; supplemented by many MC candidates from the SLR. One very large topic are complexity reductions such as (excerpt):

- One-dimensional (restricted) programmer perspective: e.g., A3, A4, A7, A8, C1, C3, [43, p. 4; item A], [44, p. 11; item 4B], [49, p. 33; items 6, 8 & 12]
- Linearity and sequentiality of the process: e.g., A6, A9, [44, p. 11; item 4L], [53, p. 32; all items]
- Simplification of requirements elicitation: e.g., B1-B5, B7, [49, p. 33; items 5],
- Focusing on a sub-area of the SE: e.g., K1-K3, [49, p. 33; items 8],

These thinking patterns could also be interpreted as possible causes for MCs.

IX. Summary & Outlook

The paper's purpose, to identify and analyse misconceptions in SE to use these findings in higher education, has been pursued using a systematic literature review for already known MCs as well as a interview study getting new insights.

Looking at the SLR, predefined search queries have been applied to search 10 databases before the publications have been filtered using the selection strategy described. Out of 2,158 publications, 18 could be identified as appropriate for the selection criteria. These contain 167 statements, which the authors of these papers refer to as misconceptions. 20 of them met the quality criteria specified; i.e., only 3 publications cover valuable data.

To conclude, the results show that currently evidence-based research on misconceptions in SE is limited. So, in addition a primary study to identify misconception in SE is indispensable before addressing them.

Therefore, the procedure of qualitative data elicitation, analysis, and interpretation has been exploited to detect Misconceptions in SE through Straussian's Grounded Theory Methodology [62]. Building on these findings, a brief look was taken at the subsequent formulation of misconceptions in accordance with previous literature (see Sec. V-B).

Bringing the results of both approaches and data sources together, a comparison on a meta level has been conducted. As a result, it can be stated that there are similarities concerning thinking patterns, which have to be further investigated.

As an outlook, in order to strive to achieve sustainable learning (in higher education) – as displayed in the introduction – another step is taken to get an understanding of undergraduates conceptions in order to do justice to the learners' "points of departure" [3, p. 6] and misconcetpions as learning obstacles [10].

ACKNOWLEDGEMENT

The present publication is based on the work of the EVELIN project funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01PL17022B. The authors are responsible for the content of this publication.

The author thanks the reviewers of the ICSEA 2020 conference and the IARIA Journal to have been selected for the invitation to submit this paper; beyond the very useful comments that have contributed to enhance both the original and the extended version of this paper.

Above all, I would like to thank my two co-investigators during the undergraduate survey [2], Rebecca Reuter and Sabrina Jahn, both from the OTH Regensburg, Germany as well as Theresa Stark for her support.

Also, I would like to thank Nermin Saray for her support in conducting the SLR [1].

REFERENCES

[1] C. Gold-Veerkamp and N. Saray, "A Systematic Literature Review on Misconceptions in Software Engineering," in *Proceedings of the Fifteenth International Conference on Software Engineering Advances (ICSEA)*, 2020, pp. 1–8.

[2] C. Gold-Veerkamp, "Validated Undergraduates' Misconceptions about Software Engineering," in *Proceedings of the 2021 IEEE Global Engineering Education Conference (EDUCON)*, Institute of Electrical and Electronics Engineers, Ed. Piscataway, NJ: IEEE, 2021, pp. 609–618.

[3] R. Duit, "Science education research internationally: Conceptions, research methods, domains of research," *EURASIA Journal of Mathematics, Science & Technology Education*, vol. 3, no. 1, pp. 3–15, 2007.

[4] R. Duit, H. Gropengießer, U. Kattmann, M. Komorek, and I. Parchmann, "The model of educational reconstruction," in *Science Education Research and Practice*, D. Jorde and J. Dillon, Eds. Springer, 2012, pp. 13–37.

[5] J. P. Smith, A. A. diSessa, and J. Roschelle, "Misconceptions reconceived: A constructivist analysis of knowledge in transition," *Journal of the Learning Sciences*, vol. 3, no. 2, pp. 115–163, 1994.

[6] J. R. Read, "Children's misconceptions and conceptual change in science education," 2004, [retrieved: 09, 2020]. [Online]. Available: http://acell.chem.usyd.edu.au/Conceptual-Change.cfm

[7] I. Diethelm and S. Zumbrägel, "An investigation of secondary school students' conceptions on how the internet works," in *Koli Calling International Conference on Computing Education Research*, M.-J. Laakso and R. McCartney, Eds. ACM Press, 2012, pp. 67–73.

[8] S. Todtenhaupt, *To develop an understanding of chemistry in schoolchildren: An investigation into the redox topic at a high school. (Original title: "Zur Entwicklung des Chemieverständnisses bei Schülern: Eine Untersuchung zur Redox-Thematik an einem Gymnasium" [German]).* Frankfurt a.M.: Lang, 1995.

[9] A. Taylor Kujawski and P. Kowalski, "Naïve psychological science: the prevalence, strength, and source of misconceptions," *The Psychological Record*, vol. 54, pp. 15–25, 2004.

[10] R. Reuter, F. Hauser, C. Gold-Veerkamp, J. Mottok, and J. Abke, "Towards a definition and identification of learning obstacles in higher software engineering education," in *Annual International Conference on Education and New Learning Technologies (EDULEARN)*, IATED, Ed., 2017, pp. 10 259–10 267.

[11] R. Duit, "STCSE: Students' and teachers' conceptions and science education," Bibiography, 2009, [retrieved: 09, 2020]. [Online]. Available: http://archiv.ipn.uni-kiel.de/stcse/download_stcse.html

[12] D. N. Perkins and R. Simmons, "Patterns of misunderstanding: An integrative model of misconceptions in science, math and programming," in *2. Int. Seminar Misconceptions and Educational Strategies in Science and Mathematics: Vol. I*, J. D. Novak, Ed., 1987, pp. 381–395.

[13] ——, "Patterns of misunderstanding: An integrative model for science, math, and programming," *Review of Educational Research*, vol. 58, no. 3, pp. 303–326, 1988.

[14] L. Louca and Z. C. Zacharia, "The use of computer-based programming environments as computer modelling tools in early science education: The cases of textual and graphical program languages," *International Journal of Science Education*, vol. 30, no. 3, pp. 287–324, 2008.

[15] J. Confrey, "Misconceptions across subject matter: science, mathematics, programming," in *2. Int. Seminar Misconceptions and Educational Strategies in Science and Mathematics: Vol. I*, J. D. Novak, Ed., 1987, pp. 81–106.

[16] N. Taylor and G. Corrigan, "New South Wales primary school teachers' perceptions of the role of ICT in the primary science curriculum - A rural and regional perspective," *International Journal of Science and Mathematics Education*, vol. 5, no. 1, pp. 85–109, 2007.

[17] R. D. Pea, "Language-independent conceptual "bugs" in novice programming," *Journal educational computing research*, vol. 2, no. 1, pp. 25–36, 1986.

[18] J. Sorva, "Visual program simulation in introductory programming education." Dissertation, Espoo, Aalto Univ., Finland, 2012.

[19] A. Swidan, F. Hermans, and M. Smit, "Programming misconceptions for school students," in *Conference on International Computing Education Research (ICER)*. ACM, 2018, pp. 151–159.

[20] Ž. Žanko, M. Mladenović, and I. Boljat, "Misconceptions about variables at the K-12 level," *Education and Information Technologies*, vol. 24, no. 2, pp. 1251–1268, 2019.

[21] R. Kelter, M. Kramer, and T. Brinda, "Statistical frequency-analysis of misconceptions in object-oriented-programming: Regularized pcr models for frequency analysis across oop concepts and related factors," in *Koli Calling International Conference on Computing Education Research*, M. Joy and P. Ihantola, Eds. ACM, 2018, pp. 6:1–6:10.

[22] S. Holland, R. Griffiths, and M. Woodman, "Avoiding object misconceptions," in *SIGCSE technical symposium on Computer science education*, C. M. White, C. Erickson, B. Klein, and J. E. Miller, Eds. ACM, 1997, pp. 131–134.

[23] M. T. Rücker and N. Pinkwart, ""How else should it work?" A grounded theory of pre-college students' understanding of computing devices," *ACM Transactions on Computing Education*, vol. 19, no. 1, pp. 2:1–2:23, 2018.

[24] I. Diethelm, P. Hubwieser, and R. Klaus, “Students, teachers and phenomena: Educational reconstruction for computer science education,” in *Koli Calling International Conference on Computing Education Research*, M.-J. Laakso and R. McCartney, Eds. ACM, 2012, pp. 164–173.

[25] B. A. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering: Version 2.3,” EBSE Technical Report (EBSE-2007-01), Keele University & University of Durham, 2007.

[26] P. O. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, “Lessons from applying the systematic literature review process within the software engineering domain,” *Journal of Systems and Software*, vol. 80, no. 4, pp. 571–583, 2007.

[27] B. A. Kitchenham et al., “Systematic literature reviews in software engineering – a systematic literature review,” *Information and Software Technology*, vol. 51, no. 1, pp. 7–15, 2009.

[28] A. Bartel, “Conception and development of a DSM-based gamification authoring system to support university teaching. (Original title: ”Konzeption und Entwicklung eines DSM-basierten Gamification Authoring Systems zur Unterstützung hochschulischer Lehre” [German]),” Dissertation, Universität Regensburg, 2018.

[29] IEEE, “IEEE Xplore Digital Library,” 2020. [Online]. Available: https://ieeexplore.ieee.org/Xplore/home.jsp

[30] ACM, “ACM Digital Library,” 2020. [Online]. Available: http://dl.acm.org/

[31] Springer Nature Switzerland AG, “SpringerLink,” 2020. [Online]. Available: https://link.springer.com/

[32] Elsevier B.V., “Scopus,” 2020. [Online]. Available: http://www.scopus.com/

[33] ——, “ScienceDirect,” 2020. [Online]. Available: https://www.sciencedirect.com/

[34] Institute of Education Sciences of the US Department of Education, “Eric – education resources information center,” 2020. [Online]. Available: https://eric.ed.gov/

[35] Clarivate Analytics, “Web of Science,” 2020. [Online]. Available: http://www.webofknowledge.com

[36] Google LLC, “Google Scholar,” 2020. [Online]. Available: http://scholar.google.de/

[37] Cornell University, “arXiv.org: e-Print archive,” 2020. [Online]. Available: https://arxiv.org/

[38] Schloss Dagstuhl and Universität Trier, “dblp: computer science bibliography,” 2020. [Online]. Available: https://dblp.uni-trier.de/

[39] B. A. Kitchenham and P. O. Brereton, “A systematic review of systematic review process research in software engineering,” *Information and Software Technology*, vol. 55, no. 12, pp. 2049–2075, 2013.

[40] Z. Wan et al., “Perceptions, expectations, and challenges in defect prediction,” *IEEE Transactions on Software Engineering*, pp. 1–26, 2018.

[41] N. C. Shrikanth and T. Menzies, “Assessing practitioner beliefs about software defect prediction,” 2020, accepted at ICSE’20, [retrieved: 09, 2020]. [Online]. Available: arXiv:1912.10093v3

[42] P. Devanbu, T. Zimmermann, and C. Bird, “Belief evidence in empirical software engineering,” in *International Conference on Software Engineering (ICSE)*, 2016, pp. 108–119.

[43] M. M. Inuwa and A. Varol, “Intensity of misconception in software engineering,” in *International Informatics and Software Engineering Conference (UBMYK)*, 2019, pp. 1–6.

[44] J. Ivins, B. R. von Konsky, D. Cooper, and M. Robey, “Software engineers and engineering: A survey of undergraduate preconceptions,” in *Frontiers in Education (FIE)*, 2006, pp. MIF–6–11.

[45] B. Özkan and O. Demirors, “On the seven misconceptions about functional size measurement,” in *Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, 2016, pp. 45–52.

[46] J. S. van der Ven and J. Bosch, “Busting software architecture beliefs: A survey on success factors in architecture decision making,” in *Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2016, pp. 42–49.

[47] A. Begel and B. Simon, “Struggles of new college graduates in their first software development job,” *SIGCSE Bull*, vol. 40, no. 1, pp. 226–230, 2008.

[48] D. DeMarco Brown, “Five agile ux myths,” *Journal of Usability Studies*, vol. 8, no. 3, pp. 55–60, 2013.

[49] L. A. Sudol and C. Jaspan, “Analyzing the strength of undergraduate misconceptions about software engineering,” in *International Computing Education Research (ICER)*. ACM, 2010, pp. 31–39.

[50] R. H. Wilcox, “Behavioral misconceptions facing the software engineer,” in *Computer and Information Sciences*, ser. SEN Report Series Software Engineering, J. T. Tou, Ed. Elsevier, 1971, vol. 2, pp. 285–287.

[51] J. P. Bowen and M. G. Hinchey, “Seven more myths of formal methods: Dispelling industrial prejudices,” in *FME’94: Industrial Benefit of Formal Methods*, ser. LNCS 873. Springer, 1994, pp. 105–117.

[52] A. Hall, “Seven myths of formal methods,” *IEEE Software*, vol. 7, no. 5, pp. 11–19, 1990.

[53] D. Carlson, “Debunking agile myths,” *CrossTalk*, vol. 30, no. 3, pp. 32–36, 2017.

[54] S. Jahn, C. Gold-Veerkamp, R. Reuter, J. Mottok, and J. Abke, “Secure software engineering in academic education: Students’ preconceptions of it security,” in *International Conference of Education, Research and Innovation (ICERI)*, IATED, Ed., 2019, pp. 6825–6834.

[55] P. Ralph and B. J. Oates, “The dangerous dogmas of software engineering,” 2018, [retrieved: 09, 2020]. [Online]. Available: arXiv: 1802.06321v1

[56] N. C. Shrikanth and T. Menzies, “Assessing developer beliefs: A reply to ”perceptions, expectations, and challenges in defect prediction”,” 2019, [retrieved: 09, 2020]. [Online]. Available: arXiv:1904.05794v1

[57] D. Rombach and F. Seelisch, “Formalism in software engineerings: Myths versus empirical facts,” in *Balancing Agility and Formalism in Software Engineering*, D. Hutchison, et al., Ed. Springer, 2008, pp. 13–25.

[58] R. Duit, “Schülervorstellungen – von Lerndefiziten zu neuen Unterrichtsansätzen [German],” in *Schülervorstellungen in der Physik*, R. Müller, R. Wodzinski, and M. Hopf, Eds. Köln: Aulis, 2011, pp. 8–14.

[59] C. Schulte and J. Magenheim, “Novices’ expectations and prior knowledge of software development,” in *First international Workshop on Computing education research*, R. Anderson, S. A. Fincher, and M. Guzdial, Eds. ACM Press, 2005, pp. 143–153.

[60] C. Gold-Veerkamp, “Big picture: für das evelin-doktorandenkoll.” 2018.

[61] B. G. Glaser, *Basics of Grounded Theory Analysis – Emergence vs. Forcing*, 1st ed. Mill Valley, CA: Sociology Press, 1992.

[62] J. M. Corbin and A. L. Strauss, *Basics of qualitative research: Techniques and procedures for developing grounded theory*, fourth edition ed. Los Angeles and London and New Delhi and Singapore and Washington DC and Boston: SAGE Publications Ltd, 2015.

[63] K. Charmaz, *Constructing Grounded Theory*, 2nd ed. London: SAGE Publications Ltd, 2014.

[64] K.-J. Stol, P. Ralph, and B. Fitzgerald, “Grounded theory in software engineering research: A critical review and guidelines,” in *The 38th International Conference on Software Engineering (ICSE)*, L. Dillon, W. Visser, and L. Williams, Eds. ACM Press, 2016, pp. 120–131.

[65] C. Gold-Veerkamp, J. Abke, and I. Diethelm, “What about misconceptions in software engineering? a research proposal,” in *Proceedings of the 2017 IEEE Global Engineering Education Conference (EDUCON)*, Institute of Electrical and Electronics Engineers, Ed. Piscataway, N.J.: IEEE, 2017, pp. 709–713.

[66] B. G. Glaser, *Theoretical sensitivity: Advances in the methodology of grounded theory*. Mill Valley, Calif.: Sociology Press, 1978.

[67] M. Selting, P. Auer, D. Barth-Weingarten, J. Bergmann, P. Bergmann, K. Birkner, E. Couper-Kuhlen, Arnulf Deppermann, P. Gilles, S. Günthner, M. Hartung, F. Kern, C. Mertzlufft, C. Meyer, M. Morek, F. Oberzaucher, J. Peters, U. Quasthoff, W. Schütte, Anja Stukenbrock, and S. Uhmann, “Gesprächsanalytisches Transkriptionssystem 2 (GAT 2),” *Gesprächsforschung - Online-Zeitschrift zur verbalen Interaktion*, no. Ausgabe 10, pp. 353–402, 2009.

[68] A. Boehm, “Grounded theory: Wie aus texten modelle und theorien gemacht werden,” in *Texte verstehen*, A. Boehm, A. Mengel, and T. Muhr, Eds. Konstanz: UVK Verlagsgesellschaft mbH, 1994, pp. 121–140.

[69] VERBI Software GmbH, “Maxqda: All-in-one qualitative & mixed methods data analysis tool,” 2020, (last access 2021-02-28). [Online]. Available: https://www.maxqda.com/

[70] M. M. Archibald, “Investigator triangulation: A collaborative strategy with potential for mixed methods research,” *Journal of Mixed Methods Research*, vol. 10, no. 3, pp. 228–250, 2016.

[71] B. Eisner, U. Kattmann, M. Kremer, J. Langlet, D. Plappert, and B. Ralle, “Gemeinsamer referenzrahmen für naturwissenschaften

(gerrn): Mindeststandards für die auf naturwissenschaften bezogene bildung. ein vorschlag."

[72] U. Kattmann, *Schüler besser verstehen: Alltagsvorstellungen im Biologieunterricht : [zusätzliche Stichwörter zum Download*. Hallbergmoos: Aulis Verlag, 2015.

[73] M. T. Rücker and N. Pinkwart, "Review and discussion of children's conceptions of computers," *Journal of Science Education and Technology*, vol. 25, no. 2, pp. 274–283, 2016.

[74] J. Erpenbeck, Ed., *Der Königsweg zur Kompetenz: Grundlagen qualitativ-quantitativer Kompetenzerfassung*, ser. Kompetenzmanagement in der Praxis. Münster and New York and München and Berlin: Waxmann, 2012, vol. Band 6.

[75] Y. Sedelmaier, "Interdisziplinäre Fachdidaktik für Software Engineering: Forschungsbasierte Entwicklung und Evaluation eines anwendungsbezogenen didaktischen Ansatzes," Ph.D. dissertation, Otto-Friedrich-Universität, Bamberg, 2015.

[76] H. Balzert, *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements-Engineering*, 3rd ed., ser. Lehrbücher der Informatik. Heidelberg: Spektrum Akad. Verl., 2009.

Validation of a Framework for Bias Identification and Mitigation in Algorithmic Systems

Thea Gasser, Rémy Bohler, Eduard Klein
Business Department
Bern University of Applied Sciences (BUAS)
Bern, Switzerland
email: thea.gasser@live.com
remy.bohler@sunrise.ch
eduard.klein@bfh.ch

Lasse Seppänen
Business Information Technology Department
Häme University of Applied Sciences (HAMK)
Hämeenlinna, Finland
email: lasse.seppanen@hamk.fi

***Abstract*—Bias in algorithmic systems is a major cause of unfair and discriminatory decisions in the use of such systems. Cognitive bias is very likely to be reflected in algorithmic systems as humankind aims to map Human Intelligence (HI) to Artificial Intelligence (AI). We conducted an extensive literature review on the identification and mitigation of bias, leading to precise measures for project teams building AI systems. Moreover, we developed an awareness-raising framework for use as a guideline for project teams, addressing AI responsibility, AI fairness, and AI safety. The framework proposes measures in the form of checklists to identify and mitigate bias in algorithmic systems considering all steps during system design, implementation, and application. We validated the framework successfully in the context of industrial AI projects.**

Keywords – Bias Framework; Artificial intelligence; Algorithmic system; Validation.

I. INTRODUCTION

This paper is a long version and an extension of the Bias Identification and Mitigation Framework presented in [1]. The original material contains the framework definition and application. Since validating the approach is crucial and contributes to the improvement and optimization of the framework, the validation process was carried out in an industrial context and is discussed in detail below.

Artificial intelligence is present in almost every area of our society, be it in medicine, finance, social media, education, human resource management, and many more. AI will become a greater part of people's lives since the Accenture Trend Report [2] states, that about 85% of the executives surveyed plan to invest widely in AI (artificial intelligence) technologies over the next three years. Moreover, AI will play a central role in how customers perceive a company and define to a large extent how interactions with their employees and customers take place. AI will become a core competency and will reflect a large part of a company's character. In five years, more than 50% of the customers will no longer choose a service based on the brand but will focus on how much AI is offered for that service [2].

Recently, however, there has been growing concern about unfair decisions made with the help of algorithmic systems that have led to discrimination against social groups or individuals [3] [4] [5]. As an example, Google's image search had been accused of bias indicating fewer women than the reality when searching for the term "CEO". Additionally, Google's advertising system displayed high-income jobs much less to women than to men [6]. The COMPAS algorithm has been accused of misclassifying black defendants as at risk of recidivism far more often than white defendants, while white defendants are misclassified as low risk far more often than black defendants [7].

Microsoft's Tay robot held racist and inflammatory conversations with Twitter users, which contained many political statements. It learned from the users' inputs and reflected it in its answers [8]. These and many other well-known examples show the demand for methods to measure algorithms, recognize and mitigate bias and provide fair AI software, especially in a high data-oriented machine learning context [4] [9].

This article contributes to AI safety by highlighting that bias in AI is very likely, illustrating possible sources of bias, and proposing a framework that supports the identification and mitigation of bias during the design, implementation, and application phases of AI systems.

The following research questions from Gasser [10] and Bohler [11] are addressed to tackle the above-mentioned aspects:

(1) What can be expected from AI systems compared to human decision-making?
(2) In what form is bias present in algorithmic systems?
(3) How can bias in algorithmic systems be identified?
(4) What measures can be taken to mitigate bias in algorithmic systems?
(5) How could bias be identified and referenced?
(6) To what extent does the framework application contribute to AI project improvement?

Sections III and IV discuss questions (1) and (2) based on literature research, and the proposed framework in Section V advises answering questions (3) and (4) in the context of machine learning based AI projects. Section VI discusses questions (5) and (6) during framework validation.

The rest of this paper is organized as follows. Section II describes the research design. Section III discusses various types of bias, followed by related research in Section IV. Section V addresses the bias mitigation framework in finer detail. Section VI discusses the framework validation in the context of industrial projects. The conclusions in Section VI close the article.

II. RESEARCH DESIGN

We conducted a literature search, mainly in SAGE Journals, ScienceDirect, Springer Link, Google Scholar, and the JSTOR Journal Storage. A range of search terms was used, such as "expectations towards AI", "human intelligence", "algorithmic bias", "bias in software development", “mitigating algorithmic bias”, resulting in the selection of about 125 sources. These were narrowed down to 75 relevant sources by cross-reading the abstracts and restricting them to articles from 2016 or later.

Based on the findings of the literature research, sources of bias and methods for identifying and mitigating bias in algorithmic systems were identified and structured and are systematically presented in Sections III and IV. The findings led to a framework for use in project settings, which is described in Section V, thereby identifying and mitigating bias using a metamodel, a set of checklists, and a one-pager template including the bias assessment criteria visualizing the assessment result.

We conducted validation through a Delphi method [13] with members from industrial AI projects as experts. After an initial phase of introducing the framework, we applied it to suitable projects. We collected feedback in the form of responses to the framework checklists, and subsequently summarized and structured it. The results were presented to the project members and refined in subsequent iterations until sufficient stability was achieved.

In addition to project-related aspects, we also raised meta-questions about applicability, usefulness, size, comprehensibility, and coverage concerning framework improvements.

In addition to a three-part document (project description, framework application, recommendations for bias mitigation), the validation results are visualized as a one-pager, making the bias assessment visible at a glance.

III. FROM HI TO AI

With AI, terms like imitation, simulation, or mimicking are repeatedly applied, which implies copying something, respectively, someone as, e.g., acting, learning, and reasoning like humans [14]. Therefore, if today’s AI behavior such as Apple’s Siri is considered, it could be claimed that the voice assistant is not intelligent. Looking into details, Apple’s voice assistant is based on evaluated data and facts permitting to offer an appropriate answer [15]. An independently thinking and reasoning machine is not yet present since, amongst other things, input is still needed. Even though AI acquires intelligence and learns through an autonomous process, it lacks sentience and self-awareness and is still only a simulation of HI (human intelligence) and nothing more [14].

Despite the expectations and efforts to map HI to AI, to date, no system can be classified as "strong AI", since this would include machines that act completely autonomously and have their intelligence and self-awareness like humans. However, "weak AI" systems working in a narrowly defined area are used successfully already [16]. Even in the case of self-learning machines, there is initial program code, a model, and learning rules so that machine learning can be effective [17]. Because human traits like self-awareness or empathy are missing in today's AI systems, there is still a gap between AI and HI. This, in turn, implies that partly intelligent systems are shaped by the influence of humankind and with it by cognitive bias, which is naturally present in humans and subsequently reflected through individuals and societies in algorithmic systems [18]. Research questions (1) and (2) relate to the decision-making aspects of AI systems.

A. *Lack of Transparency in AI Systems*

Algorithms are penetrating more and more into people’s lives and are likely to play an increasingly important role in their everyday lives, so they will depend to a large extent on how secure and efficient these algorithms are [19]. Algorithms are becoming increasingly complex, and systems may become opaque so that it becomes partly unclear even to the creators of such systems how exactly the interactions in the system(s) take place [20]. Measures must therefore be taken to minimize undesirable ethical consequences that could arise from the use of such systems. The main focus must be on the potential bias that might occur in the system design, implementation, and application phases.

B. *Bias and Fairness*

Since the term bias is defined as “the action of supporting or opposing a particular person or a thing in an unfair way, because of allowing personal opinions to influence your judgment” (Cambridge Dictionary) the topic of fairness plays a central role. A system might be viewed as fair in some circumstances and in other situations, it might be considered unfair. In addition, the presence of bias in an AI system cannot be regarded as evidence of the classification of a system as unfair, which means that neutral or even desirable biases may be present in AI systems without producing undesirable results [21]. Therefore, classifying an AI system as fair or unfair is subjective and may depend on the viewer, e.g., based on the application context's cultural setting.

Based on these factors, it is important to identify bias and consider whether there is a need for action for reducing it or whether bias should even be used specifically to prevent other bias in a different part of the system that would have more undesired consequences [21].

The question of whether recognized bias needs to be reduced at all should always be assessed in the individual system context since mitigating bias can be a major effort. On the one hand, several associations demonstrate differences in how and which values are put in the foreground and which seem less important. On the other hand, the situation can reach a level of complexity that no matter what perspective is adopted, some bias will always be identified from a certain point of view. In the end, technology cannot fully answer questions about social and individual values. It is therefore up to humans to make sure that the particular situation is always evaluated in a comprehensive context, meaning taking into account the whole ecosystem around the machine [21].

C. Sources of Bias

Various sources of bias in AI systems have been identified. Barfield & Bagallo [22] consider what we call *direct bias* whose sources are related to the core of AI systems:

- *Input bias* where the source data is biased due to the absence of specific information, non-representativeness, or reflecting historical biases.
- *Training bias* arises when the baseline data is categorized, or the output is assessed.
- *Programming bias* emerges in the design phase or when an algorithm modifies itself through a self-learning process.

In [23], sources are identified of what we call *indirect bias*, which is not located in the core of AI systems but in the surrounding ecosystem:

- *pre-existing bias*, which often emerges through social institutions, practices, and attitudes even before a system is designed.
- *Technical bias*, emerging from technical constraints, e.g., by favoring data (combinations) due to the order or size of screens and visual results presentation.
- *Emergent bias* arises when using a system outside its intended context of operation.

IV. Related Research

Recently, human aspects of AI have attracted a lot of attention. Not only private companies, research institutions, and nonprofit organizations, but also public sector organizations and governments have issued policies and guidelines on human aspects of AI. Many recent publications cite or build on the IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems called "Ethically Aligned Design" (EAD). This presents methodologies to guide ethical research to promote public debate on how these intelligent and autonomous technologies can be aligned with moral values and ethical principles that prioritize human well-being [24].

The non-profit research organization AlgorithmWatch is developing an "AI Ethics Guideline Global Inventory" [25] to address the question of how automated decision-making systems should be regulated. At the time of writing, more than 80 movements are listed, ranging from a few private companies (e.g., Google, Microsoft, IBM) to organizations (e.g., IEEE, ACM, Bitkom) and government-related organizations (e.g., China, European Commission, Canada, Singapore).

Several meta-studies presented the state of the art in human aspects of AI at the time of writing. In [26], an extended list is supplemented by a geographical distribution displayed on a world map. Global convergence of ethical aspects is revealed, emerging around five ethical principles: transparency, fairness, nonmaleficence, responsibility, and privacy. It highlights the importance of integrating efforts to develop guidelines and its implementation strategies.

In [27], a comprehensive literature review is presented based on key publications and proceedings complementing existing surveys of psychological, social, and legal discussions on the subject with recent advances in technical solutions for AI governance. Based on the literature research, a taxonomy is proposed that divides the field into areas, for each of which the most important techniques for the successful use of ethical AI systems are discussed.

[28] presents a framework for algorithmic hygiene and employs best practices to identify and mitigate them. A set of questions is compiled analyzing bias impact by the insight of 40 thought leaders who participated in a roundtable.

All publications mentioned present principles and guidelines for the consideration of ethical aspects in AI systems, thereby addressing research questions (1) and (2). However, they are general and generic and could be used as high-level recommendations only, which are not sufficiently specific for AI projects. The framework presented in Section V further develops these ideas and therefore points the way to the next step in incorporating ethical aspects in a project-oriented environment. Based on a metamodel and a set of checklists, it allows to identify and mitigate bias in AI systems in a project-oriented setting, thereby addressing research questions (3) and (4). The integration of ethical aspects into all project phases during the conception, development, and use of a system guarantees a high level of awareness among all project stakeholders.

V. The Bias Mitigation Framework

Awareness of the topic is the first step towards addressing bias in algorithmic systems. [29] states, that 92% of AI leaders make sure their technologists receive ethics training and 74% of the leaders assess AI outcomes every week. However, it is not enough to just dispose ethics codes that prevent harm. Therefore, establishing usage and technical guidelines and an appropriate mindset among the stakeholders are suggested.

To address bias in algorithmic systems appropriately, overarching and comprehensive governance must be in place in companies. Using the proposed framework, the project members should be committed to the framework, considering it as a binding standard.

In literature, many possibilities are described to identify bias such as (1) monitoring and auditing an AI system's creation process [30], (2) applying rapid prototyping, formative evaluation, and field testing [23], (3) manipulating test data purposefully to determine whether the results are an indication of existing bias in the system [31], (4) using the Socratic method promoting critical thinking and challenging assumptions through answering questions, where scrutiny and reformulation play a central role in the identification and reduction of bias [32].

The methods that aim to prevent bias in AI can be divided into technical approaches and awareness-raising approaches. Technical approaches attempt to integrate ethical principles into the design process of AI systems. Awareness-raising approaches aim to highlight the presence and the risks of bias through education and awareness initiatives so that members of AI projects could take care of bias problems and act responsibly in projects in this regard [33].

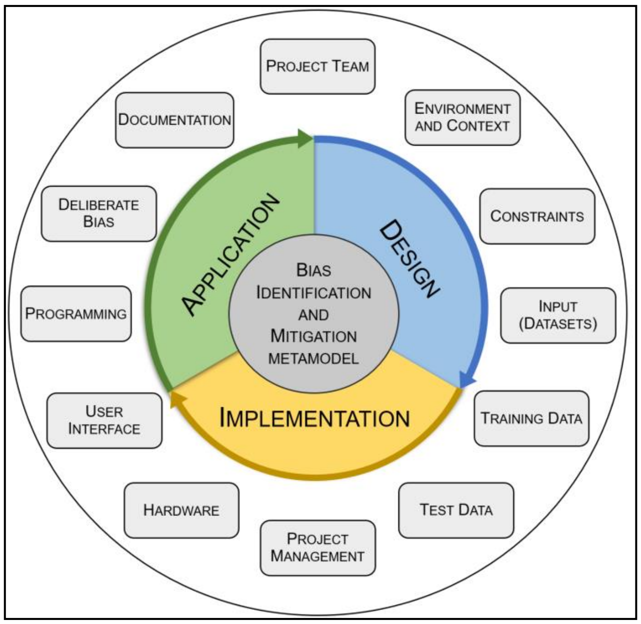


Figure 1. Metamodel for the Bias Mitigation Framework.

In contrast to the awareness-raising approach adopted in the presented framework, technical approaches such as IBM's "AI Fairness 360" offer metrics to check for unwanted bias in datasets and machine learning models [34]. Google's "What-If" tool enables visualization of inference results, e.g., for exploring the effect of a certain algorithmic feature and also testing algorithmic fairness constraints [35]. [36] tackles algorithmic bias, building models incorporating fair representation learning.

Many approaches have been suggested in the literature and tools are available focusing on specific topics in ethical aspects. Justification for the proposed framework is in incorporating aspects for all members involved in the process of creating an algorithmic system and all relevant aspects researched.

The framework consists of a metamodel (see Fig. 1), which is completed by checklists for areas covering the whole software life cycle around design, implementation, and application. The areas (e.g., Project Team, Environment, Context) are illustrated as rectangles in Fig. 1. The elements of each checklist consist of statements and questions that need to be addressed by the project team. The checklists are derived from the findings of the research described in Sections II, III, and IV and relate to the research questions (3) and (4).

Project Team

(a) All project members have had ethical training
- Members have a confirmation that they have completed courses or workshops or similar
- The minimum requirements to consider this element as fulfilled must be defined in the company

(b) All project members are aware of the topic of bias that exists in the human decision-making process
- Members took part in courses or workshops or similar
- The minimum requirements to consider this element as fulfilled must be defined on a project or company level

(c) All project members know about the fact that human bias can be reflected in an algorithmic system
- Members took part in courses or workshops or similar
- The minimum requirements to consider this element as fulfilled must be defined on a project or company level

(d) All project members consider the same attributes and factors as most relevant in the system context.
- A workshop is held where members share their views. Discrepancies are pointed out and a common understanding is developed. The workshops aim to share views, ideas and openly reveal conflicts and misunderstandings
- Due to cultural and background dissimilarities members might (unconsciously) weigh attributes differently

(e) The project team represents stakeholders of all possible end-user groups
- Stakeholder analysis comprehensively identifies end-user groups with a focus on identifying users who might be disadvantaged through the system outcomes
- Stakeholder analysis should be carried out with a change of perspective, where the worst scenario, i.e., if the system behaves discriminatory, identifies the groups that would be disadvantaged. (see area Project Management)

(f) The project team is a cross-functional team including diversity in ethnicity, gender, culture, education, age, and socioeconomic status
- The inputs of all the diverse individuals must be taken into consideration

(g) The project team has representatives from the public and private sector
-Exclusions need to be avoided

(h) Independent consultants are included for comparison with competing products
- Pre-existing bias in the context of the company's culture, attitude, and values can be revealed
- Independent consultants are needed because they are not biased by the companies' views

Figure 2. Checklist for the metamodel area *Project Team* with commented elements (a), (b), …

As an example, the area *Project Team* is subsequently described and detailed in Fig. 2. Knowledge, views, and attitudes of individual team members cannot be deleted or hidden, as these are usually unconscious factors, due to everyone's various backgrounds and experiences.

The resulting bias is likely to be transferred into the algorithmic system. Therefore, measures must be taken to ensure system neutrality as far as appropriate. There must be an exchange among project members where everyone shares their views and concerns openly, fully, and transparently before creating the system. Misunderstandings, ideas of conflict, too much euphoria, and unconscious assumptions or invisible aspects might get revealed this way. The checklist in Fig. 2 proposes the following concrete measures for addressing the above-mentioned issues: All project members (1) have had ethical training, (2) are aware of the bias topic that exists in the human decision-making process, (3) know about the fact that bias can be reflected in an algorithmic system, and (4) consider the same attributes and factors as most relevant in the system context.

Ideally, the project team (1) represents stakeholders of all possible end-user groups, (2) is a cross-functional team including diversity in ethnicity, gender, culture, education, age and socioeconomic status, (3) has representatives from the public as well as the private sector. Moreover, independent consultants are included for comparison with competing products.

A. *Checklists*

The metamodel in Fig. 1 illustrates 12 areas of interest, where the project team area was detailed already in Section V. This subsection provides an overview of the 11 remaining areas. For each area, the checklist is presented, and the corresponding literature references are explained.

Environment and Context
(a) All end-user groups are included in the testing phase
(b) End-user groups have been evaluated
- End-user groups' behavior is monitored and evaluated from various perspectives (surveys, interviews, recording behavior, letting them explain what they do and think while testing)
(c) Consequences and intentions have been considered
- For what and with what intentions was the system created?
- What is the worst thing that can happen in this algorithm if it starts interacting with others?
(d) Context is faithful to the source
- Does the current context represent the one, for which the system was created?

Constraints
(a) Business aspect reviewed
- Under what circumstances will the system be developed?
(b) Scope reviewed
- The requirements for the scope of the data set and the diversity are to be determined in the project in question
(c) Technical aspect reviewed
- Do technical constraints affect the way the system is designed?
(d) Legal aspect reviewed
- Do regulatory constraints affect the design of the system?

Figure 3. Checklist for areas *Environment and Context* and *Constraints.*

In [23], the various cultural values and attitudes of individuals are emphasized that could collide as they incorporate those into the project work. These aspects are covered by the areas *Environment and Context* and *Constraints* (Fig. 3) in the Framework. [17] [21] [31] [37] discuss the influence of direct bias (see "sources of bias" in Section III), leading to the areas in Fig. 4.

Input (Datasets)
(a) The data set is fully understood
- The meaning of each attribute is understood and its purpose in the system context is clear
(b) Data is transparent
- Data must be reliable, accurate, and kept up to date
(c) It is ensured that the data set represents the correct scope (enough data representing a population resp. a target group)
- Enough data and diversity are available
- The requirements for the scope of the data set and the diversity are to be determined in the project in question.
(d) The data source is known and verified
- Unknown data sources might lead to that the data being used in a context it was originally not intended to
(e) Data quality is ensured
- Low-quality data will cause even worse outputs since AI systems might reinforce errors in data sets
(f) It is clarified which attributes can legally be used
- Use of illegal attributes leads to a system becoming biased even though the attribute itself is not causing bias

Training Data
(a) The training data set is still as representative as the original data set
- Adjusting source data to training data can bear exclusion which needs to be prevented
(b) Added or omitted attributes are carefully chosen and justified
- One attribute can influence various areas in a system. Interconnectedness needs to be considered

Test Data
(a) Test data is independent
- The system uses test data it has never seen before
(b) Test data is defined
- Test scenarios are defined which are designed to detect bias that could be caused by a certain attribute
(c) Test data is reviewed
- Tests include omission and addition of attributes to test how system output changes

Figure 4. Checklists for the areas concerning *direct bias*, derived from "sources of bias" in Section III.

It is suggested that the complete algorithmic system lifecycle is accompanied and controlled through all phases with a project management approach. The classical element "risk analysis" must be expanded with a focus on risk factors that could favor bias and the effects recognized bias could have. Isele [32] suggests that critical questions should be asked, critical thinking adopted, assumptions challenged, and the systems' results evaluated. Project Management area aspects are gathered in Fig. 5.

Project Management
(a) The project management process includes methods that focus on bias issues
- Stakeholder analysis is adjusted for disadvantaged group identification in the worst case
(b) Risks concerning bias are assessed and known to each team member
- Risk analysis is adjusted for additional focus on bias and worst-case scenarios provoking bias
(c) Critical thinking is promoted and demanded at every stage of the system creation process
- How would changes to a data point affect the model's prediction?
- Does it perform differently for various groups? For example, historically marginalized people?
- How diverse is the dataset I am testing my model on?
- Is the system context the one the system was intended to?
- Can the outcome/result/system recommendation be justified?
- How diverse is the dataset I am testing my model on?
- Does it perform differently for various groups–for example, historically marginalized people?
- How would changes to a data point affect my model's prediction?
(d) Perspectives are changed continuously to challenge assumptions
- Various points of view ensure the identification of hidden assumptions
(e) Monitoring measures are defined, communicated, and applied
- End-user groups' behavior is monitored and evaluated from various perspectives (surveys, interviews, recording behavior, letting them explain what they do and think while testing)
(f) Auditing measures are defined, communicated, and applied
(g) Workshops/meetings are set frequently which address upcoming doubts of team members
- Critical thinking is continuously fostered in workshops and outside
(h) Scenario thinking is fostered
(i) Freedom of expression is guaranteed and desired
- Every input of any team member can reveal hidden bias

Figure 5. Checklist for the area *Project Management*

Hardware limitations, such as screen size or performance bottlenecks, could influence system output [23]. The design of visual representations of objects could also be a source of bias, requiring a careful design of the graphical user interface [38]. Checklists for hardware limitations and Graphical User Interface (GUI) design are detailed in Fig. 6.

The presence of deliberate bias might be surprising at first, however, is applied in some cases to prevent bias from arising in another, more important area of a system. As an example, a statistically biased estimator in an algorithm might exhibit significantly reduced variance on small sample sizes, thereby greatly increasing reliability and robustness in future use [21].

Sources of bias in programming and documentation and discussion on deliberate bias [21] are given in Fig. 7.

Hardware
(a) Hardware limitations
- Do hardware limitations exist?
(b) Influence on the creation process
- Do these limitations influence the system creation process?
(c) Influence on production environment
- Do these limitations influence the system's functionality in the production environment?

User Interface
(a) Visual aspects are determined appropriately
- The font style, font size, font color, and placement of text are justified and reflect the intention of the system's functionality
- Color, size, and placement of forms and graphics are justified and reflect the intention of the system's functionality
(b) Visual result
- Does visual result representation (alphabetically or random) make any difference (user always chooses the results displayed first?)
(c) Navigation
- Does a change in navigation representation lead the user to favor different results?
(d) Graphical User Interface
- Is graphical UI limiting/favoring data over other data?
(e) Language Aspects
- How do the chosen language influence the user's perception and interpretation in various contexts and circumstances?
- Is a translation of data/information necessary?
- Do the information and results become distorted through the application of translation?
- How is the translation interpreted by the end-users?
(f) Alternative GUI
- The system features are changed, and end-users are monitored once more to see how their behavior changes
- Several features may need to be changed various times to reveal hidden assumptions of end-users

Figure 6. Checklist for areas *Hardware* and *UI*.

The checklists in table form for use in a bias assessment and supplementary material are available at https://instructor.github.io/bias/.

B. *Framework Use*

Based on the outcome of the above-mentioned literature research, the approach presented is intended to be an initial framework that can be adapted to specific needs within a given project context. It comes in the shape of a guideline complemented with checklists, e.g., for the members of a project team.

The adjustments could be made based on an adapted understanding of system neutrality, which may be specific for the application or application domain in question. If the proposed framework is used mandatory within a project, it is very likely that the developed application reflects the neutrality defined by the project team or company.

Programming
(a) Code reviews take place
- Measures aim to understand adapted or reused code fully
(b) Independent code audits are conducted
- Independent audits foster considering the code from various points of view and reveal unconscious assumptions
(c) Possible user behavior is analyzed beforehand to keep a learning system from adopting discriminatory behavior
- Thinking outside the box is fostered especially considering word and language usage in the system context
- The system can handle discriminatory user behavior

Deliberate Bias
(a) Bias is identified and categorized
- Are the identified biases considered as good, neutral, or bad ones?
- Is there any bias that was implemented on purpose to mitigate others?
(b) It is ensured that all the identified biases are monitored during the whole system creation process
- Bias needs to be tracked and changes identified as well as recorded throughout every project stage

Documentation
(a) Availability of relevant information
- Traceability, justification and business continuity is ensured
(b) Comprehensible documentation
- The language may only contain such a high degree of complexity and technical language that every project member understands it
- Prevention of misunderstandings is ensured
(c) Documentation has been reviewed and approved
- The documentation needs to be reviewed by several project members and stakeholders

Figure 7. Checklist for areas *Programming, Documentation, and Deliberate Bias.*

Verifying that the framework has been applied and the requirements have been met will help to determine the extent to which the system is neutral and the need for appropriate action.

VI. FRAMEWORK VALIDATION

To reveal the advantages and disadvantages of the proposed framework, we conducted its application and evaluation in practical AI projects. Recommendations for the proper use and improvement of the approach are derived from the results.

First, we looked at a chatbot project of a Swiss insurance company to automate customer communication. Technically, it is based on NLP (natural language processing). The chatbot helps customers orientate and navigate their website to find the proper information based on customer queries. The learning mechanism enables the chatbot to improve customer help constantly. At the time of writing even non-specific customer input is being properly processed in many cases [39].

The second study object was the Smart Animal Health project from a Swiss government agency collecting data on farm animals to evaluate the effectiveness of measures in the farm animal sector. The system is designed to improve the well-being of animals, establishing an early-warning system for on-site problems. Another goal is the identification of trends.

Source data are extracted from governmental open data and private data sources deducing key indicators such as disease symptoms, animal behavior, and husbandry conditions. By linking the data, a more complete picture of livestock farms emerges. Statements on animal health can be derived for farms or the farm animal population under consideration as a whole. Technically, the project uses machine learning algorithms, specifically supervised learning [40].

The Delphi-based project member involvement resulted in a textual summary for each of the 12 framework areas of interest (detailed in [11]) and a visual presentation in the form of a one-pager (see Fig. 8) reflecting each area of interest through a rectangular shape. The areas of interest are placed in three columns, arranged from left to right and top to bottom. Each rectangular area contains the elements from the corresponding checklist.

Based on the project members' feedbacks, the elements are highlighted in green (darker background in black and white print) if the element has successfully been addressed, in yellow (lighter background in black and white print) if the element has partially been addressed, or in white resp. not highlighted if it is not applicable in the project context in question. The distinction between successful and partially successful is open to interpretation. We determined the final classifications during the iterations in the Delphi process.

The one-pager template, the results of the two projects studied, and supplementary material can be found at https://instructor.github.io/bias/.

A. *Framework Application*

In both projects, a sequence of activities has been carried out:

a) explanation of the framework
b) application to the project
c) evaluation of the results
d) bias identification
e) derivation of recommendations for mitigating bias.

Steps b) to d) were repeated until sufficient stability was achieved.

In the one-pager for the chatbot project (see Fig. 8) the yellow regions show potential for improvement concerning bias aspects in the project, while the green regions satisfy the bias criteria in question to a sufficient degree. Since all elements in the one-pager are highlighted in green or yellow, they could be well reflected and were considered relevant by the project members involved.

Elements from the following areas of interest were identified in the yellow regions: project team, project management, programming, and deliberate bias. After analyzing the comments on the yellow elements, the following recommendations were derived to mitigate bias:

- Interaction bias [41] can arise as the learning chatbot adopts biased customer statements into its interaction pattern. It is recommended to monitor the interactions and proactively adjust them if necessary.
- In learning systems, bias can change dynamically based on the machine learning process. Deliberate bias should be monitored carefully to prevent its spillover to unwanted bias.
- To raise awareness and establish know-how, it is recommended to conduct training and workshops on the topic of bias.
- It is recommended that the bias assessment process be repeated at appropriate intervals to achieve ethical sustainability in AI projects.

Based on the project partners' answers to the framework application, we used color-coding to mark not only the areas of compliance with the framework but also the areas of deviation. Subsequently, we further broke down the areas marked in yellow by documenting in a project-specific way how the potential for bias in these areas could have a negative impact on the project.

Finally, we discussed the color-coding and documentation with the project partners and made recommendations to address the yellow highlighted areas, thereby answering the research questions from Section I. Through the documentation and recommendations, we were able to both raise awareness of bias in the respective projects and, with the help of the framework, provide a guide for dealing with and avoiding potential bias.

The one-pager for the smart animal health project revealed similar bias issues (and recommendations) with an additional yellow region for the input data sets (which are still too few). Some elements are not highlighted in green or yellow because not all areas and elements could be meaningfully applied due to the early project stage. Bohler [11] documents both one-pagers for the above-mentioned projects in detail.

B. Framework Validation

After framework application, the projects replied to a set of meta-questions ([11], p.37) about applicability (strength, weaknesses), usefulness (relevance, increased awareness for bias), size, comprehensibility, and coverage:

- *Applicability:* The framework is perceived as applicable to the project in question. Specific strengths are the sensitization to the topic of bias and the holistic perspective on AI projects nudging for reflection on the framework areas of interest and their set of elements. Applicability increases significantly when training in bias or ethical topics had taken place beforehand.
- *Usefulness:* The framework addresses a relevant problem in the context of AI projects, especially when decision-making algorithms are involved. It raises awareness of ethical issues in the design of AI systems. Bias mitigation can only be achieved after referencing identified bias in the project context. Moreover, the possible weighting of elements and the use of metrics could enable more targeted referencing and recommendations for improvement in many project settings.
- *Size:* The framework is considered extensive with its mixture of organizational, social, and technical aspects. Its application in an AI project is therefore complex and may require several people with appropriate expertise. Especially for smaller projects, a holistic application is challenging, since e.g., it cannot be assumed to establish a cross-functional and diverse project team. Its deployment can be reflected at most imaginary instead.
- *Coverage:* The scope was considered appropriate, as no elements were felt to be superfluous or missing.
- *Comprehensibility:* The areas and checklist elements are largely found to be understandable. Examples or sample comments could further increase comprehensibility and give orientation filling in the one-pager.

C. Reflection on Framework Application

During framework application and based on the feedback on the above meta-questions, it showed to be helpful to follow specific restrictions and to establish several preconditions:

- Before application, awareness of the bias problem in general and for a specific project must be created. The risks of bias potentials should be illustrated. Otherwise, there is a risk that project participants will underestimate the importance of the framework criteria or not consider them important.
- There should be an introduction to the framework in terms of goals, content, and procedures. In particular, it should be conveyed that the framework is to be understood as a guideline rather than a prescription. The strengths of the framework should be emphasized, in part because it is based on a sound literature review. It is beneficial to answer the questions as given.
- The framework can be suitably adapted for the application (e.g., textual formulations, specification of examples).
- The application should be accompanied and supported by trained persons. If this person is outside the project, a regular exchange between the trained persons and the project experts should be established.
- To establish a sustainable awareness effect, the bias assessment should be repeated after a reasonable period.
- The results of a bias assessment should be documented in the form of the one-pager, the summarized expert feedback (e.g., per area of interest), and the set of project-specific recommendations for bias mitigation. By highlighting the risks, awareness of the relevance of measures for mitigating bias is promoted. [11]

contains sample documentation for both above-mentioned projects.

- Identified bias potentials do not necessarily confirm the existence of bias. Nevertheless, bias potentials should be considered in-depth and appropriate measures taken. This prevents bias potentials from developing into bias.
- After completion of a bias assessment, users are to be surveyed regarding their application experience. The resulting insights can be used to improve the framework and to increase its practical validity.

D. *Framework Improvements*

Although the framework was successfully applied to practical AI projects, we could identify several aspects for framework improvement and its future application:

- To focus on particularly relevant areas of interest or checklist elements, a project-specific weighting scheme could be helpful, by which the checklists are instrumented in advance.
- Because the framework contains text-based checklists, there is always room for interpretation. Although the iterative bias assessment process can settle the scope for interpretation in certain cases, the introduction of metrics and the use of a glossary could reduce diffusivity.
- The presence of sample answers or examples for checklist criteria would, on the one hand, speed-up orientation. On the other hand, it could act as a time-saving cheat sheet that prevents users from creatively developing their ideas.
- To add more stability to the framework, more bias assessments should be performed in AI projects from various application domains. This could also identify, for example, areas where bias potential is particularly common (or rare) and thereby give valuable hints for future projects. Moreover, valuable indications of the general situation around bias in AI could be derived.
- From time to time, the framework should be reflected based on user feedback regarding the meta-questions and formulations within the checklists.
- With the rapid development of the AI field, new framework conditions and facts may arise about bias in AI. The framework should therefore be reviewed to ensure that it is up to date and adapted if necessary.

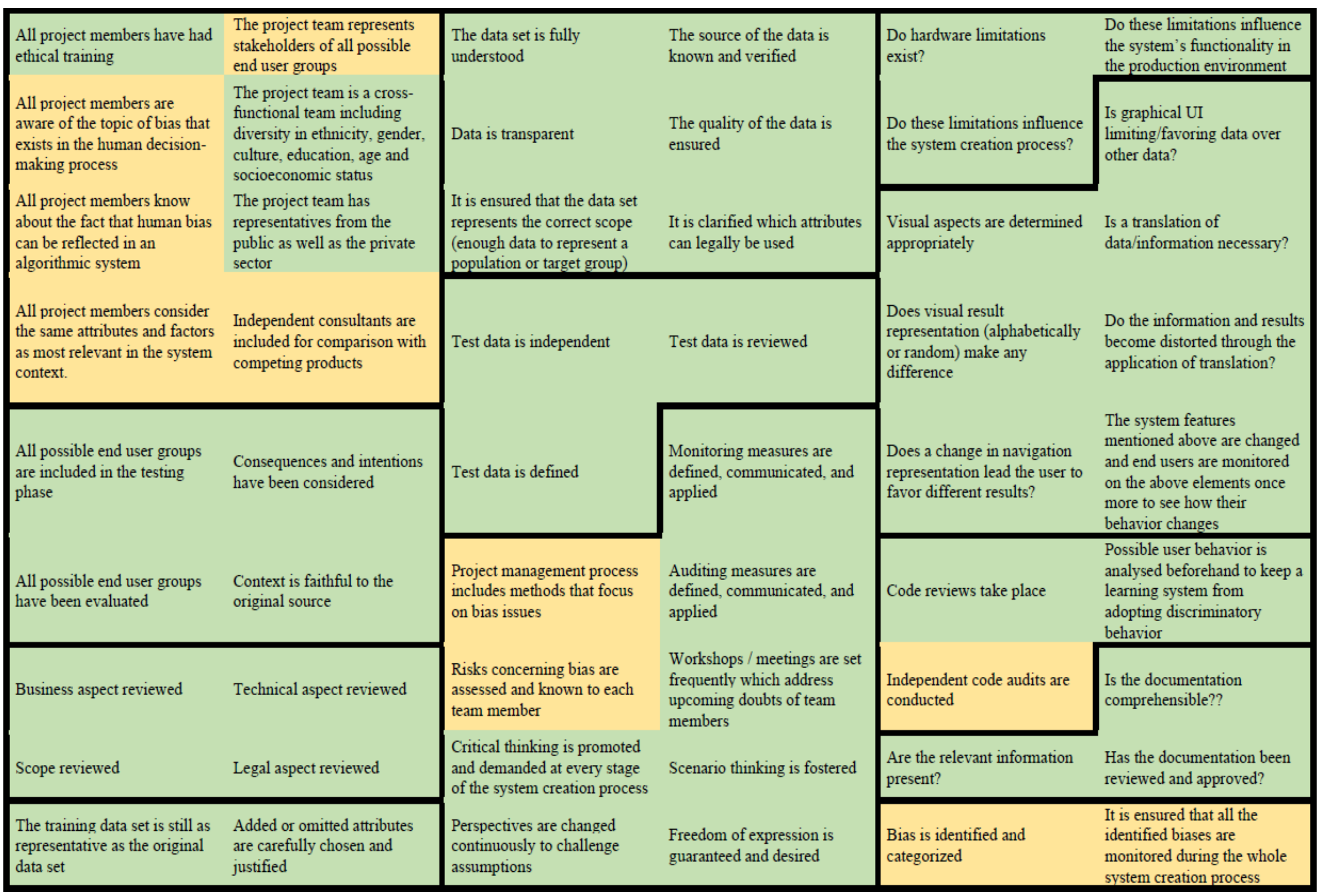


Figure 8. One-pager for the chatbot project presenting 12 areas of interest (rectangular shapes) containing its corresponding checklist elements. Green (resp. yellow) elements have successfully (resp. partially) been addressed. In black and white print, green means darker background and yellow means lighter background.

VII. Conclusion

Since currently there are only weak AI systems that lack self-awareness and depend on human advice in the shape of created models and selected training data, human bias is naturally and unintentionally reflected in crafted algorithmic systems. A framework has been proposed and validated, which helps to identify and mitigate bias in algorithmic systems, covering aspects of the complete life cycle of such software systems.

The framework was developed based on desk research, and validation was conducted through field research. The approach was implemented in realistic software project situations and its added value could be observed, evaluated, and validated. During validation, each area of interest of the metamodel was evaluated against the criteria and questions in the checklists, and user feedback was summarized and structured along with the areas of interest. Subsequent reflection on the bias assessments led to slight framework improvements.

As a future improvement, it would be useful to investigate to what extent automation of the framework use could mitigate subjective opinions and views of the stakeholders involved. As an example, the following scenario could be realized: Information about the adapted framework (metamodel areas and checklist elements), which is considered standard for ensuring system neutrality up to a certain point in the project in question, could be supported by a software system. During the project, the checklists are continuously filled with data by the project team, thereby enabling process analysis, comparison of different framework implementations, and revealing indications if and where the recommendations were complied with.

On the one hand, a specific project team would always be aware when creating an algorithmic system, which of the specified areas would not be adhered to and could exhibit potential bias. On the other hand, this mechanism could also be used for end-users. They could more easily assess the reliability of the results of an AI system are, and which areas need more attention regarding bias. The impact of decisions taken through the AI system's suggestions can be better analyzed by knowing which areas do not comply with the elaborated standard.

However, to reach this point, several aspects need to be considered. Elements from the checklists would have to be detailed at the micro level to define, for example, what a stakeholder is or how it can be verified that the test user belongs to a specific gender. Instead of a yes/no checkmark in the checklists, there could be more detailed measures, e.g., an indication of the level to which a team member has received ethical training. Furthermore, the integration of mechanisms that consider the plausibility of the answers in the checklists would be helpful.

References

[1] T. Gasser, E. Klein, and L. Seppänen, "Bias – A Lurking Danger that Can Convert Algorithmic Systems into Discriminatory Entities," in *Centric2020 - The 13th Int. Conf. on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services*, 2020, pp. 1–7.

[2] Accenture AG, "AI is the new UI – Experience Above All," *Accenture Technology Vision*, 2017. [Online]. Available: https://www.accenture.com/_acnmedia/Accenture/next-gen-4/tech-vision-2017/pdf/Accenture-TV17-Full.pdf. [Accessed: 24-Nov-2021].

[3] A. Koene, "Algorithmic Bias: Addressing Growing Concerns," *IEEE Technol. Soc. Mag.*, vol. 36, no. 2, pp. 31–32, Jun. 2017.

[4] M. Veale and R. Binns, "Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data," *Big Data Soc.*, vol. 4, no. 2, pp. 1–17, Dec. 2017.

[5] S. Feuerriegel, M. Dolata, and G. Schwabe, "Fair AI," *Bus. Inf. Syst. Eng.*, vol. 62, no. 4, pp. 379–384, Aug. 2020.

[6] D. Cossins, "Discriminating algorithms: 5 times AI showed prejudice," *New Scientist*, 2018. [Online]. Available: https://www.newscientist.com/article/2166207-discriminating-algorithms-5-times-ai-showed-prejudice/. [Accessed: 24-Nov-2021].

[7] J. Larson, S. Mattu, L. Kirchner, and J. Angwin, "How We Analyzed the COMPAS Recidivism Algorithm," *ProPublica*, 2016. [Online]. Available: https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm. [Accessed: 24-Nov-2021].

[8] E. Hunt, "Tay, Microsoft's AI chatbot, gets a crash course in racism from Twitter," *The Guardian*, 24-Mar-2016. [Online]. Available: https://www.theguardian.com/technology/2016/mar/24/tay-microsofts-ai-chatbot-gets-a-crash-course-in-racism-from-twitter. [Accessed: 24-Nov-2021].

[9] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 1–9.

[10] T. Gasser, "Bias – A lurking danger that can convert algorithmic systems into discriminatory entities," 2019. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/167429/Gasser_Thea.pdf. [Accessed: 24-Nov-2021].

[11] R. Bohler, "Validation of the Framework for Bias Identification and Mitigation," BUAS - Bern University of Applied Sciences, Master Thesis (in German), 2021.

[12] H. Snyder, "Literature review as a research methodology: An overview and guidelines," *J. Bus. Res.*, vol. 104, pp. 333–339, Nov. 2019.

[13] L. Harold, M. Turoff, and O. Helmer, *The Delphi Method - Techniques and Applications*. Addison-Wesley, 1975.

[14] S. Holder, "What is AI, really? And what does it

mean to my business?," 2018. [Online]. Available: https://www.sas.com/en_ca/insights/articles/analytics/local/what-is-artificial-intelligence-business.html. [Accessed: 24-Nov-2021].

[15] A. Goel, "How Does Siri Work? The Science Behind Siri," *Magoosh Data Science Blog*, 2018. [Online]. Available: https://magoosh.com/data-science/siri-work-science-behind-siri/. [Accessed: 24-Nov-2021].

[16] J. R. Searle, "Minds, brains, and programs," *Behav. Brain Sci.*, vol. 3, no. 3, pp. 417–457, 1980.

[17] E. Alpaydın, *Introduction to Machine Learning*, 4th ed. MIT Press, 2020.

[18] J. Burrell, "How the machine 'thinks': Understanding opacity in machine learning algorithms," *Big Data Soc.*, vol. 3, no. 1, pp. 1–12, 2016.

[19] A. Smith, "Franken-algorithms: the deadly consequences of unpredictable code," *The Guardian*, 2018. [Online]. Available: https://www.theguardian.com/technology/2018/aug/29/coding-algorithms-frankenalgos-program-danger. [Accessed: 24-Nov-2021].

[20] C. Smith, B. McGuire, T. Huang, and G. Yang, "The History of Artificial Intelligence," Washington, 2006.

[21] D. Danks and A. J. London, "Algorithmic Bias in Autonomous Systems," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 4691–4697.

[22] W. Barfield and U. Pagallo, "Research Handbook on the Law of Artificial Intelligence. Edited by Woodrow Barfield and Ugo Pagallo. Cheltenham, UK," *Int. J. Leg. Inf.*, vol. 47, no. 02, pp. 122–123, Sep. 2019.

[23] B. Friedman and H. Nissenbaum, "Bias in computer systems," *ACM Trans. Inf. Syst.*, vol. 14, no. 3, pp. 330–347, Jul. 1996.

[24] IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems, "Ethically Aligned Design," 2019.

[25] AlgorithmWatch, "AI Ethics Guidelines Global Inventory," 2019. [Online]. Available: https://algorithmwatch.org/en/project/ai-ethics-guidelines-global-inventory/. [Accessed: 24-Nov-2021].

[26] A. Jobin, M. Ienca, and E. Vayena, "The global landscape of AI ethics guidelines," *Nat. Mach. Intell.*, vol. 1, no. 9, pp. 389–399, Sep. 2019.

[27] H. Yu, Z. Shen, C. Miao, C. Leung, V. R. Lesser, and Q. Yang, "Building Ethics into Artificial Intelligence," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018, pp. 5527–5533.

[28] N. Turner Lee, P. Resnick, and G. Barton, "Algorithmic bias detection and mitigation: Best practices and policies to reduce consumer harms," 22-May-2019. [Online]. Available: https://www.brookings.edu/research/algorithmic-bias-detection-and-mitigation-best-practices-and-policies-to-reduce-consumer-harms/. [Accessed: 24-Nov-2021].

[29] SAS, "Organizations Are Gearing Up for More Ethical and Responsible Use of Artificial Intelligence, Finds Study," 2018. [Online]. Available: https://www.sas.com/en_id/news/press-releases/2018/september/artificial-intelligence-survey-ax-san-diego.html. [Accessed: 24-Nov-2021].

[30] A. Raymond, "The Dilemma of Private Justice Systems: Big Data Sources, the Cloud and Predictive Analytics," 2014.

[31] I. Žliobaitė, "Measuring discrimination in algorithmic decision making," *Data Min. Knowl. Discov.*, vol. 31, no. 4, pp. 1060–1089, 2017.

[32] E. Isele, "The Human Factor Is Essential to Eliminating Bias in Artificial Intelligence," *Chatham House*, 2018. [Online]. Available: https://www.chathamhouse.org/expert/comment/human-factor-essential-eliminating-bias-artificial-intelligence. [Accessed: 24-Nov-2021].

[33] S. Beck, A. Grunwald, K. Jacob, and T. Matzner, "Künstliche Intelligenz und Diskriminierung," 2019. [Online]. Available: https://www.plattform-lernende-systeme.de/publikationen-details/kuenstliche-intelligenz-und-diskriminierung-herausforderungen-und-loesungsansaetze.html. [Accessed: 24-Nov-2021].

[34] K. R. Varshney, "Introducing AI Fairness 360, A Step Towards Trusted AI," *IBM Research Blog*, 2018. [Online]. Available: https://www.ibm.com/blogs/research/2018/09/ai-fairness-360/. [Accessed: 24-Nov-2021].

[35] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viegas, and J. Wilson, "The What-If Tool: Interactive Probing of Machine Learning Models," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 1, pp. 56–65, 2020.

[36] E. Adeli *et al.*, "Representation Learning with Statistical Independence to Mitigate Bias," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 2512–2522.

[37] S. Barocas and A. D. Selbst, "Big Data's Disparate Impact," *SSRN Electron. J.*, vol. 104, pp. 671–732, 2016.

[38] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River: Prentice Hall, 2010.

[39] Sozialversicherungsanstalt des Kantons St. Gallen, "Der IPV-Chatbot," 2019. [Online]. Available: https://www.svasg.ch/news/meldungen/ipv-chatbot.php. [Accessed: 24-Nov-2021].

[40] Swiss Federal Food Safety and Veterinary Office, "Research project «Smart Animal Health»," 2020. [Online]. Available: https://www.blv.admin.ch/blv/en/home/tiere/forschungsprojekte-tiere/forschungsprojekt-smart-animal-health.html. [Accessed: 24-Nov-2021].

[41] K. LLoyd, "Bias Amplification in Artificial Intelligence Systems," in *AAAI FSS-18: Artificial Intelligence in Government and Public Sector*, 2018.

Natural Language Processing Techniques for Enhancing Formation Evaluation

Klemens Katterbauer
EXPEC ARC
Saudi Aramco
Dhahran, Saudi Arabia
klemens.katterbauer@aramco.com

Rabeah Al Zaidy
Faculty of Computer Science
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia
rabeah.alzaidy@kfupm.edu.sa

Sara Abu Al Saud
EXPEC ARC
Saudi Aramco
Dhahran, Saudi Arabia
Sara.abualsaud@aramco.com

Alberto Marsala
EXPEC ARC
Saudi Aramco
Dhahran, Saudi Arabia
Alberto.marsala@aramco.com

***Abstract*— Formation evaluation literature and reports in the oil and gas industry are crucial in decision making and understanding of optimizing recovery. The literature provides a comprehensive summary of tools and interpretations, as well as use cases for individuals to learn and utilize the information for enhancing their formation evaluation interpretations and decision-making. A major challenge in practice is the abundance and heterogeneity of information available that leads to individuals facing enormous obstacles to retrieving the right information within an adequate timeframe. We present an overview of several approaches in natural language processing for creating an ontology framework of formation evaluation data and literature, as well as conversational AI tools to extract information for the users. The review outlines the challenges that are faced when categorizing data related to formation evaluation, as well as establishing correlations and connections between various information sources. Finally, the review will provide a summary of different conversational AI approaches and systems for assisting well log and formation evaluation interpretation, as well as the opportunities and challenges faced. In conclusion, we will dedicate the way forward for NLP-driven approaches for assisting formation evaluation interpretation in real-time, and the business impact it has in the oil and gas industry and relationship to other initiatives both in the oil and gas industry as well as beyond.**

Keywords – reservoir formation evaluation; natural language processing; artificial intelligence; Petroleum industry

I. Introduction

Natural language processing (NLP) has become a cornerstone in several fields, including the oil and gas industry, and has become a cornerstone technology with crucial potential in the area of the 4th industrial revolution technology. NLP began in the 1950s as the intersection of artificial intelligence and linguistics. NLP was originally distinct from text information retrieval (IR), which employs highly scalable statistics-based techniques to index and search large volumes of text efficiently [1].

The statistical techniques utilized for IR encompass a wide range of frequency and distribution statistical methods. With time, however, NLP and IR have converged somewhat. Currently, NLP borrows from several very diverse fields, requiring today's NLP researchers and developers to broaden their mental knowledge base significantly. While statistical techniques represent a major area of NLP, advanced neural networks have become an important element to expand the utilization of NLP to learn in multiple settings by machines themselves. Simple statistical approaches face the challenge that it requires humans to provide and specify the dedicated responses for each human response. The response may differ depending on the context and in the light of the overall conversation, which made it almost impossible to compete with a human interpreter.

Word-for-word Russian-to-English machine translations, due to their primitive nature, were in the early days easily defeated by homographs[1] and metaphor. For example, the statement "the spirit is will, but the flesh is weak," was translated into "vodka can be agreed on, but it spoiled the meat," which easily showed the limitations and potential wrong conclusions that may be derived from word-for-word translations [2]. While the test failed tremendously, it provided a breakthrough for the computing industry, which showed that a computer is able to provide machine translations.

The first theoretical analysis of the complexity of language grammar was carried out by Chomsky [3]. This significantly influenced the creation of the Backus-Naur Form (BNF) notation, which is still widely utilized [4]. The focus of BNF is to define context-free grammar in a similar form as a programming language syntax. The main objective is to translate context-free grammar into a form that is

[1] Identically spelled words with multiple meanings

understandable for computer scientists and can be easily implemented on a computer.

When analyzing a language, the BNF specification consists of a number of derivation rules that syntactically validate the program code. A crucial understanding in this context is that the rules do not represent expert systems heuristics but solely constraints.

Another crucial part was the development of text-search patterns, based on which the concept of regular expression syntax was developed [5].

These developments led in the 1970s to heavily exploit lexical-analyzer (lexer) generators and parsers that incorporated grammars. A lexer is a transformer that transforms a text into tokens, where the subsequent parser validates the sequence of the tokens. The combination of lexers and parsers provides a solid foundation for the implementation in a programming language as it takes regular expressions and the BNF specifications and transforms it into code and lookup tables to determine decisions related to lexing and parsing [6].

Although context-free grammar (CFG) may not theoretically be adequate for natural language processing, its ability to transform easily into programming language syntax makes them very attractive in practice [7]. This has to do with the fact that there is a deliberate attempt to have a restrictive CFG variant in order to improve the implementation. Such a form of grammar is called a look-ahead parser with left-to-right processing and rightmost (bottom-up) derivation (LALR) [8]. The operating procedure of LALR is that the text is scanned first of all from left to right and then performs a bottom-up approach, where the compounds are constructed gradually from simpler ones. The look-ahead implies that the parsing decisions are made based on taking into account a single token ahead of the existing token. Given that there is only a single token that is taken into account when determining a parsing decision, this may represent a challenge to adequately infer the meaning of a sentence structure [9].

The 1970s also led to the development of the Prolog language, whose syntax is focused on writing grammars [10]. In order to achieve the simplest implementation mode (top-down parsing), the rules have to be changed to right-recursively. The challenge with a top-down approach is that they are considerably slower than bottom-up parsers, as they do not need generators.

II. Statistical NLP: Overcoming the Challenges of Specified, Explicit Rules

The difference between various natural languages differs tremendously, which exacerbates the challenge of determining the intent and meaning of sentences and statements in a specific language. The huge size, as well as unrestricted nature of natural languages, present significant problems that are further exacerbated by the ambiguity of language [11]. Hence, standard parsing approaches based on symbolic and manual rules are set to face two major critical challenges (Figure 1).

- The first challenge is that NLP has to extract the meanings of the text, which are the semantics. These are the formal grammars that outline the relationship between the text units. [2] that address primarily syntax. Extension of grammars by expanding sub-categorization and incorporating additional constraints and rules can help understanding better the natural language semantics; however, the increasing number of rules can lead to an unmanageable set that may unpredictably interact with each other and can lead to multiple interpretations of the word sequence. The arising ambiguity represents a major challenge, as the user is interested in the context and avoids ambiguity in interpretation.
- Another challenge is that handwritten rules face significant challenges with ungrammatical spoken sentences, even though the sentence is comprehensible by humans.

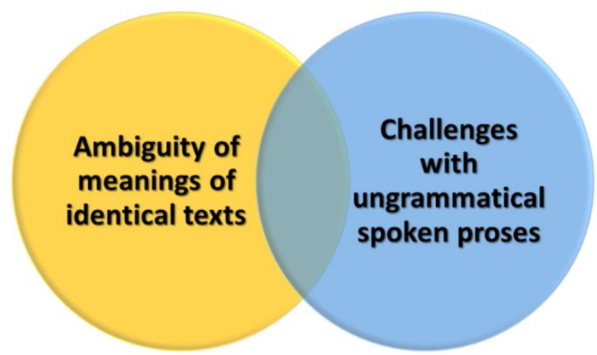


Figure 1: Major challenges of handwritten rules-based NLP.

These two challenges led to a significant rethinking of how to approach the processing of natural language via focusing on simple and robust approximations of the natural language instead of deep analysis (Figure 2). Additionally, evaluation became considerably more rigorous as compared to before, and the utilization of machine learning techniques.

The move from deterministic to probabilistic language models was a decisive factor given the inherent ambiguity of language and also the probabilistic determination of the meaning of sentences by humans themselves. Almost anyone has experienced that the meaning of a sentence or prose may very much differ in the context or how it is spoken. The same form and way how a prose is stated may even differ in terms of its interpretation between different cultures [12].

[2] These are parts of speech, such as nouns, verbs, and adjectives

Additionally, larger documented text statements were utilized for training these new machine learning algorithms, which provided a ground truth for the evaluation, and hence better determination of how to correctly interpret the text fragments and sentences.

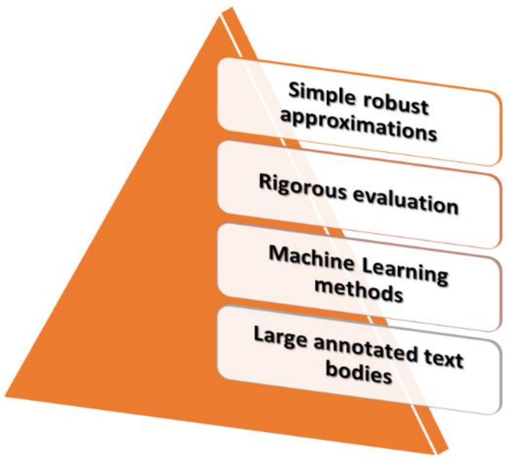


Figure 2: NLP reorientation in the 1980s.

This reorientation led to the rise of statistical NLP where statistical parsing utilizes probability for the context-free grammar rules [13]. Each rule has an associated probability, which is typically derived via machine learning on a described text corpora. This is also considered to be a supervised machine learning approach that represents an important part in NLP. The advantage of such an approach is that very detailed rules are replaces with statistical-frequency information lookup to avoid the ambiguity that may arise.

A different approach is that the rules are created from the annotated data, which builds then a decision tree from the feature-vector data. The statistical parser evaluates the highest probability for a parse of a phrase or sentence and then utilizes this parse to process the sentence and assign a meaning. The probabilistic approach depends considerably on the context, however, so having an acceptable training corpus is essential [14]. A training set consisting of annotated text bodies from the Wall Street Journal may be unsuitable for formation evaluation, as many words and meanings are not incorporated into the training set.

The main advantage of statistical approaches in practice is that the algorithms train with real data and utilize the most common cases. This implies that the more abundant and representative the data are for the phrase or text under consideration, the better they get. Another advantage is that unfamiliar or erroneous input may lead to lesser challenges, given that they indicate a low probability of matching. Handwritten rule-based and statistical approaches are complementary with each other, which is crucial for the success of NLP approaches.

III. Application of NLP in Reservoir Formation Evaluation

Within NLP there are typically several sub-problems that can be gradually addressed and solved, such as speech synthetics and connected speech recognition. Question answering, especially in technical domains, represents a major challenge.

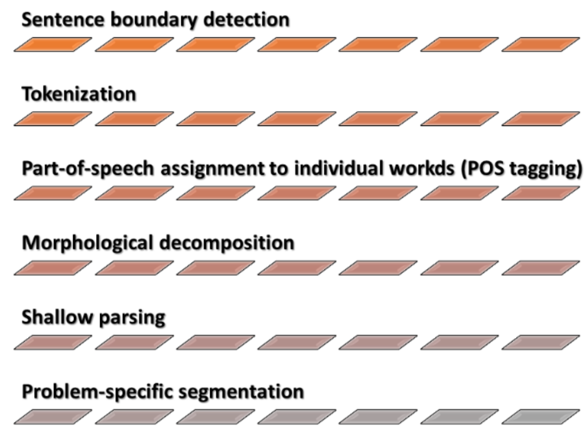


Figure 3: Low-level NLP tasks for reservoir formation evaluation.

Conventional low-level NLP (Figure 3) tasks involve sentence boundary detection, where the end of a sentence is to be looked for [15]. Conventionally, this is rather simple, given that a full stop ends a sentence. Abbreviations and titles represent a considerable challenging task in addition to items in a list.

Another task is tokenization, which identifies the individual tokens in a sentence. These tasks are conventionally covered by lexers. However, characters, such as dashes and forward slashes, may cause issues as they do not necessarily separate different tokens.

Part of speech tagging represents another challenge as they may represent a verb as well as a noun in certain circumstances. This involves the use of -ing that may be used in both verbs and nouns [16].

Another challenge is morphological decompositions of compound words that require a decomposition of the word to comprehend them. This is especially true for technical disciplines that contain many technical terms, which are hard to understand by themselves. Lemmatization typically helps in this context, but this depends on the language under consideration [17].

Shallow parsing is another low-level NLP task that identifies phrases from tokens that are tagged as part of the speech. For example, an adjective may precede a noun, which it describes [18].

Segmentation, according to specific problems, represents another challenge that is rather low-level.

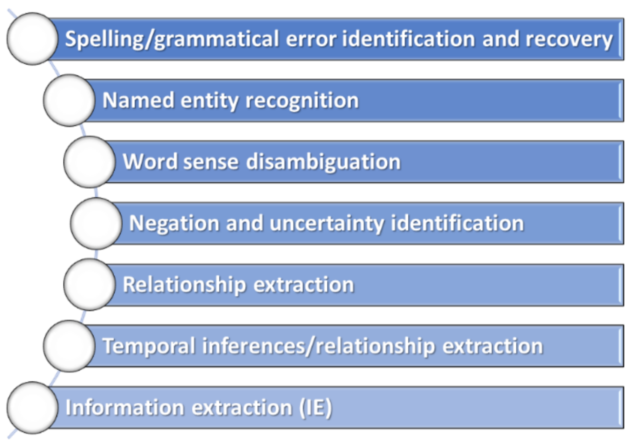


Figure 4: High-level tasks for NLP in reservoir formation evaluation.

Higher-level tasks build on low-level tasks and are usually problem-specific (Figure 4). They include:

1) **Spelling/grammatical error identification and recovery:** This is a very interactive task but is not that perfect from an implementation perspective. These phrases may lead to false positives, which are words that are correct but are flagged as false. Homophones may be used incorrectly and lead to false negatives. Typically, a homophone for reservoir formation evaluation is "their" and "there" [12].
2) **Named entity recognition (NER):** NER stands for the identification of entities, which are specific words or phrases, and then categorizes them into entities, such as persons, machines, locations, etc. The most common task is to develop a mapping between the named entities and concepts in a vocabulary, which partially utilizes shallow parsing. This may be separated into multiple phrases, however [19].
3) Some major issues that are faced in NER are:
   - Word/phrase order variation: This may be, for example, formation reservoir evaluation in contrast to reservoir formation evaluation
   - Derivation: This may lead to the derivation of suffixes
   - Inflection: This may be, for example, changes in numbers
   - Synonymy is abundant in formation evaluation and engineering.
   - Homographs**:** Homographs with related meanings are called "polysemy" and there are numerous examples of such.
4) Word sense disambiguation (WSD): This involves the determination of the correct meaning.
5) Negation and uncertainty identification: uncertainty identification has become essential, as synonyms or named entities are widespread encountered. Determining the absence or presence as well as quantifying the inference's uncertainty is a major challenge. Negations, on the other hand, can be explicit but can also be expressed in the form of uncertainty, which allows one to hedge. When talking about uncertainty, one determines that the reasoning process is hard to understand.
6) Relationship extraction: A crucial part is to determine relationships between an entity and events that are taking place. This is commonly encountered in formation evaluation and referencing to a thesauri or databases typically assists in overcoming this challenge and helps to extract the relationships.

Another sub-task for determining relationships between entities that are hierarchically related is called anaphora reference resolution [20]. This includes:

- Identity: In formation evaluation there are many instances where there are pronouns that refer to a named entity or where an abbreviation is used after the first time mentioning.
- Part/whole: This occurs when there is a location within a field;
- Superset/subset: For example, formation evaluation, logging.

7) Temporal inferences/relationship extraction: This refers to the inference from expressions or relations that are temporal. In particular, studying the past may allow inferring whether an event may occur in the future again or order the narrative.
8) Information extraction (IE): This refers to the identification of information that is problem-specific or focuses on the transformation into a structured form [21].

IV. Statistical Machine Learning – Data-Driven Approaches

Statistical and machine learning is a well-known area that involves the development of algorithms for inferring patterns from data. This shall help to be able to generalize and make predictions for new data via learning from the previously recorded data [22]. The process is typically separated into the training and prediction phase, where the parameters of the algorithm are optimized in order to minimize the discrepancy between the expected numerical target and the estimated.
The learning can be either supervised or unsupervised. In the supervised instance, the items in the training data are correctly labeled. In the unsupervised instance, the training data are not labeled, and the training process tries to determine the pattern automatically. This may be in the form of a cluster or factory analysis, or various other approaches [23].
One of the major challenges faced for any learning approach is overfitting. This implies that the model fits the data almost perfectly; however, the predictions for new data are rather poor. This is a major challenge if the data are not

representative of the instances one will face in the future or if the model is very erratically behaving [12].
This is primarily due to the fact that the models learn the random noise in the training data instead of retrieving the essential features that are desired. A great way to overcome the challenge of overfitting is to utilize cross-validation, which partitions the training dataset into tests and training sets where these are then internally validated. When repeating this process over several rounds, wherein each step the data are partitioned randomly, it allows to obtain a better average of the performance of the model and improve it [22].
Machine learning can be further classified according to how the probability distributions are utilized. Generative methods have the aim to create probability distributions for models, which allows the model to create synthetic data with these probability distributions. A more utilitarian approach is to use discriminative methods that estimate based on the observations directly the posterior probability.
In natural language processing, a generative approach would be to utilize in-depth knowledge of various languages to determine the undetermined language of a speaker, while a discriminative approach would utilize the difference between the various languages and the spoken language and then try to find the closest match.
The challenge of generative models is that they relatively easily become intractable for more features. In contrast, discriminative models have the benefit that they allow more features.
Typical examples of discriminative methods are logistic regression and conditional random fields (CRFs), while generative methods encompass Naïve Bayes classifiers and hidden Markov models (HMMs) [22].
There are, however, several major machine learning methods that are most often used for natural language processing tasks in formation evaluation [13].

Support vector machines (SVMs)

Learning via a discriminative approach is achieved via support vector machines (SVMs) that utilize inputs, such as words, to classify them into categories. This may be part of speech or other classification forms. The input in the SVM is conventionally transformed in order to enable the linear separation of the data into various categories. A crucial part of this is the transformation function, also called the kernel function, that transforms the data [24].
To outline the application of support vector machines, in a two-feature case, such as classifying a written report in terms of whether it categorizes a productive formation or nonproductive, typically can be separated by a straight line if solely two input features are utilized (see Figure 5). For the case of N-features, the separator will be conventionally an N-1 hyperplane, where the separating hyperplane aims to maximize the distance between the support vectors for each category. The support vectors are the data points that are closest to the hyperplane that differentiates each category. The most widely utilized kernel function for the transformation utilizes the normal distribution, given that in lots instances, the data are normally distributed [25].

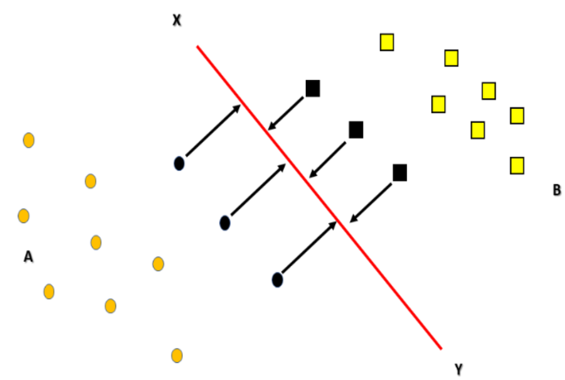


Figure 5: Support vector machines: We outline a 2-D case, where the points are separated by a straight line. The data are categorized in two categories, specifically category A (circles), and category B (diamonds). The data points can be separated by a straight line in the 2-dimensional plot. The SVM algorithm identifies the points that are closest to different categories and then determines the line that maximizes the margin between both sides. Linear separation may not always be feasible. Hence a transformation via a kernel function is necessary. This requires, in many instances, a trial and error approach in case the distribution of the data or transformation to allow linear separation is unknown.

Hidden Markov models (HMMs)

Hidden Markov models are systems that allow variables to move between different states that leads to various output possibilities. The move between the various states depends on the probabilities of the moves, which then also encounters various probabilities. The word "hidden" in HMM refers to that the system's state-switch probabilities and output probabilities being hidden, while only the outputs are known. While the number of possible states and unique identifiers may be large, they are still finite and known (see Figure 6) [26]. There are several crucial aspects in hidden Markov models.

- **Inference:** Inference refers to the computation of the probabilities of one or multiple candidates for a state-switch sequence.
- **Pattern matching:** Pattern matching refers to the switch sequence between the states that are with a high probability generating the output-symbol sequence.
- **Training:** Whenever the output-symbol sequence data are known, then the state-switch/output probabilities can be computed in terms of that it best fits the data.

The pattern matching and the training are similar to Naïve Bayesian reasoning extended to sequences, which can then be considered a generative model [27]. The main simplifying assumptions utilized for HMM are that

1) the state switching probability depends strongly on the states previously. This may also allow to switch back to the same state. In the simplest case, where there is only one state, the current state alone determines the probability. Hence, HMMs of the first order are very useful for situations where the likelihood solely depends on the last event and not the previous.
2) A specific output has a probability that solely depends on the state and no other state.

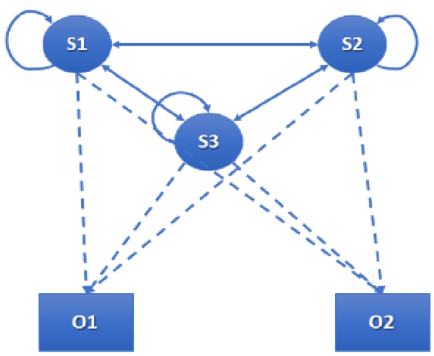


Figure 6: A graphical illustration of hidden Markov models. The rectangles with the letter O refer to the output values, whereas the circles starting with the letter S represent the states. The solid lines represent the state switches between connected states, where the arrow allows to indicate the switch's direction. It is noteworthy that the states may switch back to themselves with a certain probability. The probability may differ for the various lines. The dashed lines connect the states to the output values, which allows inferring the output probability. Important to note is that the sum of the probabilities of a switch leaving it is equal to 1, as this ensures consistency that all possible state transitions are considered.

The underlying assumptions enable to easily calculate the probability of a state switch sequence via simple multiplication, which can be easily addressed with algorithms such as the Viterbi algorithm. There are various problems in reservoir formation evaluation, in particular when considering the sensing part, that can be addressed with these existing algorithms [28].

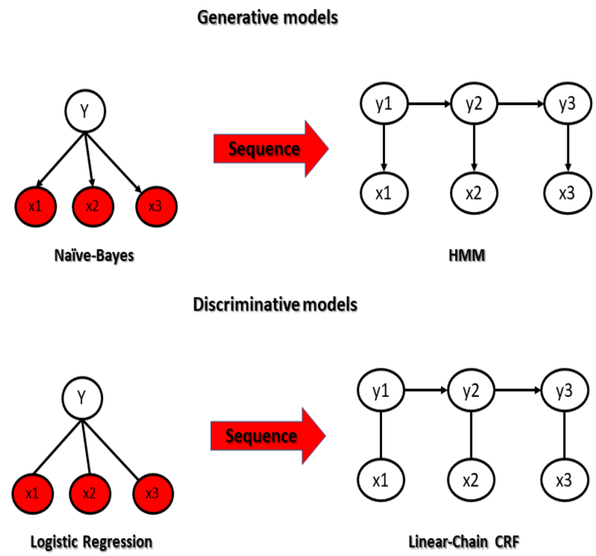


Figure 7: We outline the relationship between the Naïve Bayes, logistic regression, and conditional random fields. Naïve Bayes and Logistic regression distinguish each other from that Naïve Bayes is a generative model, while logistic regression is a generative model, which can be either transformed for sequences in a hidden Markov model or into a linear conditional random fields model. The dependence is indicated in both instances by the directional arrows that show the dependence between the various states.

The extension of HMMs to multivariate scenarios is possible. However the challenge arises from the potential intractability of the training problem. This leads to that multiple-variable applications deploy single variables, partially artificial, in order to determine the composites of the categorical variables. This requires much more training data to be available. When referring to speech recognition, the word's waveform, in terms of how it is spoken, is then connected to a sequence of the individual states (phonemes) that may be best at reproducing it. While speech recognition has improved significantly, formation evaluation still faces challenges in the field due to the complex terminologies and similarities between words.

Conditional random fields (CRFs)

Conditional random fields are discriminative forms, where the linear chain form of CRFs resemble hidden Markov models in that the next state solely depends on the current state. This indicates a linear dependency, which allows for fast and efficient computation. The conditional random fields are primarily a generalization of logistic regression to sequential data as compared to the previous discussion of the extension of Naïve Bayes to HMM (see Figure 7).

CRFs are widely applied to NER challenges, where the state variables are the categories of the named entities. Then the objective is to predict the sequence of named entity categories within the phrase or word pattern. The observation may involve prefixes and suffixes, as well as capitalization and embedded numbers. Hyphenation may be applied. For formation evaluation aspects, a well needs to be succeeded by an entity that has to be a number. If the field is called "Resfield," then to indicate a specific well, the named entity needs to be followed by a number, such that it states, for example, "Resfield 1." The main benefit from CRF is that it can be easier applied to sequential multivariate data as compared HMMs, as the training problem will be tractable.

N-grams

N-grams are powerful tools in statistical machine learning, where an n-gram is a sequence of n items that may consist of letters, words, or phonemes. Certain item pairs may occur with various statistical frequencies, where the relationship between various characters may be easily determined. This connection depends on the language under consideration, as certain combinations of word characters are rather unusual. The challenge in reservoir formation evaluation is that there are lots of abbreviations which makes the distribution broader. However, if sufficient data are available, then the frequency distribution for the n-grams can be computed. The permutations may increase dramatically, as in English alone, there are 26^2 letter pairs alone, which n-tuplets amounting to 26^n possible forms. This shows that n-grams depend on the n-th position on the previous n-1 items that were computed from the data.
The n-gram data has several purposes:

- Auto-completion suggestions of words, phrases, wells, etc. that are widely encountered on smartphones.
- Correction of misspelled words or names can be done automatically. This may also refer to reservoir names.
- For speech recognition, the ambiguity can be reduced based on determining the neighboring words
- probabilistically.
- The word "well" may have different meanings. In formation evaluation, it primarily relates to a noun, while in normal English, it is typically referred to as an adverb. Given the non-ambiguous neighboring words, the correct meaning of the homograph can be easily determined.

The challenge with n-grams is that they are voluminous, and this may become a challenge when retrieving the data. With modern data structures, such as n-gram indexes, searching of such data can be significantly sped up. The advantage is that n-grams need relatively little linguistic and domain knowledge.

V. NPL ANALYTICAL PIPELINES

For any NLP tasks, there are typically several sub-tasks that need to be focused on. These sub-problems require these low-level tasks to be executed sequentially before any higher-level task can be started. Hence, a pipelined design system is crucial as the output of one module may be connected to another module, which allows for mixing and matching of the various approaches.
For example, a CRF may be combined with a named entity recognition framework, which improves robustness. A single module could be easily replaced with another without having to conduct substantial changes to the remainder of the system [29].
A famous pipelined NLP framework is the Unstructured Information Management Architecture, where the scope allows to integrate structured-format databases, images, and multi-media, in addition to arbitrary technology.
This becomes even more important for reservoir formation evaluation applications that require a multi-step pipelined approach to move from voice interpretation over to technical specification understanding, over to automatic interpretation and recommender engines.

VI. The future of NLP in Formation Evaluation

Recent advances in artificial intelligence have outlined the importance of NLP in formation evaluation, and the huge potential encountered in the area. The large disk capacities, as well as data compression and efficient search allows modern statistical NLP methods to mimic human thoughts and speech patterns.
Multipurpose NLP technology will become mainstream for well log interpretation, and well report summary creation, which will also incorporate the automatic analysis of drilling reports for crucial information.

VII. Conclusions

Natural Language processing has in the last century undergone a revolution from being a fringe technology to powering many tools and services in today's environment. Formation evaluation represents a crucial area where natural language processing can play a vital role for enhancing interpretation and subsurface understanding. This will go beyond just pure textual understanding over to automatic speech recognition and interpretation, as well as hands-free tool deployment and automation.

VIII. References

[1] K. Katterbauer, A. Marsala, R. Alyami, R. Al Zaidy, "An overview of natural language processing driven approaches towards assisted formation evaluation interpretation", in Proc. *ICSNC, 2020.*
[2] C. Manning, P. Raghavan and H. Schuetze, Introduction to Information Retrieval, Stanford: Cambridge University Press, 2008.
[3] J. Hutchins, "The first public demonstration of machine translation: the Georgetown-IBM system," 7th January 1954.

[4] N. Chomsky, "Three models for the description of language," *IRE Transactions on Information Theory,* vol. 2, no. 3, pp. 113-124, 1956.

[5] R. Heckendorn, "A practical tutorial on conext free grammars," University of Idaho, 2020.

[6] J. Friedl, Mastering Regular Expressions, New York: O'Reilly Media, 2006.

[7] M. Zahran, "Lexical Analysis I," New Yrok University, New York, 2012.

[8] H. Bordihn, "Mildly Context-Sensitive Grammars," *Formal Languages and Applications,* pp. 163-173, 2004.

[9] J. Levine, Flex & Bison: Text Processing Tools, O'Reilly Media, 2009.

[10] S.-i. Morimoto and M. Sassa, "Yet another generation of LALR parsers for regular right part grammars," *Acta Informatica,* vol. 37, no. 9, pp. 671-697, 2001.

[11] M. Bramer, Logic Programming with Prolog, Portsmouth: Springer, 2005.

[12] D. Klein, "A core-tools statistical NLP course," in *Proceedings of the Second ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL*, 2005.

[13] W. Heng, "Natural Language Processing - Introduction and Practice," Amazon China, Beijing, 2013.

[14] P. Kantor, "Foundations of statistical natural language processing," *Information Retrieval,* vol. 4, no. 1, p. 80, 2001.

[15] B. Larsen, "A trainable summarizer with knowledge acquired from robust NLP techniques," *Advances in automatic text summarization,* p. 71, 1999.

[16] M. Straka and J. Strakova, " Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe," in *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies,* 2017.

[17] A. Voutilainen, "Part-of-speech tagging," *The Oxford handbook of computational linguistics,* pp. 219-232, 2003.

[18] E. Roche and Y. Schabes, Finite-state language processing., MIT Press, 1997.

[19] J. Hammerton, M. Osborne, S. Armstrong and W. Daelemans, "Introduction to Special Issue on Machine Learning Approaches to Shallow Parsing," *Journal of Machine Learning Research,* vol. 2, no. 4, 2002.

[20] A. Mansouri, L. S. Affendey and A. Mamat, "Named entity recognition approaches," *International Journal of Computer Science and Network Security 8,* vol. 2, pp. 339-344, 2008.

[21] L. Pineda and G. Garza, "A model for multimodal reference resolution," *Computational Linguistics,* vol. 26, no. 2, pp. 139-193, 2000.

[22] C. Cardie, "Empirical methods in information extraction," *AI magazine,* vol. 18, no. 4, p. 65, 1997.

[23] S. Masashi, Introduction to statistical machine learning., Morgan Kaufmann, 2015.

[24] J. Lafferty and L. Wasserman, "Challenges in statistical machine learning," *Statistica Sinica,* vol. 16, no. 2, p. 307, 2006.

[25] W. Noble, "What is a support vector machine?," *Nature biotechnology,* vol. 24, no. 12, pp. 1565-1567, 2006.

[26] L. Wang, Support vector machines: theory and applications, Springer Science & Business Media, 2005.

[27] O. Cappe, E. Moulines and T. Ryden, Inference in hidden Markov models, Springer Science & Business Media, 2006.

[28] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE,* vol. 77, no. 2, pp. 257-286, 1989.

[29] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks," *Hidden Markov models: applications in computer vision,* pp. 9-41, 2001.

Using Static Analysis and Static Measurement for Industrial Software Quality Evaluation

Luigi Lavazza
Dipartimento di Scienze Teoriche e Applicate
Università degli Studi dell'Insubria
Varese, Italy
Email: luigi.lavazza@uninsubria.it

***Abstract*—Business organizations that outsource software development need to evaluate the quality of the code delivered by suppliers. In this paper, we illustrate an experience in setting up and using a toolset for evaluating code quality for a company that outsources software development. The selected tools perform static code analysis and static measurement, and provide evidence of possible quality issues. To verify whether the issues reported by tools are associated to real problems, code inspections were carried out. The combination of automated analysis and inspections proved effective, in that several types of defects were identified. Based on our findings, the business company was able to learn what are the most frequent and dangerous types of defects that affect the acquired code: this knowledge is now being used on a regular basis to perform focused verification activities.**

***Index Terms*—Software quality; Static analysis; Software measurement; Code clones; Code measures.**

I. Introduction

Today, software is necessary for running practically any kind of business. However, poor quality software is generally expensive, because it can cause expensive failures, increased maintenance cost and security breaches. Hence, organizations that rely on software for running their business need to keep the quality of their software under control.

Many companies do not have the possibility or the will of developing the software they need. Hence, they outsource software development. In such case, an organization has no direct visibility and control of the development process; instead, they can only check the quality of the product they receive from developers.

In this paper, we report about an experience in setting up the toolset needed for evaluating the quality of the code provided by a supplier to an organization. This paper details and extends a previous report [1].

The software involved in the reported activities is used by the organization to run two Business-to-Consumer (B2C) portals. The organization needed to evaluate the quality of the supplied code; specifically, they wanted to check that the code was correctly structured, cleanly organized, well programmed, and free from defects that could cause failures or vulnerabilities that could be exploited by attackers. The organization had already in place a testing environment to evaluate the quality of the code from the point of view of behavioral correctness. They wanted to complement test-based verification with checks on the internal qualities that could affect maintainability, fault-proneness and vulnerability.

To accomplish these goals, we selected a set of tools that provide useful indications based on static analysis and measurement of code. The toolset was intended to be used to evaluate two releases of the portal, and then to be set up at the company's premises, and used to evaluate the following releases.

The contributions of the paper are threefold:

1) We provide methodological guidance on the selection of a small set of tools that can provide quite useful insights on code quality.
2) We show how to use the tools and how to combine the automated analysis with manual verifications.
3) We provide some results, which can give the reader a measure of the results that can be achieved via the proposed approach.

Because of confidentiality constraints, in this paper we shall not identify the involved parties, and we shall omit some non-essential details.

The paper is structured as follows. In Section II, we provide some details concerning the evaluated software and the goals of the study. In Section III, the tools used for the static analysis and measurement are described. Section IV illustrates the methodological approach. In Section V, the results of the evaluation are described. Section VI provides some suggestions about the organization of a software development process that takes advantage of a static analysis and measurement toolset. Section VII illustrates the related work, while Section VIII draws some conclusions and outlines future work.

II. The Object of the Evaluation

In this section, we describe the code evaluation issues that were tackled in the problem context.

The evaluation addressed two B2C portals, coded almost entirely in Java. The analyses concentrated exclusively on the Java code. Table I provides a few descriptive statistics concerning the two portals, where LOC is the total number of lines of code, while LLOC is the number of logical lines of code, i.e., the lines that contain executable code (excluding blank lines, comments, etc.).

TABLE I. CHARACTERISTICS OF THE ANALYZED PORTALS.

	Portal 1	Portal 2
Number of files	1,507	280
LLOC	100,375	37,467
LOC	202,249	55,934
Number of Classes	1,158	247
Number of Methods	13,351	5,370

The aim of the study consisted in evaluating the quality of the products, highlighting weaknesses and improvement opportunities. In this sense, it was important to spot the types of the most frequently recurring issues, rather than finding *all* the actual defects and issues.

It was also required that the toolset could be seamlessly transferred to the company's premises. To this end, open-source (or free to use) software was to be preferred.

Accordingly, we looked for tools that can

– Detect bad programming practices, based on the identification of specific code patterns.
– Detect bad programming practices, based on code measures (e.g., methods too long, classes excessively coupled, etc.).
– Detect duplicated code.
– Identify vulnerabilities.

III. TOOLS USED

After some search and evaluation, we selected the tools mentioned in Table II. These tools are described in some detail in the following sections.

A. Static Analysis for Identifying Defects

SpotBugs (formerly known as FindBugs) is a program which uses static analysis to look for bugs in Java code [2][3]. SpotBugs looks for code patterns that are most likely associated to defects. For instance, SpotBugs is able to identify the usage of a reference that is possibly null.

SpotBugs was chosen because it is open-source and one among the best known tools of its kind. Besides, SpotBugs proved to be quite efficient: on a reasonably powerful laptop, it took less than a minute to analyze Portal 1.

SpotBugs provides "confidence" and "rank" for each of the issued warnings. Confidence indicates how much SpotBugs is confident that the issued warning is associated to a real problem; confidence is expressed in a three-level ordinal scale (high, medium, low). The rank indicates how serious the problem is believed to be. The rank ranges from 1 (highest) to 20 (lowest); SpotBugs also indicates levels: "scariest" ($1 \leq$rank≤ 4), "scary" ($5 \leq$rank≤ 9), "worrying" ($10 \leq$rank≤ 14), "of concern" ($15 \leq$rank≤ 20).

SpotBugs is quite expressive. Figure 1 shows an example of the output provided by SpotBugs. The upper-left window shows the hierarchy of bugs, organized by type and/or by rank. The upper-right window shows the code, highlighting the part that is responsible of the bug. The bottom right window gives the "conceptual" description of the bug type: in the show case the problem is due to using `String.equals` to compare objects having different types. The bottom-left windows explains why the shown code was recognized as responsible for the bug: it reports the file and line number where the problem was found, and which code elements are involved in the detected issue.

The provided information indicates clearly which problem was identified and where: usually it is easy for developers to manually check whether the detected bug is a real problem or a false positive. In fact, static analysis cannot provide exclusively correct indications. To avoid overlooking important problems whose occurrence the tool cannot decide, the provided results are usually somewhat "pessimistic." That is, some indications are false positives, and do not correspond to actual bugs. Because of this intrinsic characteristic of static analysis tools, manual verification of the reported issues is advisable. To get an idea of the level of precision yielded by SpotBugs, experimental evaluations found that the SpotBugs' precision is between 58% and 80% [4].

SpotBugs can save results in XML files. It is thus possible to write specific code to read and analyze such results. As an example of this kind of analysis, we wrote a script in the R language [5] to graphically show the distribution of bugs through code files. The result is given in Figure 2, where the file names have been hidden for confidentiality. The size of each box is proportional to the maximum gravity of the bugs it contains, while the color indicates the number of bugs: the more bugs in the file, the darker the box. This kind of representation was useful to drive the attention of project managers onto the most critical files.

B. Static Analysis for Identifying Vulnerabilities

Having selected SpotBugs as a static analyzer, it was fairly natural to equip it with FindSecBugs [6], a plug-in that addresses security problems.

FindSecBugs works like FindBugs and SpotBugs, looking for code patterns that can be associated to security issues, with reference to problems reported by the Open Web Application Security Project (OWASP) [7] an the weaknesses listed in the Common Weaknesses Enumeration (CWE) [8].

C. Static Measures of Code

Several tools are available to measure the most relevant characteristics of code, including size, complexity, coupling, cohesion, etc.

SourceMeter [9] was chosen because it is free to use, efficient and provides many measures, including all the most popular and relevant. Of the many measures that SourceMeter can provide, we used those listed in Table III. Specifically, we selected a relatively small set of measures that cover all the potentially interesting aspects of code, including size, complexity, cohesion, and coupling.

TABLE II. TOOLS USED AND THEIR PURPOSE.

Purpose	Tool	Main features
Identify defects	SpotBugs	Static analysis is used to identify code patterns that are likely associated to defects.
Collect static measures	SourceMeter	Static measurement is applied at different granularity levels (class, method, etc.) to provide a variety of measures.
Detect code clones	SourceMeter	Structurally similar code blocks are identified.
Identify security issues	FindSecBugs	A plug-in for FindBugs, specifically oriented to identifying vulnerable code.

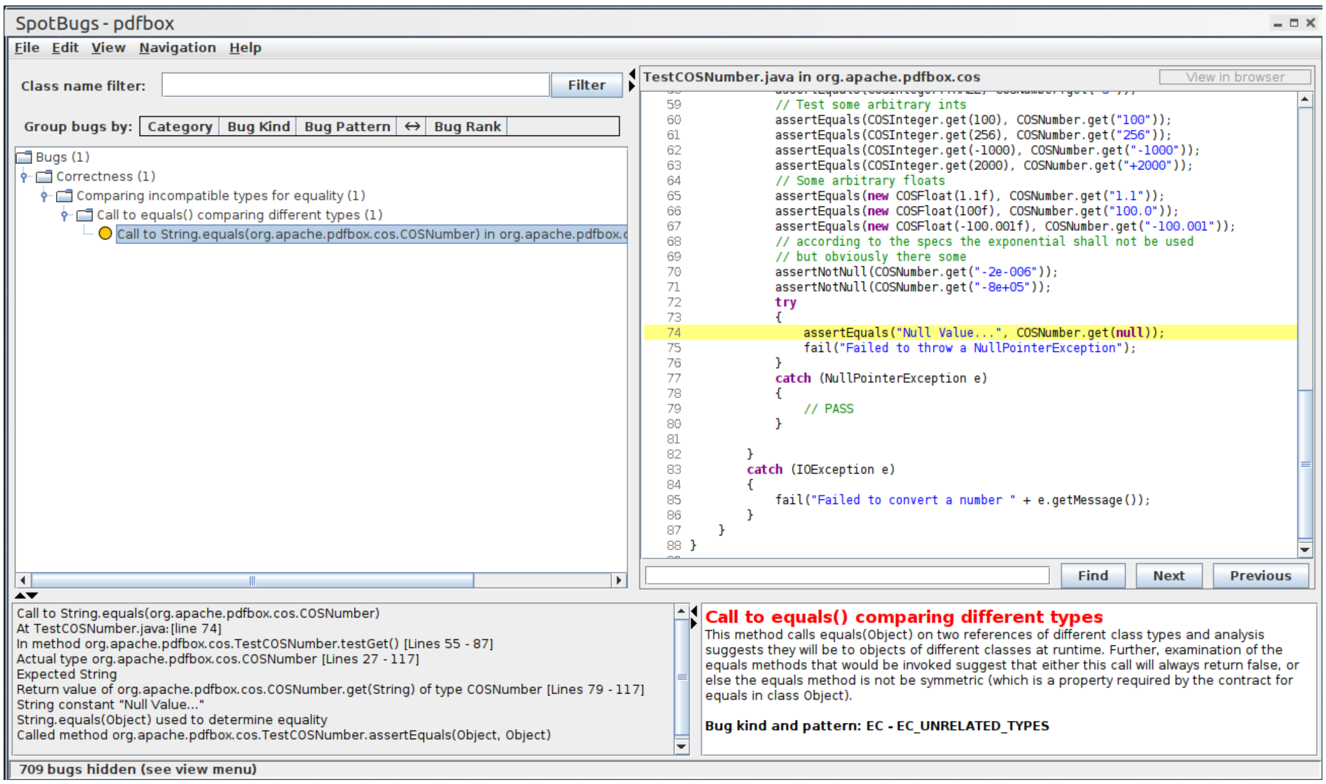


Fig. 1. A screenshot of SpotBugs.

TABLE III. Measures provided by SourceMeter and used in the evaluation.

Metric name	Abbreviation	Class	Method
Lack of Cohesion in Methods 5	LCOM5	X	
McCabe's Cyclomatic Complexity	McCC		X
Nesting Level Else-If	NLE	X	X
Weighted Methods per Class	WMC	X	
Coupling Between Object classes	CBO	X	
Response set For Class	RFC	X	
Depth of Inheritance Tree	DIT	X	
Number of Children	NOC	X	
Logical Lines of Code	LLOC	X	X
Number of Attributes	NA	X	
Number of Parameters	NUMPAR		X

D. Code Clone Detection

Noticeably, SourceMeter is also able to detect code clones. Specifically, SourceMeter is capable of identifying the so-called Type-2 clones, i.e. code fragments that are structurally identical, but may differ in variable names, literals, identifiers, etc.

IV. The Method

Since the most interesting properties of code are undecidable, tools that perform static analysis often issue warnings concerning problems that are likely—but not certain—to occur. In practice, the issues reported by static analysis tools can be false positives. Therefore, we always inspected manually the code that had been flagged as possibly incorrect by the tools.

The inspection of issues reported by static analysis tools was generally quick and required little effort, because the tools provide a very clear indication of the nature of the problem and where it is. Sometimes the inspections required a substantial amount of effort, but that is probably because we were not at all familiar with the code: the developers of the code would have completed the inspection much more quickly.

Also code that is characterized by unusual values of static measures needed inspections. For instance, consider a method that has unusually high McCabe complexity: only via manual inspection we can check whether the program was badly structured or the coded algorithm is intrinsically complex.

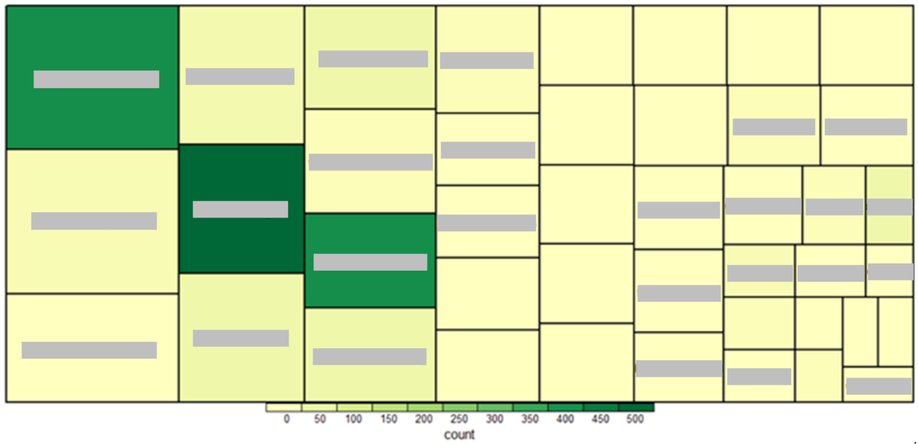


Fig. 2. A graphical representation of the distribution of bugs through code files.

In several cases static measures indicated as suspicious code elements that had already been flagged by other measures or by static analysis: in those cases, recognizing the nature of the problem (if any) was practically immediate.

Problem detection was performed as described in Figure 3. The real problems identified via the process described in Figure 3 were classified according to their type, so that the company that asked for the code quality analysis could focus improvement efforts on the most frequent and serious problems.

V. Results

Here, we describe the code quality problems that were identified.

A. Warnings issued by SpotBugs

Tables IV and V illustrate the number of warnings that SpotBugs issued for the analyzed code, respectively by confidence level and by rank. In Table IV, the density indicates the number of warnings per line of code.

SpotBugs also classifies warning by type (for additional information on warning types, see [10]). Table VI illustrates the warnings we obtained, by type. It can be noticed that most warnings concerned security (types "Security" and "Malicious code vulnerability").

1) Results deriving from the inspection of SpotBugs warnings: The effort allocated to the project did not allow analyzing all the warnings issued by SpotBugs. Therefore, we inspected the code where SpotBugs had identified issues ranked "scary" and "scariest." Specifically, we analyzed the warnings described in Table VII.

Our inspections revealed several code quality problems:

- The existence of problems matching the types of warning issued by SpotBugs was confirmed.
- Some language constructs were not used properly. For instance, class `Boolean` was incorrectly used instead of `boolean`; objects of type `String` were used instead of `boolean` and `enumeration`; etc.
- We found redundant code, i.e., some pieces of code were unnecessarily repeated, even where avoiding code duplication—e.g., via inheritance or even simply by creating methods that could be used in different places—would have been easy to use and definitely convenient.
- We found some pieces of code that were conceptually incorrect. The types of defect were not of any type that a static analyzer could find, but were quite apparent when inspecting the code.

Concerning the correctness of warnings issued by SpotBugs, we found just one false positive: the "comparison of `String` objects using == or `!=`" was not an error, in the examined case. We also found that the four instances of "Method ignores return value" were of little practical consequences. In summary, the great majority of warnings indicated real problems, which could cause possibly serious consequences. The remaining warning indicated situations where a better coding discipline could make the code less error prone, if applied systematically.

2) Results deriving from the inspection of FindSecBugs warnings: The great majority of the security warnings (types "Security" and "Malicious code vulnerability") were ranked by FindSecBugs as not very worrying. Specifically, no "scariest" warning was issued, and only one "scary" warning was issued. Therefore, we inspected the only "scary" warning (rank 7, see Table VIII), and all the warnings at the highest rank of the level "troubling" (rank 10, see Table VIII).

We found that all the warnings pointed to code that had security problems. In many cases, SpotBugs documentation provided quite straightforward ways for correcting the code.

B. Inspection of code elements having measures beyond threshold

Static measures concerning size, complexity, cohesion, coupling, among others, are expected to provide indications on the quality of code. In fact, one expects that code characterized by large size, high complexity, low cohesion, strong coupling and similar "bad" characteristics is error-prone. Accordingly, we inspected code elements having measures definitely out of

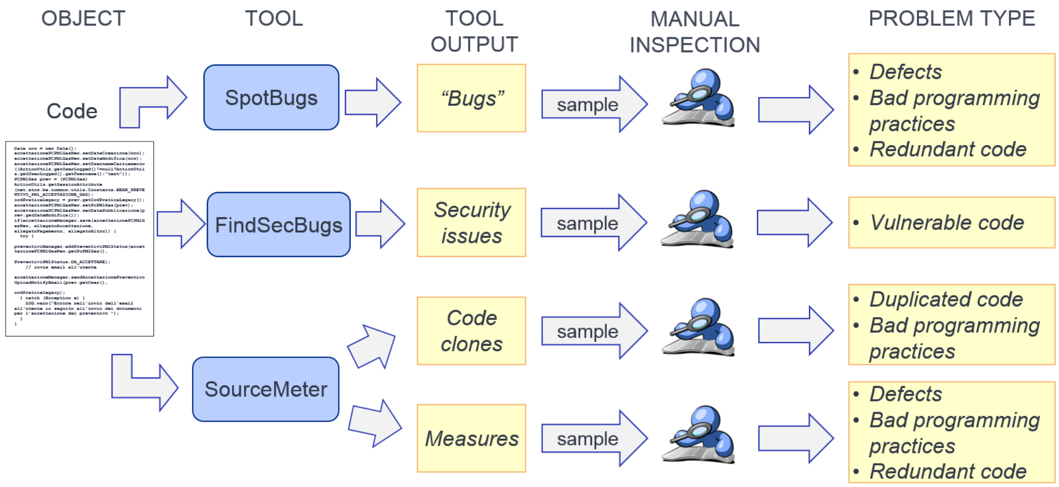


Fig. 3. The evaluation process: problem detection phase.

TABLE IV. SPOTBUGS WARNINGS BY CONFIDENCE.

	Portal 1		Portal 2	
Metric	Warnings	Density	Warnings	Density
High Confidence	68	0.07%	50	0.13%
Medium Confidence	774	0.77%	502	1.34%
Low Confidence	824	0.82%	420	1.12%
Total	1666	1.66%	972	2.59%

TABLE V. SPOTBUGS WARNINGS BY RANK

Rank	1	6	7	8	9	10	11	12	14	15	16	17	18	19	20
Portal 1	24	9	1	42		12	39	21	2	604	17	129	562	82	122
Portal 2	10			6	1	7	8	27	18	191	10	59	401	66	168

TABLE VI. SPOTBUGS WARNINGS BY TYPE.

	Portal 1		Portal 2	
Warning Type	Number	Percentage	Number	Percentage
Bad practice	73	4.38%	81	8.33%
Correctness	91	5.46%	30	3.09%
Experimental	0	0.00%	1	0.10%
Internationalization	16	0.96%	39	4.01%
Malicious code vulnerability	496	29.77%	316	32.51%
Multithreaded correctness	28	1.68%	1	0.10%
Performance	74	4.44%	143	14.71%
Security	631	37.88%	215	22.12%
Dodgy code	257	15.43%	146	15.02%
Total	1666	100%	972	100%

the usually considered safe ranges. Specifically, we considered McCabe complexity [11], Logical Lines of Code and Response for Class (RFC) [12] as possibly correlated with problems. In fact, we also looked at Coupling Between Objects, Lack of Cohesion in Methods and Weighted Method Count, but these measures turned out to provide no additional information with respect to the aforementioned three measures, i.e., they pointed to the same classes or methods identified as possibly problematic by the aforementioned measures.

We found that several methods featured McCabe complexity well over the threshold that is generally considered safe. Figure 4 shows the distributions of McCabe complexity of methods (excluding setters and getters) together with two thresholds representing values that should and must not be exceeded, according to common knowledge [11][13][14][15]. Specifically, we found methods with McCabe complexity close to 200.

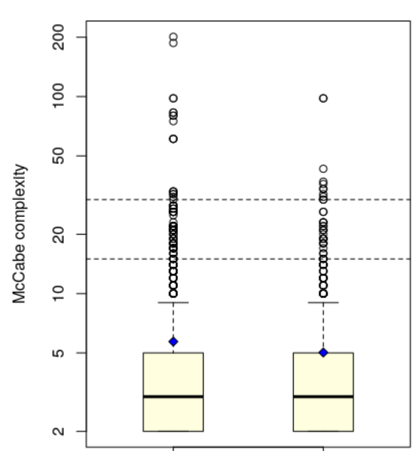


Fig. 4. Boxplots illustrating the distributions of McCabe complexity in the two portals (blue diamonds indicate mean values). The scale is logarithmic.

TABLE VII. SPOTBUGS WARNINGS THAT WERE VERIFIED MANUALLY.

Rank	Type	Occurrences Portal 1	Portal 2
1	Suspicious reference comparison	10	9
1	Call to `equals()` comparing different types	14	1
6	Possible null pointer dereference	8	–
8	Possible null pointer dereference	–	2
8	Method ignores return value	–	4
9	Comparison of `String` objects using == or !=	–	1

TABLE VIII. FINDSECBUGS WARNINGS THAT WERE VERIFIED MANUALLY.

Rank	Type	Occurrences Portal 1	Portal 2
7	HTTP response splitting vulnerability	1	–
10	Cipher with no data integrity	4	2
10	ECB mode is insecure	4	2
10	URL Connection Server-Side Request	1	–
	Forgery and File Disclosure	–	–
10	Unvalidated Redirect	2	–
10	Request Dispatcher File Disclosure	–	1

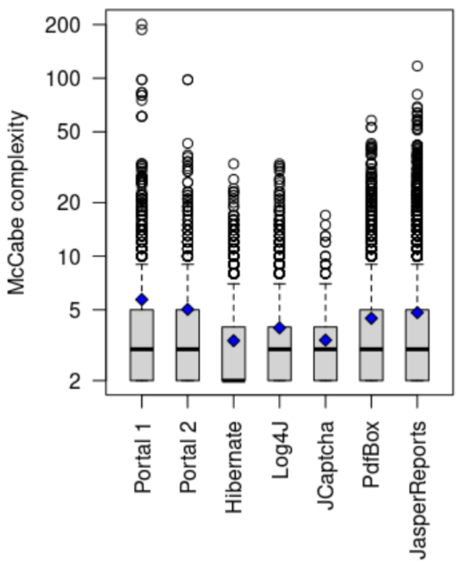


Fig. 5. Boxplots illustrating the distributions of McCabe complexity in the two portals in comparison with a few popular open-source products. The scale is logarithmic.

The values of McCabe complexity in the evaluated portals may seem exceedingly large. To assess these values, we compared them to the McCabe complexity of a few popular open-source projects, which are widely used and generally considered of good quality. The comparison is given in Figure 5. It can be seen that the distributions of McCabe complexity values in the evaluated portals is not very different from other programs'. However, in all cases the methods having McCabe complexity greater than 10 are considered outliers: they deserve careful quality analysis.

When considering size, we found several classes featuring over 1000 LLOC; the largest class contained slightly less then 6000 LLOC. When considering RFC, we found 12 classes having RFC greater than 200. Interestingly, the class with the highest RFC (709) was also the one containing the method with the greatest McCabe complexity. The biggest class contained the second most complex method. These results were not surprising, since it is known that several measures are correlated to size.

Inspections revealed that the classes and methods featuring excessively high values of LLOC, RFC and McCabe complexity were all affected by the same problem. The considered code had to deal with several types of services, which where very similar under several respects, although each one had its own specificity. The analyzed code ignored the similarities among the services to be managed, so that the code dealing with similar service aspects was duplicated in multiple methods. The code could have been organized differently using basic object-oriented features: a generic class could collect the features that are common to similar services, and a specialized class for every service type could take care of the specificity of different service types.

In conclusion, by inspecting code featuring unusual static measures, we found design problems, namely inheritance and late binding were not used where it was possible and convenient.

C. *Inspection of duplicated code*

SourceMeter was also used to find duplicated code. Specifically, structurally similar blocks of 10 or more lines of code where looked for. Many duplicated blocks were found. For instance, in Portal 1, 434 duplicated blocks were found. In many cases, blocks included more than one hundred lines. The largest duplicated blocks contained 205 lines. A small minority of detections concerned false positives.

We found three types of duplications:

a) Duplicates within the same file. That is, the same code was found in different parts of the same file (or the same class, often).
b) Duplicates in different files. That is, the same code fragment was found in different files (of the same portal).
c) Duplicates in different portals. That is, the same code fragment was found in files belonging to different portal.

Duplicates of type c) highlighted the existence of versioning problems: different versions of the same class were used in the two portals.

Duplicates of types a) and b) pointed to the same type of problem already identified, i.e., not using inheritance to factor code that can be shared among classes dealing with similar services. Concerning this issue, it is worth noting that static measures revealed a general problem with the design of code,

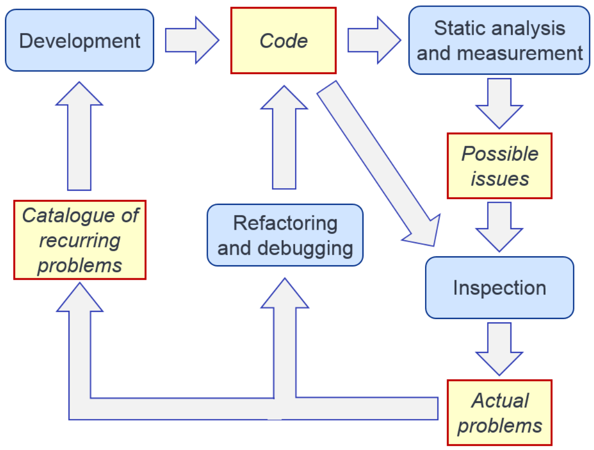


Fig. 6. Suggested Development Process.

but were not able to indicate precisely which parts of the code could be factorized. On the contrary, duplicated code detection was quite effective in identifying all the cases where code could be factorized, with little need of inspecting the code. In this sense, code clone detection added some value to inspections aiming at understanding the reasons for 'out of range' measures.

VI. Suggestions for Improving the Development Process

Given the results described in Section V, it seems convenient that the capabilities of static analysis and measurement tools are exploited on a regular basis. To this end, we can observe that two not exclusive approaches are possible.

1) Evaluation of code: The toolset can be used to evaluate the released code as described in Section IV. However, it would be advisable that developers verify their own code via SpotBugs and SourceMeter *before* releasing it: in such a way, a not negligible number of bugs would be removed even before testing and other Verification&Validation activities, thus saving time and effort. With respect to the evaluation described in Section IV, where just a sample of the issues reported by the tools were inspected, in the actual development process all issues should be inspected.

2) Prevention: The practice of issue identification and verification leads to identifying the most frequently recurring types of problems. It is therefore possible to compile a catalogue of the most frequent and dangerous problems. Accordingly, programmers could be instructed to carefully avoid such issues. This could imply teaching programmers specific techniques and good programming practices.

As a result of the considerations illustrated above, software development activities could be organized as described in Figure 6. If development is outsourced, as in the cases described in this paper, the catalogue of recurrent problems could be used as part of the contract annex that specifies the required code quality level.

Finally, it is worth noting that the proposed approach can be applied in practically any type of lifecycle. For instance, in an agile development environment, the proposed evaluation practices could be applied at the end of every sprint.

VII. Related Work

The effectiveness of using automated static analysis tools for detecting and removing bugs was documented by Zheng et al. [16]. Among other facts, they found that the cost per detected fault is of the same order of magnitude for static analysis tools and inspections, and the defect removal yield of static analysis tools is not significantly different from that of inspections.

Thung et al. performed an empirical study to evaluate to what extent could field defects be detected by FindBugs and similar tools [17]. To this end, FindBugs was applied to three open-source programs (Lucene, Rhino and AspectJ). The study by Thung et al. takes into consideration only known bugs, and is performed on open-source programs. On the contrary, we analyzed programs developed in an industrial context, and relied on manual inspection to identify actual bugs.

Habib and Pradel performed a study to determine how many of all real-world bugs do static bug detectors find [18]. They used three static bug detectors, including SpotBugs, to analyze a version of the Defects4J dataset that consisted of 15 Java projects with 594 known bugs. They found that static bug detectors find a small but non-negligible amount of all bugs.

Vetrò et al. [19] evaluated the accuracy of FindBugs. The code base used for the evaluation consisted of Java projects developed by students in the context of an object-oriented programming course. The code is equipped with acceptance tests written by teachers of the course in such a way that all functionalities are checked. To determine true positives, they used temporal and spatial coincidence: an issue was considered related to a bug when an issues disappeared at the same time as a bug get fixed (according to tests). In a later paper [20] Vetrò et al. repeated the analysis, with a larger code set and performing inspections concerning four types of issues found by FindBugs, namely the types of findings that are considered more reliable.

Tomassi [21] considered 320 Java bugs from the BugSwarm dataset, and determine which of these bugs can potentially be found by SpotBugs and another analyzer—namely, ErrorProne (https://github.com/google/error-prone)— and how many are indeed detected. He found that 40.3% of the bugs were of types that SpotBugs should detect, but only one of such bugs was actually detected by SpotBugs.

In general, the papers mentioned above have goals and use methods that are somewhat similar to ours, but are nonetheless different in important respects. A work that shares context, goals and methods with ours was reported by Steidl et al. [22]. They observed that companies often use static analyses tools, but they do not learn from results, so that they fail to improve code quality. Steidl et al. propose a continuous quality control process that combines measures, manual action, and a close cooperation between quality engineers, developers, and managers. Although there are evident differences between the work by Steidl et al. and the work reported in this paper (for instance, the situation addressed by Steidl et al. does not involve outsourcing), the suggestions for improving the development process given in Section VI are conceptually coherent with the proposal by Steidl et al.

Similarly, Wagner et al. [23] performed an evaluation of the effectiveness of static analysis tools in combination with other techniques (including testing and reviews). They observed that a combination of the usage of bug finding tools together with reviews and tests is advisable if the number of false positives is low, as in fact is in the cases we analyzed (many false positives would imply that a relevant effort is wasted).

An alternative to static analyzers like SpotBugs is given by tools that detect the presence of "code smells" [24] in code. A comparison of these types of tools was performed by applying SpotBugs and JDeodorant [25][26] to a set of set of open-source applications [4]. The study showed that the considered tools can help software practitioners detect and remove defects in an effective way, to limit the amount of resources that would otherwise be spent in more cost-intensive activities, such as software inspections. Specifically, SpotBugs appeared to detect defects with good Precision, hence manual inspection of the code flagged defective by SpotBugs becomes cost-effective.

Another empirical study evaluated the persistence of SpotBugs issues in open-source software evolution [27]. This study showed that around half the issues discovered by SpotBugs are actually removed from code. This fact is interpreted as a confirmation that SpotBugs identifies situations that are considered worth correcting by developers.

VIII. Conclusions

Evaluating the quality of software is important in general, and especially for business organization that outsource development, and do not have visibility and control of the development process. Software testing can provide some kind of quality evaluations, but to a limited extent. In fact, some aspects of code quality (e.g., whether the code is organized in a way that favors maintainability) cannot be assessed via testing.

This paper describes an approach to software quality evaluation that consists of two phases: in the first phase, tools are used to identify possible issues in the code; in the second phase, code is manually inspected to verify whether the reported issues are associated to real problems. The tools used are of two kinds: the first performs static analysis of code looking for patterns that are likely associated to problematic code; the second type yields measures of static code properties (like size, complexity, cohesion, coupling etc.), thus helping identifying software elements having excessive, hence probably problematic, characteristics.

The mentioned approach was applied to the code of the web portals used by a European company to let its customers use a set of services. The experience was successful, as tool-driven inspections uncovered several types of defects. In the process, the tools (namely SpotBugs and SourceMeter) identified problems of inherently different nature, hence it is advisable to use both types of tools.

Based on our findings, the business company was able to learn what are the most frequent and dangerous types of defects that affect the acquired code: this knowledge is being used to perform focused verification activities.

The proposed approach and toolset (possibly composed of equivalent tools) can be useful in several contexts where code quality evaluation is needed. Noticeably, the proposed approach can be used in different types of development process, including agile processes.

Among the future possible evolutions of this work, the most intriguing one concerns studying the possibility of replacing inspection via some sort of AI-based models that can discriminate false positives and true problems. Specifically, collecting the results of manual inspections driven by SpotBugs could lead to some sort of supervised learning.

Another interesting evolution of the presented work involves using automatic code smell detectors, like JDeodorant [28], for instance, in addition to SpotBugs.

Acknowledgment

This work has been partially supported by the "Fondo di ricerca d'Ateneo" of the Università degli Studi dell'Insubria.

References

[1] L. Lavazza, "Software quality evaluation via static analysis and static measurement: an industrial experience," in The Fifteenth International Conference on Software Engineering Advances—ICSEA, 2020, pp. 55–60.

[2] D. Hovemeyer and W. Pugh, "Finding bugs is easy," ACM Sigplan notices, vol. 39, no. 12, 2004, pp. 92–106.

[3] https://spotbugs.github.io/ [retrieved: August 2020].

[4] L. Lavazza, S. Morasca, and D. Tosi, "Comparing static analysis and code smells as defect predictors: an empirical study," in 17th IFIP International Conference on Open Source Systems (OSS). Springer International Publishing, 2021, pp. 1–15.

[5] R. C. Team et al., "R: A language and environment for statistical computing," 2013.

[6] https://find-sec-bugs.github.io/ [retrieved: August 2020].

[7] "Open Web Application Security Project® (OWASP)," https://www.owasp.org [retrieved: August 2020].

[8] "Common Weaknesses Enumeration," https://cwe.mitre.org [retrieved: August 2020].

[9] R. Ferenc, L. Langó, I. Siket, T. Gyimóthy, and T. Bakota, "Source meter sonar qube plug-in," in 2014 IEEE 14th International Working Conference on Source Code Analysis and Manipulation, 2014, pp. 77–82.

[10] https://spotbugs.readthedocs.io/en/stable/bugDescriptions.html/ [retrieved: August 2020].

[11] T. J. McCabe, "A complexity measure," IEEE Transactions on software Engineering, no. 4, 1976, pp. 308–320.

[12] S. R. Chidamber and C. F. Kemerer, "A metrics suite for object oriented design," IEEE Transactions on software engineering, vol. 20, no. 6, 1994, pp. 476–493.

[13] A. H. Watson, D. R. Wallace, and T. J. McCabe, Structured testing: A testing methodology using the cyclomatic complexity metric. NIST – National Institute of Standards and Technology, 1996, vol. 500, no. 235.

[14] M. Bray, K. Brune, D. A. Fisher, J. Foreman, and M. Gerken, "C4 software technology reference guide-a prototype." Carnegie-Mellon Univ., Pittsburgh PA, Software Engineering Inst, Tech. Rep., 1997.

[15] JPL, "JPL Institutional Coding Standard for the C Programming Language," Jet Propulsion Laboratory, Tech. Rep., 2009.

[16] J. Zheng, L. Williams, N. Nagappan, W. Snipes, J. P. Hudepohl, and M. A. Vouk, "On the value of static analysis for fault detection in software," IEEE transactions on software engineering, vol. 32, no. 4, 2006, pp. 240–253.

[17] F. Thung, L. Lucia, D. Lo, L. Jiang, P. Devanbu, and F. Rahman, “To what extent could we detect field defects? an extended empirical study of false negatives in static bug-finding tools,” Automated Software Engineering, vol. 22, no. 4, 2015, pp. 561–602.
[18] A. Habib and M. Pradel, “How many of all bugs do we find? a study of static bug detectors,” in Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, 2018, pp. 317–328.
[19] A. Vetrò, M. Torchiano, and M. Morisio, “Assessing the precision of FindBugs by mining java projects developed at a university,” in 2010 7th IEEE Working Conference on Mining Software Repositories (MSR 2010). IEEE, 2010, pp. 110–113.
[20] A. Vetrò, M. Morisio, and M. Torchiano, “An empirical validation of findbugs issues related to defects,” in 15th Annual Conference on Evaluation and Assessment in Software Engineering (EASE 2011). IET, 2011, pp. 144–153.
[21] D. A. Tomassi, “Bugs in the wild: examining the effectiveness of static analyzers at finding real-world bugs,” in Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2018, pp. 980–982.
[22] D. Steidl, F. Deissenboeck, M. Poehlmann, R. Heinke, and B. Uhink-Mergenthaler, “Continuous software quality control in practice,” in 2014 IEEE International Conference on Software Maintenance and Evolution. IEEE, 2014, pp. 561–564.
[23] S. Wagner, J. Jürjens, C. Koller, and P. Trischberger, “Comparing bug finding tools with reviews and tests,” in IFIP International Conference on Testing of Communicating Systems. Springer, 2005, pp. 40–55.
[24] M. Fowler, Refactoring: improving the design of existing code. Addison-Wesley Professional, 2018.
[25] M. Fokaefs, N. Tsantalis, and A. Chatzigeorgiou, “Jdeodorant: Identification and removal of feature envy bad smells,” in 2007 IEEE International Conference on Software Maintenance. IEEE, 2007, pp. 519–520.
[26] “JDeodorant website,” 2020. [Online]. Available: https://github.com/tsantalis/JDeodorant
[27] L. Lavazza, D. Tosi, and S. Morasca, “An empirical study on the persistence of spotbugs issues in open-source software evolution,” in 13th International Conference on the Quality of Information and Communications Technology – QUATIC, 2020.
[28] N. Tsantalis, T. Chaikalis, and A. Chatzigeorgiou, “Ten years of jdeodorant: Lessons learned from the hunt for smells,” in 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE, 2018, pp. 4–14.

International Journal on Advances in Software, vol 14 no 1 & 2, year 2021, http://www.iariajournals.org/software/

Simulation of Push- and Pull-Processes in Logistics

Usage, Limitations, and Result Presentation of Clock Pulse and Event Triggered Models

Carlo Simon, Stefan Haag, and Lara Zakfeld

Hochschule Worms
Erenburgerstr. 19, 67549 Worms, Germany
Email: {simon,haag,zakfeld}@hs-worms.de

***Abstract*—The change from a push to a pull strategy constitutes a considerable intervention in the operational logistics and has effects on procurement, the design of processes and the evaluated, internal inventories. Simulation models that anticipate the consequences of such a change need to fulfill two main objectives: 1.) They have to represent the system behavior over time, but in time lapse instead of real time. To achieve this, two approaches can be used, the first of which is a clock pulse simulation where the system's state is calculated for every discrete time step, e. g. each second. The second one is an event triggered simulation where only those points in time are computed at which an actual change occurs. Both methods have benefits and drawbacks as is elaborated. 2.) They need to take into account actual production data and, in order to use this data, implement decision rules. These aspects can be realized with Petri nets that the authors use as preferred modeling language for decades, because Petri net models are illustrative and can be executed or simulated, respectively. A novel, web-based Petri net modeling and simulation environment - the Process-Simulation.Center - allows for training modelers and testing different procedures and techniques of model generation. Using a teaching laboratory for logistics as a sample application, clock pulse and event triggered simulation models are demonstrated, as well as how they can be developed, how they have to be interpreted, and which possible obstacles have to be considered. Concretely, the consequences of switching logistics processes from push to pull principles are regarded concerning the storage costs. This paper demonstrates the interplay between new modeling approaches with the aid of Petri nets and the novel tool without which these models would not have been possible.**

***Keywords*—Conceptual modeling of timed dynamic systems; Clock Pulse Simulation; Event Triggered Simulation; Petri nets; Logistics.**

I. Introduction

This paper is a revised and extended version of a contribution to *SIMUL 2020: The Twelfth International Conference on Advances in System Simulation* [1].

Change is an integral part in every organizational context. One option to examine varying approaches and their differing results is simulation. Depending on the circumstances there may exist various foci. Hence, simulation models and their implementation differ in accordance to these goals.

For example, reducing costs while at the same time increasing the production's flexibility is a combined goal for manufacturers. Beside an investment in better and faster machines, rethinking production strategies and processes is also feasible.

Changing production from push to pull is one (possibly cheap) option. The advantages have been demonstrated in many production lines. Nonetheless, push strategies are still widely in practical use. What is the reason for this? The authors assume that producers are uncertain about the consequences of such changes. In this case, conceptual and simulatable models of the current and intended production lines could objectify decisions on the reorganization of production. Depending on the objective, different information, models and simulations are needed. How to choose between two modeling methods is subject of Section VIII where usage and some limitations of these methods are also discussed.

Petri nets are used to present two approaches for implementing corresponding models. As examined in Section VII, one is event triggered simulation that eventually leads to results where the final outcome is more important than the path to this outcome [1]. Demonstrated in Section VI is the other one, namely clock pulse simulation whose results allow for closer examination of the actual process execution [2]. Section V outlines basic considerations for IT-based simulations.

The setting for these models is a training laboratory for logistics students at the Worms University of Applied Sciences. The so-called *Box Game*, which is introduced in Section IV, was developed to impart knowledge and practical experience in highly relevant logistics processes while at the same time maintaining a relatively simple structure. It is ideally suited to explore different possibilities of conceptual modeling and simulation. In Section III the methodological approach used by the authors in the context of modeling and simulation with the Process-Simulation.Center (*P-S.C*) is described. A part of the directly following Section II about related work considers the reasoning as to why the tool's development cannot be separated from the modeling and simulation approach discussed.

II. Related Work

Since this paper combines conceptual modeling with Petri nets and the simulation of laboratory processes in logistics, related work for both fields is considered.

A. Push, Pull and Kanban

In a push production, every workstation produces as soon as being supplied sufficiently regardless of a given demand. This leads to a steady production and a high utilization rate.

In a pull production, the workstations only produce for a given actual demand, resulting in lower stocks and a more flexible production. Which of these paradigms is advantageous over the other depends on the circumstances. Sometimes, mixed solutions are best [3].

Kanban is a method to realize pull principles in logistics and, hence, to lower unnecessary stocks by controlling the replenishment of material to be processed. If a threshold is recognized, a kanban signal initiates a pull request that includes information on the batch size, leading to stable production sizes. Dependening on the used variant, the replenishment can be controlled by cards, empty containers, via e-kanban or by use of a supermarket system [4], [5].

As the pull requests establish an order chain starting at the dispatch warehouse, information in kanban systems flows upstream, while material flows downstream. Different types of kanban may be used to account for the type of material, set up times for production or relevant internal factors [6].

B. Petri Nets

Petri nets can be used to study the performance of push and pull approaches. Practically highly relevant constraints like manufacturing and setup times, vehicle routing or concurrent processing become operational and, thus, flexible manufacturing systems can be examined [3]. Large and interlocked systems can be modeled by expanding on local components; applying different Petri net specifications suited for respective tasks is beneficiary [7].

Originally, Petri nets are defined as Place/Transition nets (P/T) with anonymous tokens indicating a system's state [8]. Diverse concepts for representing high level information in Petri nets exist, the most widely known being the following:

Predicate/Transition nets (Pr/T) omit anonymous tokens for ones carrying data that can be processed and altered by use of functions encoded on transitions. When firing, these functions accept data from tokens on the preset. Functions return their results by putting appropriate tokens on the postset. The places serve as predicates according to which transitions may fire. Thus, it is possible to model interactions of tokens according to real-world influences or with each other [9].

Colored Petri Nets (CPN) integrate colors into Petri nets such that tokens, places and transitions have an assigned identity, their color. When determining if a transition is enabled, the adjacent places and their tokens are examined by color separately. This allows for more compact net representations under certain circumstances [10], [11].

C. Time Concepts in Petri Nets

Since time is an important dimension for the modeling of processes and dynamic systems in general, there exist numerous approaches for handling time aspects in Petri nets. They differ concerning their expressiveness (discrete or continuous time) and which Petri net elements are used to express time constraints (places, transitions, arcs, or tokens).

One possible implementation puts one or two time values on transitions, the lower being a delay up to which the transition is not enabled while the higher presents the latest possible moment of firing. This may lead either to a forced firing, the reset of a clock (where the lower value describes a kind of preparation time and the higher one an expiration time from when a new preparation needs to be conducted) or even to a dead net. Variations include time consuming firing [12] as well as firing without time consumption [13].

Another obvious possibility is to assign time values to the places, again representing lower and upper bounds. These bounds represent the availability of tokens, either as delay until a token becomes available [14] or as time windows in which they are available [15], [16].

Yet another possible implementation is to define the permeability of arcs relative to the moment an adjacent place was marked or an adjacent transition was enabled. The cited concepts are equivalent [17]. However, they have the disadvantage that the state of such nets does not only rely on the respective markings, but also on some kind of timer clocks.

D. Timestamps and Petri Nets

Timestamps are a means to encode time data in the marking.

Timestamp Nets introduce tokens with (only) timestamps as information designating the moment the corresponding token was placed. Transitions may fire in time windows as given by two non-negative values on the transitions' incoming arcs [18]. The permeability of arcs depends on these timestamps [19].

Extended Timestamp Nets integrate the concepts of Pr/T and Timestamp nets such that tokens carry timestamps and any further information [20].

In (extended) timestamp nets, a separate clock becomes unnecessary. Durations or important points in time can be processed by means of simple algebraic operations.

A potential drawback of all described time-dependent Petri net concepts could be a more complex situation of enablement where a transition bearing a token can be enabled, but also (depending on the arcs' permeability) not yet or no longer. This may possibly lead to a user's perception of a quasi-existence of different markings at the same time.

Some of the approaches presented in Sections II-C and II-D may be transformed into each other quite effortlessly [21], [22]. All of the presented Petri net formalisms, however, use artificial, abstract time units. To model and simulate real-world applications, real time values should be used. To this end, actual date and time data types seem beneficial to be included as possible information on tokens. Such nets can be regarded as natural extension of Pr/T nets, allowing for both time calculations inside the model and controlling the execution.

E. Further Modeling Approaches

For reference, there are other modeling methods that were developed to combine time and process structures.

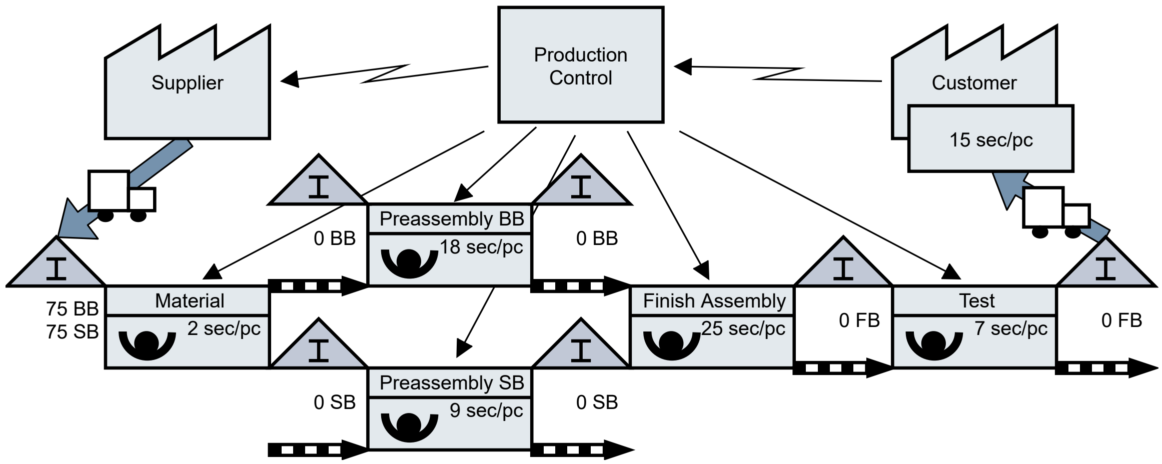


Fig. 1. Value stream diagram of the *Box Game*.

Value Stream Diagrams (VSD) establish models of flows of information and material in order to evaluate value streams. To optimize the value streams, wait times (beside other factors) need to be minimized. The value stream method exposes such wait times. This concept became widely known due to Toyotas Production System from 1930 and its advancements by Japanese engineers Taiichi Ohno and Eiji Toyoda, but dates back as far as 1914 when graphical nets were used to examine routings and other flows to help *Installing Efficiency Methods* in a manufacturing company [23], [24].
Figure 1 shows the scenario explained in Section IV as a VSD. It gives a suitable overview of the value stream from customer to supplier, yet cannot be simulated.

Business Process Model and Notation (BPMN) is a notation and representation language for modeling business processes. It is extensively used due to the relative ease of both creating and understanding models. Using BPMN, it is possible to create both high-level models of companies and low-level models of single processes in a graphical approach similar to flowcharts [25], [26].
Although there are similarities between BPMN and Petri nets, the former lack the mathematical toolset that can be used to analyze Petri nets in form of linear algebra.

F. Process-Simulation.Center

To develop conceptual models for process simulation or execution, tools are needed beside the formal mathematical base. The *P-S.C* supports development of P/T and Pr/T nets [27]. It is possible to assign data types to places and use these in analogy to database tables. Own types for date and time are substructures for the simulation of processes in production and logistics and enhance the approaches to timed Petri nets.

In contrast to relational algebra and SQL where operations like select or project are applied to the set of all affected tuples and result in a set again, in *P-S.C* the tuples are processed serially. This is since in business and production, work items are also treated one after another.

Supporting high-level Petri nets, the P-S.C facilitates the definition of individual data objects as tokens such that data-driven process simulations can be conducted within the tool. A decision on the concrete sequence is made locally by the transitions of the net which also have the ability to aggregate over tuple tokens on a place, a functionality know from database systems.

Also, the *P-S.C* can be used to combine the process view on a system with other views. Process maps can be used to collate different processes with each other and to express the strategic value of processes as primary, supportive or managing. The organizational structure of an institution can be combined with the Petri net view on the processes by assigning its nodes to swim lanes for the responsible organizational units. Organizational charts complete the functions of the *P-S.C.*

Contrary to most other conceptual modeling tools, especially those that have been designed for Petri nets, a specification language has been developed for the *P-S.C* with which all types of models are scripted. Due to strong algorithms for automatic layout, modelers can concentrate purely on structural aspects of the domain to be expressed.

The dearth of current Petri net tools, the quaint user experience of most of the still working ones and the unique approach of using textual programming instead of drag-and-drop modeling in combination with the added functionality are the main reasons for the implementation of the *P-S.C.*

The models presented here could not have been implemented without the *P-S.C* as no tool with comparable capabilities is known to the authors. Also, the models cannot be properly examined without (at least some) insights into the tool. Thus, it is not sensible to separate the models from the tool as is explained further in Sections V, VI and VII.

III. Methodology

Researching new and comparing different modeling techniques for (high-level) Petri nets with individual tokens relies on the existence of proper modeling and simulation tools.

The *P-S.C* is in development for several years following the guidelines for design science research according to [28] whose main point is to create a feasible artifact. The following is a brief outline of the thereby conducted research:

Design as an Artifact: The *P-S.C* is a web-based specification and simulation software for processes encompassing both user defined and primitive data types, organizational structures and process maps. Process execution can be controlled by business relevant data linked via an interface. Also, sensors and actuators of a Raspberry Pi can be used in case the tool is installed on such a device.
The established models are a second artifact, as they are used in a teaching environment at university level but also for vocational training. They aid in transferring knowledge of modeling and simulation techniques.

Problem Relevance: Examining possible consequences of change is a highly relevant task for every company. The *P-S.C* provides practitioners with the means to model, simulate, and optimize processes with high-level Petri nets in a powerful and contemporary user experience.
A simulation permits the extension of real-world experiences in a learning environment. It helps to overcome typical limitations concerning time, resources, space, and people. The simulation environment, however, must be generic enough to assure the intended learning success. From a conceptual modeling perspective, it must be determined if all these aspects can be expressed and simulated due to a formal, semantic base.

Design Evaluation: The *P-S.C* has already been used by companies in logistics and trade. Students of an integrated logistics degree program developed a simulation model for the reorganization of a returns process [29]. The tool is also used for problem-based and research-oriented learning in bachelor and master degree programs [30].

Research Contribution: The *P-S.C* is a practical application on the theoretical basis of high-level Petri nets combined with views on organizational structures, process maps and data types. The tool offers a novel user experience and provides new insights in Petri net based modeling. As an abstract concept Petri nets do not force specific modeling approaches such as flow diagrams, value stream diagrams or other pictorial modeling approaches.

Research Rigor: The benefits of a simulation approach in opposite to pure visual methods is evaluated in mentioned bachelor and master courses as well as in cooperation with partner companies of integrated degree programs.

Design as a Search Process: Both presented prototype and models are the latest in a series that starts from the initial implementation of the underlying principles and ends in a productive system. Each implementation step has been evaluated and published (for instance, [31], [32], [33]).

Communication of Research: The results achieved so far are relevant for both research and practice. They are presented on pertinent conferences but also, more eidetic, for students and practitioners in advanced training programs.

IV. A Simulation Laboratory for Processes in Logistics

The so-called *Box Game* has been developed by Prof. Dr. Christian Reuter at the Worms University of Applied Sciences to teach students in logistics and is used as a sample application (cf. [1], [2]). Despite its simplicity, very different kinds of processes that also have a high impact for practice can be observed. It is, therefore, ideal for trying out different ways of conceptual modeling and simulation.

The concrete example is a simple construction process where students assemble small and large boxes, put the smaller into the larger ones, and finally check the quality. Thus, characteristics of push and pull systems are illustrated.

Training members are present during the game. Ideally, they take part in the game in order to gain first hand experience of motivation and work situations. Being engaged in the training helps the students to recognize different types of waste (so-called muda according to [24]), such as *overproduction*, *waiting*, and *motion*, but also the transformation of waste types. Finally, discussing the shared experiences is a major part of the learning success. A complete simulation run of the *Box Game* lasts approximately two to three hours.

Despite the simplicity of the used material and the low level of technical requirements, the *Box Game* is easily transferable to assembly work stations in a more generalized form and has a highly practical impact. Mechanical production, however, where schedules, shift patterns, changeover times or multiple machine set-ups are of particular importance, is not an objective of this training.

Figure 2 shows the spatial organization of the *Box Game* in the learning laboratory: five work tables are arranged in a suitable location and standard positions like interim storages are marked with adhesive tape. As can be surmised, the setting can also be build up in locations such as conference rooms, training rooms, or even canteens.

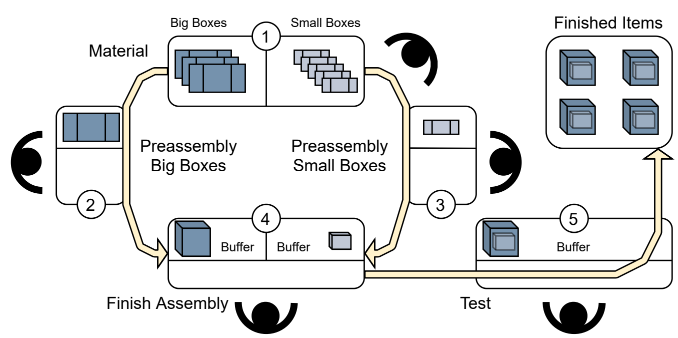


Fig. 2. Layout design of the *Box Game*.

Beside the instructor, the following can partake in the game:

5 participants who will occupy the work stations,
3 players who record the processing times,
1 observer who records inventories in the system,
1 observer who records productivity levels, and
2 further possible people who disassemble the boxes.

Though technically possible, the working stations should not be filled with more than one person. Thus, larger groups need to be split. At the five stations or (transport-wise) between them, the following activities have to be conducted:

1. **Material storage** Deliver unfolded boxes.
2. **Preassemble big boxes** Fold the big box, close its lid, and pass the box on.
3. **Preassemble small boxes** Fold the small box, close its lid, and pass the box on.
4. **Finish assembly** Open the big box, insert the small one, label the small box with a post-it as "package note", close and tape big box lid up, and pass the assembled box on.
5. **Quality test** Shake the box for an acoustic quality check, apply a red dot as "passed" to the upper left corner of the box, and place the finished box in the dispatch area.

The initial stock of the *Box Game* is 75 big and 75 small boxes. However, it is not the primary aim to produce the entire demand in the shortest possible time, but to produce them according to the customer's demand (in this case one part every 15 seconds) without much inventory and with as few employees as possible.

Recall that Figure 1 shows the value stream diagram of the *Box Game*. Although the processing times can be annotated in the diagram, it is hardly simulated due to a lacking mathematical foundation. To the authors' knowledge, even software that replicates human intuition of value stream diagrams does not exist. Nonetheless, the diagram helps to understand how the *Box Game* is played in detail.

To experience the challenges that management faces with regard to a possible strategic realignment, the box game is typically meant to be played in four rounds of 5 or 8 minutes each. During the simulation, two types of production principles with two batch sizes are examined.

Batch size 3 - push principle: The products are passed on in batches of size 3. Each process step works functionally independent from the other and the participants are rewarded for the amount of pieces they work on. Hence, it is the goal at each station to produce as much output as possible.

Batch size 3 - pull principle: Stations produce and pass on products in batches of size 3. Upstream stations have to hold their pieces and stop production until it is demanded by an internal or external customer. The capacity of a station and its buffer is limited to 3 items and items can only be replaced accordingly.

Batch size 1 - pull principle: The third round is played like the second one, but the batch size is reduced to one.

Improvement - pull principle: The last round is used to find improvements autonomously and to apply them as a team.

The advantage of this approach is that the participants gather personal experiences. This can hardly be replaced by a computer simulation. Yet, augmenting this hands-on experience by such a simulation helps to scale up both range and complexity of the considered process. This is partly due to the students' attention levels diminishing after about 5 minutes of play.

V. Basic Considerations for an IT-based Simulation

The authors' decades of experience with Petri nets and the availability of the *P-S.C* (which is freely usable for academic purposes) resulted in the decision to implement the described scenarios as Petri nets. However, this task turned out to be more challenging than assumed beforehand [34]:

1. Models and simulations of pull processes must distinguish between different customer orders. This can be expressed in high-level Petri nets with individual tokens. But although such Petri net classes have been known for many years (e. g., [9], [10]), there are no modeling patterns that can be used to build models.
   This makes testing models step by step as they are created even more important. Thus, new modeling techniques can be developed incidentally. The experience gained is therefore as much an artifact in the sense of [28] as the *P-S.C* or the models themselves.
2. Without a suitable tool for modeling and simulating high-level nets, however, this experience cannot be gained, which is a hurdle for many modelers. Almost all Petri net tools listed in [35] are either outdated, do not support time aspects or Petri nets with individual tokens, and none of them have a modern user interface. Therefore, they are all but unusable for the task at hand.
   The *P-S.C*, however, is well suited to be used for modeling the described logistics laboratory.

Section II already examined related work in order to rule out that only the authors' personal preferences justified the chosen modeling approach.

VI. Clock Pulse Simulation Models

Visualization-wise, the *P-S.C* draws nodes in a manner that allows label and token quantity to be presented inside each node. This is in contrast to the standard circles and squares (or the original lines) but allows for easier interpretation as such labels are often found in other modeling systems.

Clock pulse models evaluate every discrete time step and, thus, can be used to study the change of the systems' states. Concretely, the presented models in this section allow for the observation of fluctuating storage utilization.

Such an observation necessitates a constant flow of time. To this end, the first step in modeling the *Box Game* is the implementation of a clock that ticks every second, as depicted in Figure 3 (a). A tiny Petri net consists of a time-typed place *clock*, a corresponding transition *pulse* and two anti-parallel arcs. One arc (bearing the label *s*) removes a time token from the *clock*, while the other one adds one second to the received value and puts a token back on the *clock*. Thus, each firing increments the elapsed total time by one second.

The *P-S.C* allows for computations to be made on arcs instead of only accessing token values as is the case for the label *s*. However, when such calculations become too complex, they also become increasingly unwieldy and obstruct much space in the net's visualization.

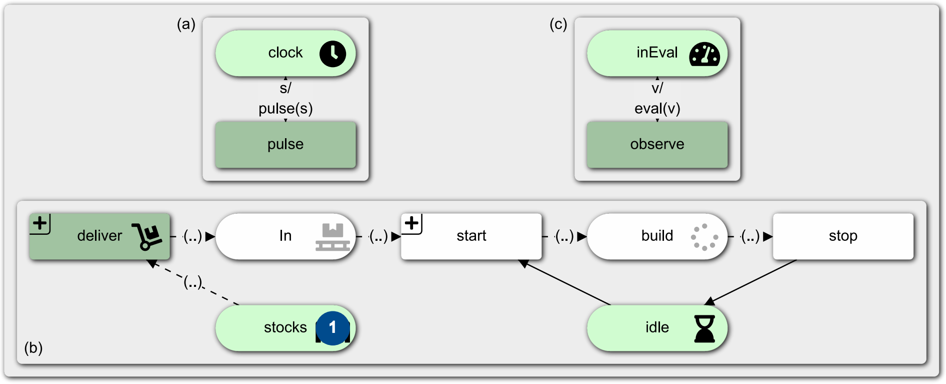


Fig. 3. Modeling concepts: (a) simple Petri net clock · (b) preassembling as instance of a single working station · (c) observer net that evaluates the upstream storage in (b).

Functions can be implemented in the net's specification and then be called from every arc - provided the passed parameters are in order. The functions' return values determine the contents of the tokens to be put on the post set's places. In case of *pulse(s)*, the function returns *s.sec+"00:01"*.

The basic model of a working station (here, folding of the big boxes as shown in Figure 3 (b)) consists of one place *in* as a buffer, one for the upstream *stocks* and one for the workplace *build* itself. These places are typed as a user-defined record set consisting of an integer as *id*, a character string as *type* descriptor and a time-value as *inStamp*, denoting the time the corresponding token was put on the place.

The transition *deliver* provides the inbound buffer *in* with feedstock from the *stocks* (and possibly other associated data) while the transition *start* connects this buffer with the workplace *build*. Both transitions carry a *select* criterion to choose the item with the minimal *id* according to the FIFO principle and put it on the post set - other queueing principles could be implemented just as well. The transition *stop* carries a *condition* to wait for the item to be finished. This *condition* is implemented as difference between the current times of the *clock* and the *inStamp* as encoded on the token. The used times are based on students' experience. Conditions and selections can be displayed in the model by clicking on the plus-symbol.

The transition *stop* is attached to an interim storage (as shown later) that serves as buffer for the succeeding working station. The (non-typed) place *idle* serves as a semaphore that prohibits *start* from firing while the workplace is busy: Only one box at a time can be processed. In the full model, it is also used as a time-typed note for the workplace.

Though single transitions can be fired by clicking on them, the *P-S.C* supports simultaneous firing of all enabled ones. This is used to synchronize all transitions with the *clock*.

An observer net, as exemplarily shown in Figure 3 (c), evaluates the costs associated with the storage. Each clock pulse, the (integer) value of the place *inEval* is increased by the amount of tokens on place *in*, thus indicating this storage's costs in this second. Again, this is implemented as a combination of a token access *v* and a function call *eval(v)*.

Combined, these three models allow for the observation of fluctuating storage levels, of bottlenecks, and of storage cost.

In the initial marking of the working station net in Figure 3 (b), there is one token each on the places *stocks* and *idle* which is indicated by their light green color. Additionally, *stocks* shows one big blue circle containing the amount of actual tokens on the place. This visual is omitted on *idle* as this semaphore may not contain more than one token (as is also the case for *clock* and *inEval*). The transition *deliver* is colored in a darker green to show it is enabled, i. e., it is ready to fire.

Figure 4 shows the remaining reachable states of the working station net. The transition *deliver* chooses the (only) token and puts it on *in*, enabling *start* as shown in Figure 4 (a). The next pulse removes the semaphore token and the one from *in*, preventing *start* from becoming enabled. This is reflected in Figure 4 (b). The marking stays stable until the *condition* on *stop* regarding the processing time evaluates to true.

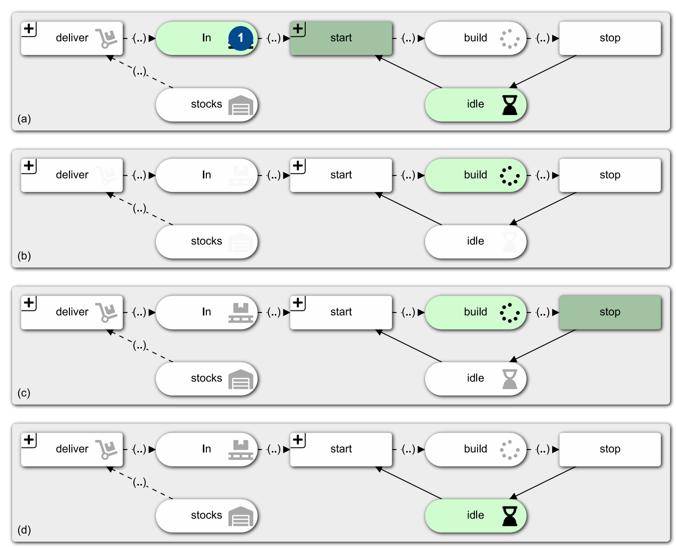


Fig. 4. Visualized states of the working station net in Figure 3 (b): (a) after clock pulse one · (b) during the building phase · (c) after clock pulse twenty · (d) after clock pulse twenty-one.

Now, the transition becomes enabled, leading to the net as shown in Figure 4 (c). Figure 4 (d) depicts the net after the item has been taken from *build* and the semaphore token has returned, rendering the working place accessible again.

TABLE I. MARKINGS OF THE MODEL AS DEPICTED IN FIGURE 3

			material			in			build		
clock	inEval	idle	id	type	inStamp	id	item	inStamp	id	type	inStamp
00:00	0	•	1	big box	00:00						
00:01	0	•				1	big box	00:01			
00:02	1								1	big box	00:02
⋮	⋮								⋮	⋮	⋮
00:20	1								1	big box	00:02
00:21	1	•									

These state changes are reflected in Table I. The first row shows the net's initial marking. As each pulse evaluates to one second, the box arrives at *build* after two seconds. There, the box is processed for 18 seconds, then leaving for the next buffer. The net reaches its final marking after 21 seconds.

All models presented here, being implemented for automated simulation, do not need any user interaction. In effect, each state has a unambiguously defined subsequent state.

A. The Push Model

By expanding on the presented working station model and using the mentioned concepts, the first iteration of the *Box Game* (the push version as presented during a simulation run in Figure 5) can be modeled. As the principle of the net can be used for the other workplaces, they can be structurally copied and then adapted with respect to the processing time.

Connected to the upstream buffer places *inBB* and *inSB* of the big and small box assembly nets, there is the already mentioned place *material* for the main warehouse. The warehouse is connected via the two delivery transitions *deliverSB* and *deliverBB*. As all these places carry item representations for tokens, they are typed with the user-defined record set *RStock* that was introduced earlier. *RStock* consists of an integer *id*, a character as *type* description and a time-value *inStamp* for the moment the token was put on the place.

Initially, *material* bears 75 tokens each for unfolded big and small boxes. The structure of this allocation can be seen in Table II. *BB* and *SB* are short for big and small box.

TABLE II. INITIAL ALLOCATION OF TOKENS ON THE PLACE *material* IN THE CLOCK PULSE PUSH MODEL AS DEPICTED IN FIGURE 5

id	type	inStamp
1	BB	00:00
⋮	⋮	⋮
75	BB	00:00
76	SB	00:00
⋮	⋮	⋮
150	SB	00:00

Both sides are structured equally, so it is sufficient to only explain one path in detail. The transition *deliverBB* carries several criteria. First, *conditions* ensure that only correct items (i. e., items designated *BB*) are chosen and that transportation has a duration of 2 seconds (i. e., *deliverBB* is only enabled if 2 seconds have elapsed since the last transport). Second, the *select* criterion chooses the item with the minimal *id*.

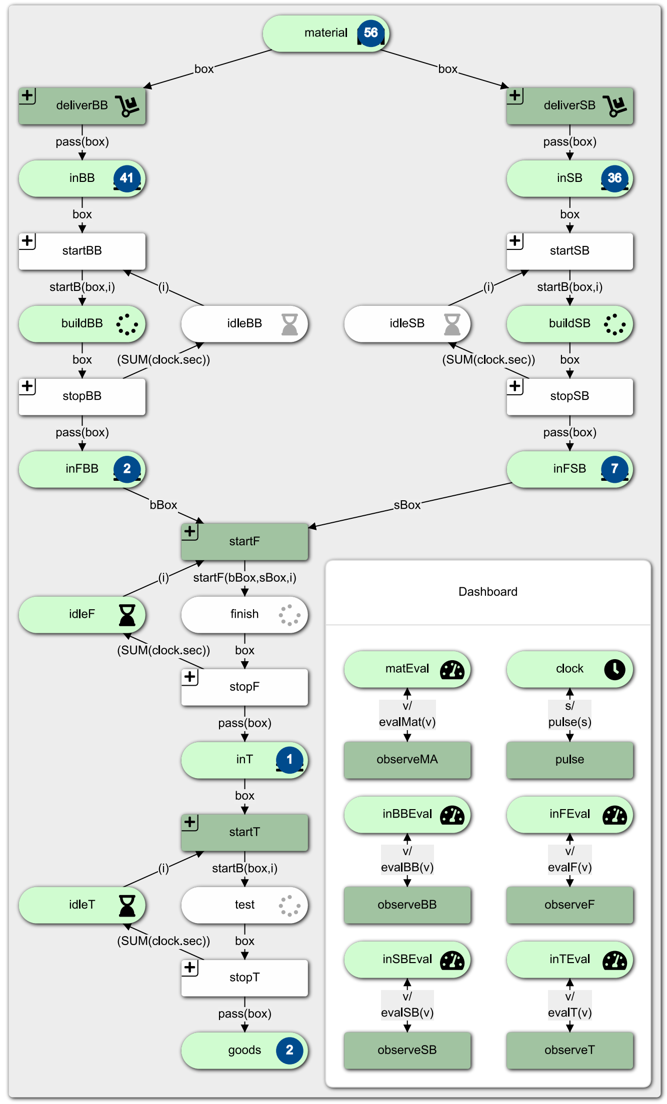


Fig. 5. Clock pulse push model after 96 clock pulses.

While the arc *box* takes one token from *material*, the function *pass(box)* sets a new timestamp (according to the elapsed time as provided by the *clock*) and puts it on *inBB*.

The transition *startBB* is enabled if there is at least one token on *inBB* and the semaphore token is available. As is the case for all semaphores, the initial token bears *"00:00"* as time value. Again, *select* criterion on *startBB* is the minimal *id*. The transition receives a *box* and the semaphore token as input data. Upon firing, the function *startB(box,i)* determines the correct putting time and places the box on *buildBB*.

Due to folding the big boxes taking 18 seconds, *stopBB* only becomes enabled after this time has elapsed.

Note the difference between the (left) big box and the (right) small box sides: Folding a small box only takes 9 seconds. Thus, the *condition* on *stopSB* adheres to this value.

Upon firing, *stopBB* puts the folded *box* on the output buffer *inFBB* that is named for its second purpose as input buffer for the following subnet to finish the assembly. Additionally, the semaphore token is put back on *idleBB*. The semaphores can be used to determine machine processing (or idle) times.

The subnet for the final assembly differs in possessing a second input buffer *inFSB* as both boxes are needed. This also causes a slightly adapted function *startF(bBox,sBox,i)* because the time-values of both incoming boxes must be considered. Structurally fully equivalent to the first two subnets is the last one for quality testing.

Like for conditions, functions and data accesses on arcs can be shown or hidden, as needed. To further enhance the visual understanding of their functionality, nodes can be provided with (animated) symbols.

Tables III to IX are meant to examine the markings' progression of the model as depicted in Figure 5.

The first row of Table III is a shorter version of Table II: Before the initial pulse (i. e., "after pulse 0") the cell *id* contains the full column *id* from Table II, the single elements being separated by the pipe symbol. The same applies to the cells *type* and *inStamp* of the first row. As we assume a transportation time of 2 seconds from *material* to the buffers, the first change occurs after pulse 2: The items *(1, BB, 00:00)* and *(76, SB, 00:00)* are consumed from *material* and put on their respective buffers. This is repeatedly shown for the first 10 pulses. The row for pulse 96 shows the tokens on *material* for the system's state in Figure 5. Pulses 150 and 151 contain the allocations just before the last consumption from the place and the final state of *material*.

As this concept also extends to all following tables, every row in them can be expanded to an own table from which the model's state after the corresponding pulse can be explored. The combination of all these tables into one huge table is the result export from the *P-S.C*. In case of the clock pulse push model, this table contains 44 columns with 1904 data sets as rows each of which corresponds to one system state. Thus, this export comprises the reachability set of the net.

On the other end of the model, the place *goods* contains the finished boxes. Initially, this place is empty. In accordance with Table IV, the first token creation takes place at pulse 53, designating the first finished product *(1, FB, 00:52)*. Thus, after 52 seconds, the first box is ready and put in the finished goods warehouse. During the next 10 pulses, no change occurs: Only after pulse 77 will the next item be ready. Pulse 96, again, shows the allocation of tokens on *goods* at the state of the model as shown in Figure 5: Two items are finished, the second one being put there after 1:17 minutes. The last token creation takes place at pulse 1903: the item *(75, FB, 31:42)* is finalized after more than half an hour. Then, the allocation does not change any more, indicating the end of the simulation run.

TABLE III. ALLOCATIONS OF TOKENS ON THE PLACE *material* IN THE CLOCK PULSE PUSH MODEL AS DEPICTED IN FIGURE 5: INITIALLY, AFTER THE FIRST 10 PULSES, AFTER PULSE 96, AND AROUND THE LAST CONSUMPTION

	material		
pulse	id	type	inStamp
0	1|...|150	BB|...|BB|SB|...|SB	00:00|...|00:00
1	1|...|150	BB|...|BB|SB|...|SB	00:00|...|00:00
2	1|...|150	BB|...|BB|SB|...|SB	00:00|...|00:00
3	2|...|75|77|...|150	BB|...|BB|SB|...|SB	00:00|...|00:00
4	2|...|75|77|...|150	BB|...|BB|SB|...|SB	00:00|...|00:00
5	3|...|75|78|...|150	BB|...|BB|SB|...|SB	00:00|...|00:00
6	3|...|75|78|...|150	BB|...|BB|SB|...|SB	00:00|...|00:00
7	4|...|75|79|...|150	BB|...|BB|SB|...|SB	00:00|...|00:00
8	4|...|75|79|...|150	BB|...|BB|SB|...|SB	00:00|...|00:00
9	5|...|75|80|...|150	BB|...|BB|SB|...|SB	00:00|...|00:00
10	5|...|75|80|...|150	BB|...|BB|SB|...|SB	00:00|...|00:00
⋮	⋮	⋮	⋮
96	48|...|75|123|...|150	BB|...|BB|SB|...|SB	00:00|...|00:00
⋮	⋮	⋮	⋮
150	75|150	BB|SB	00:00|00:00
151			

TABLE IV. ALLOCATIONS OF TOKENS ON THE PLACE *goods* IN THE CLOCK PULSE PUSH MODEL AS DEPICTED IN FIGURE 5: AT FIRST TOKEN CREATION AND THE FOLLOWING 10 PULSES, AFTER PULSE 96, AND AROUND THE LAST TOKEN CREATION

	goods		
pulse	id	type	inStamp
53	1	FB	00:52
54	1	FB	00:52
55	1	FB	00:52
56	1	FB	00:52
57	1	FB	00:52
58	1	FB	00:52
59	1	FB	00:52
60	1	FB	00:52
61	1	FB	00:52
62	1	FB	00:52
63	1	FB	00:52
⋮	⋮	⋮	⋮
96	1|2	FB|FB	00:52|01:17
⋮	⋮	⋮	⋮
1903	1|...|75	FB|...|FB	00:52|...|31:42
⋮	⋮	⋮	⋮

Between *material* and *goods*, a lot is happening that can be deducted from the different tables.

Table V shows data for the inbound buffers *inBB* and *inSB*. These places are initially empty. The first creation of tokens occurs after transportation from *material*, thus, the boxes are put with an *inStamp* of 2 seconds. Both boxes can be transferred into the preassembly and, therefore, disappear with the next pulse. After this, the buffer gets filled, because transportation takes less time than folding. Again, pulse 96 shows the allocation for the shown model in Figure 5. Due to folding of the small boxes being faster in comparison to the big boxes, the last consumption from *inSB* takes place at pulse 670, while the last one from *inBB* happens at pulse 1336.

TABLE V. ALLOCATIONS OF TOKENS ON THE PLACES *inBB* AND *inSB* IN THE CLOCK PULSE PUSH MODEL AS DEPICTED IN FIGURE 5: AT FIRST TOKEN CREATION AND THE FOLLOWING 10 PULSES, AFTER PULSE 96, AND AROUND THE LAST CONSUMPTION FROM THE PLACES

	inBB			inSB		
pulse	**id**	**type**	**inStamp**	**id**	**type**	**inStamp**
3	1	BB	00:02	76	SB	00:02
4						
5	2	BB	00:04	77	SB	00:04
6	2	BB	00:04	77	SB	00:04
7	2|3	BB|BB	00:04|00:06	77|78	SB|SB	00:04|00:06
8	2|3	BB|BB	00:04|00:06	77|78	SB|SB	00:04|00:06
9	2|3|4	BB|BB|BB	00:04|00:06|00:08	77|78|79	SB|SB|SB	00:04|00:06|00:08
10	2|3|4	BB|BB|BB	00:04|00:06|00:08	77|78|79	SB|SB|SB	00:04|00:06|00:08
11	2|3|4|5	BB|BB|BB|BB	00:04|00:06|00:08|00:10	77|78|79|80	SB|SB|SB|SB	00:04|00:06|00:08|00:10
12	2|3|4|5	BB|BB|BB|BB	00:04|00:06|00:08|00:10	77|78|79|80	SB|SB|SB|SB	00:04|00:06|00:08|00:10
13	2|...|6	BB|...|BB	00:04|...|00:12	77|...|81	SB|...|SB	00:04|...|00:12
⋮	⋮	⋮	⋮	⋮	⋮	⋮
96	7|...|47	BB|...|BB	00:14|...|01:34	87|...|122	SB|...|SB	00:24|...|01:34
⋮	⋮	⋮	⋮	⋮	⋮	⋮
669	38|...|75	BB|...|BB	01:16|...|02:30	150	SB	02:30
670	39|...|75	BB|...|BB	01:18|...|02:30			
⋮	⋮	⋮	⋮			
1335	75	BB	02:30			
1336						

TABLE VI. ALLOCATIONS OF TOKENS ON THE PLACES *buildBB* AND *buildSB* IN THE CLOCK PULSE PUSH MODEL AS DEPICTED IN FIGURE 5: AT FIRST TOKEN CREATION AND THE FOLLOWING 10 PULSES, AFTER PULSE 96, AND AROUND THE LAST CONSUMPTION FROM THE PLACES

	buildBB			buildSB		
pulse	**id**	**type**	**inStamp**	**id**	**type**	**inStamp**
4	1	BB	00:02	76	SB	00:02
5	1	BB	00:02	76	SB	00:02
6	1	BB	00:02	76	SB	00:02
7	1	BB	00:02	76	SB	00:02
8	1	BB	00:02	76	SB	00:02
9	1	BB	00:02	76	SB	00:02
10	1	BB	00:02	76	SB	00:02
11	1	BB	00:02	76	SB	00:02
12	1	BB	00:02			
13	1	BB	00:02	77	SB	00:11
14	1	BB	00:02	77	SB	00:11
⋮	⋮	⋮	⋮	⋮	⋮	⋮
96	6	BB	01:32	86	SB	01:32
⋮	⋮	⋮	⋮	⋮	⋮	⋮
677	38	BB	11:08	150	SB	11:08
678	38	BB	11:08			
⋮	⋮	⋮	⋮			
1352	75	BB	22:14			
1353						

Afterwards, the boxes are taken from the buffers directly into the first production steps. Note the assumption that this handling step consumes no time, i. e., the placing of a box into a buffer allows for immediate start of the production step (given the working place is available). This is due to the *Box Game*' layout: tables are in close vicinity to each other and buffers are directly connected to their working stations.

Handling times could be implemented like it is the case for the initial transportation between *material* and the first buffers.

As shown in Table VI, although being in pulse 4, *inStamp* reads *00:02* for the first preassembly. Folding a small box takes 9 seconds. After pulse 11 this time is reached and the folded small box can be transferred to the next buffer *inFSB*. Again, note the *inStamp* of the next small box being *00:11* due to the absence of handling times. The big box is not folded until pulse 20. After pulse 677, the last consumption from *buildSB* takes place, while pulse 1352 marks the last consumption from *buildBB*. Then, both places remain empty.

Table VII (a) shows the buffers behind *buildBB* and *buildSB* where the folded boxes are temporarily stored. As before, handling or transportation times are not considered, thus the *inStamp* of small box *76* reads *00:11*. The first folded big box *1* arrives at the buffer only at pulse 21. In the next pulse, the first available tokens get consumed, effectively transferring the boxes to the final assembly. Once more, pulse 96 depicts the allocation of tokens in the model's state from Figure 5 while the last two rows show the consumption of the last tokens. The small box *150* shows some 11 minutes more waiting time compared to the big box *75*.

Starting with pulse 22, a big and a small box each are transferred from *inFBB* and *inFSB* to the final assembly. Here, the small box and a note are put into the larger. The construct gets a seal and is passed on. Finishing the box takes time up to pulse 45, thus being not reflected in Table VII (b), although the the principle stays the same. At pulse 96, *finish* is not marked because (as elaborated shortly) a finished box is in transit to the next buffer and the two following boxes are in transit to the final assembly. Pulse 1895 marks the consumption of the last finished box.

TABLE VII. ALLOCATIONS OF TOKENS ON THE PLACES (a) *inFBB*, *inFSB* AND (b) *finish* IN THE CLOCK PULSE PUSH MODEL AS DEPICTED IN FIGURE 5: AT FIRST TOKEN CREATION AND THE FOLLOWING 10 PULSES, AFTER PULSE 96, AND AROUND THE LAST CONSUMPTION FROM THE PLACES

(a)	inFBB			inFSB		
pulse	**id**	**type**	**inStamp**	**id**	**type**	**inStamp**
12				76	SB	00:11
13				76	SB	00:11
14				76	SB	00:11
15				76	SB	00:11
16				76	SB	00:11
17				76	SB	00:11
18				76	SB	00:11
19				76	SB	00:11
20				76	SB	00:11
21	1	BB	00:20	76|77	SB|SB	00:11|00:20
22				77	SB	00:20
⋮	⋮	⋮	⋮	⋮	⋮	⋮
96	4|5	BB|BB	01:14|01:32	79|...|85	SB|...|SB	00:38|...|01:32
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1871	75	BB	22:32	150	SB	11:17
1872						

(b)	finish		
pulse	**id**	**type**	**inStamp**
22	1	FB	00:20
23	1	FB	00:20
24	1	FB	00:20
25	1	FB	00:20
26	1	FB	00:20
27	1	FB	00:20
28	1	FB	00:20
29	1	FB	00:20
30	1	FB	00:20
31	1	FB	00:20
32	1	FB	00:20
⋮	⋮	⋮	⋮
96			
⋮	⋮	⋮	⋮
1895	75	FB	31:10
1896			

TABLE VIII. ALLOCATIONS OF TOKENS ON THE PLACES (a) *inT* AND (b) *test* IN THE CLOCK PULSE PUSH MODEL AS DEPICTED IN FIGURE 5: AT FIRST TOKEN CREATION AND THE FOLLOWING 10 PULSES, AFTER PULSE 96, AND AROUND THE LAST CONSUMPTION FROM THE PLACES

(a)	inT		
pulse	**id**	**type**	**inStamp**
46	1	FB	00:45
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
⋮	⋮	⋮	⋮
96	3	FB	01:35
⋮	⋮	⋮	⋮
1896	75	FB	31:35
1897			

(b)	test		
pulse	**id**	**type**	**inStamp**
47	1	FB	00:45
48	1	FB	00:45
49	1	FB	00:45
50	1	FB	00:45
51	1	FB	00:45
52	1	FB	00:45
53			
54			
55			
56			
57			
⋮	⋮	⋮	⋮
96			
⋮	⋮	⋮	⋮
1902	75	FB	31:35
1903			

TABLE IX. ALLOCATIONS OF TOKENS ON THE PLACES (a) *matEval*, *inBBEval*, *inSBEval*, *inFEval*, AND (b) *inTEval* INDICATING THE ACCUMULATED STORAGE COSTS IN THE CLOCK PULSE PUSH MODEL AS DEPICTED IN FIGURE 5: INITIALLY, AFTER THE FIRST 10 PULSES, AROUND PULSE 96, AND AT THE LAST TOKEN CREATING PULSE

pulse	matEval	inBBEval	inSBEval	inFEval	inTEval
0	0	0	0	0	0
1	150	0	0	0	0
2	300	0	0	0	0
3	450	0	0	0	0
4	598	1	1	0	0
5	746	1	1	0	0
6	892	2	2	0	0
7	1038	3	3	0	0
8	1182	5	5	0	0
9	1326	7	7	0	0
10	1468	10	10	0	0
⋮	⋮	⋮	⋮	⋮	⋮
95	9926	1886	1656	327	2
96	9982	1927	1692	336	2
97	10038	1968	1728	345	3
⋮	⋮	⋮	⋮	⋮	⋮
1903	11550	44475	19500	64650	75

While Table VIII (a) examines the behavior for the last buffer *inT* just like it is the case for the other corresponding buffers, Table VIII (b) does so for the last working station *test* in accordance to the others.

The dashboard view shows the already explained *clock*: Initialized with *"00:00"*, each tick adds one second to the token. All evaluation places $*Eval$ are initialized with *0*: no costs have accumulated, yet. For each pulse, these values are accessed by the arcs designated *v*. Then, the amount of tokens on the corresponding buffer gets added to the value (by means of the respective $eval*(v)$ function) which gets returned to the evaluation places. Thus, every item on any of the buffers creates one unit of costs every second on these buffers.

Table IX gives an overview of the storage costs' development. Although it would be possible to also include the storage costs of the finished *goods* warehouse, this is omitted because the goal of the models is to examine the difference of internal storage costs when switching from push to pull production. As throughput is limited by the final assembly step, taking 25 seconds, the inflow on *goods* remains the same regardless of the implemented principle.

B. The Pull Model

The pull version of the *Box Game* as depicted in Figure 6 is based on the previous push version. Some additional elements facilitate the implementation of pull principles.

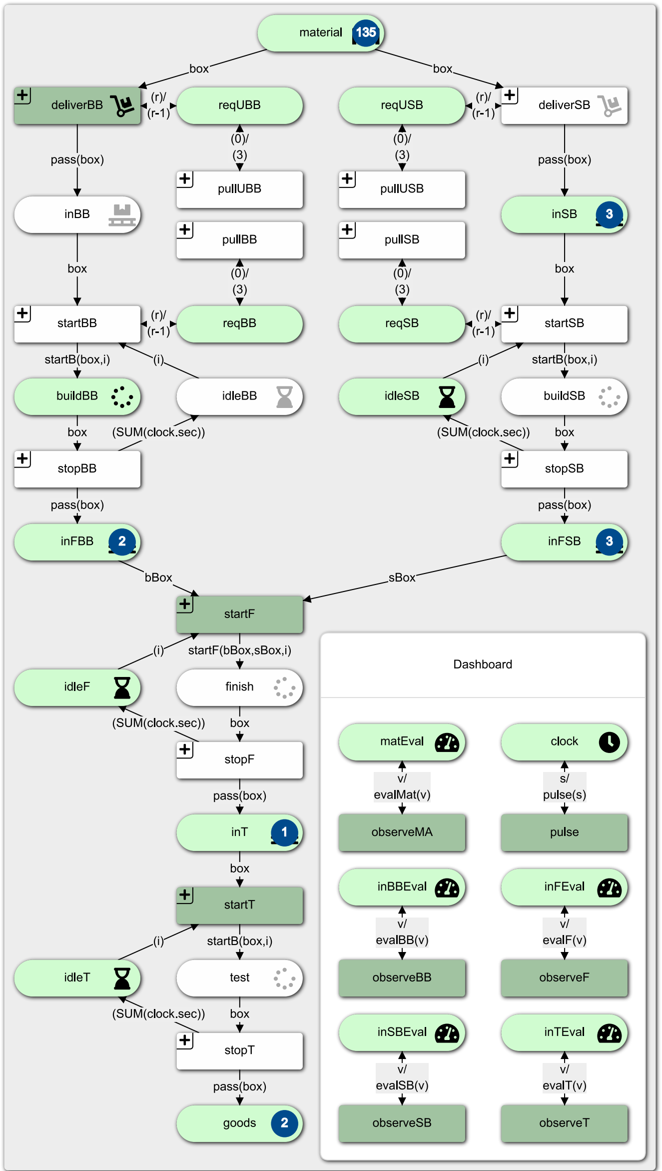


Fig. 6. Clock pulse pull model after 96 pulses.

New places *req∗* for requesting boxes are attached to the transition for delivery and starting the preassembly. These places are integer-typed and initialized with a single token bearing *0*. The transitions *pull∗* increases this value to 3, effectively implementing a batch size. This can be modeled with a functionality of the *P-S.C* not normally seen in Petri nets: Observance of other net elements. Exemplary, *pullUBB* carries a *condition='COUNT(inBB)=0'*: this transition observes the state of a place not directly connected. If *inBB* is empty, this triggers a kanban request. The displayed buffer *inBB* is empty, thus having enabled *pullUBB*. This led to the token on *reqUBB* containing *3*. Only then *deliverBB* is enabled as a *condition* asks for the request to be larger than *0*. The same principle triggers the start of the preassembly: *pullBB* observes the state of *inFBB*. Correspondingly, this is also executed on the small boxes' side.

While such an observance capability can be reproduced by more traditional Petri nets, the implemented approach may ease readability of the net as fewer arcs are needed. It is up to the modeler which visualization they deem more important.

Consequently, this pull request mechanism could also be implemented for the remainder of the net. However, these requests are omitted for two reasons:

1. The visualization would become more convoluted.

2. They are not necessary for this model. The final assembly *finish* takes the longest time of all working stations, thus representing the bottleneck. Before this place (including the associated buffers *inFBB* and *inFSB*), the kanban system is implemented. Behind this place, no congestion can occur. As only storage costs are examined, and the storage costs are the same for all buffers, total costs would not change when implementing a full kanban chain.

The remaining net is unchanged to the push version, thus, only the main difference on the resulting reachability set are examined: Indications for storage costs. As such, Table X shows the data for the pull model like Table IX does for the push model. While *material* exposes higher costs, the internal storage costs for the buffer prove smaller than in the push model. This is more deeply explored in Section VI-C.

TABLE X. ALLOCATIONS OF TOKENS ON THE PLACES (a) *matEval*, *inBBEval*, *inSBEval*, *inFEval*, AND (b) *inTEval* INDICATING THE ACCUMULATED STORAGE COSTS IN THE CLOCK PULSE PULL MODEL AS DEPICTED IN FIGURE 6: INITIALLY, AFTER THE FIRST 10 PULSES, AROUND PULSE 96, AND AT THE LAST TOKEN CREATING PULSE

pulse	matEval	inBBEval	inSBEval	inFEval	inTEval
0	0	0	0	0	0
1	150	0	0	0	0
2	300	0	0	0	0
3	450	0	0	0	0
4	598	1	1	0	0
5	746	1	1	0	0
6	892	2	2	0	0
7	1038	3	3	0	0
8	1182	5	5	0	0
9	1326	7	7	0	0
10	1470	9	9	0	0
⋮	⋮	⋮	⋮	⋮	⋮
95	13338	144	196	157	2
96	13473	144	199	162	2
97	13608	144	202	167	3
⋮	⋮	⋮	⋮	⋮	⋮
1903	127068	4133	4584	4926	75

C. Simulation Results

Comparison of the two models' simulation runs show neither a change in total processing time nor one in idle times for the different workplaces, which is as expected. Differences on utilization of storage places become visible, though. Figures 5 and 6 depict both clock pulse models after pulse 96. Thus, they are in a comparable state but clearly show distinct distributions on the earlier buffers. This is even more evident when comparing the last rows of Tables IX and X.

The first model using push principles clearly unveils the drawback of its approach: large interim storage and, as a result, high inventory costs. Figure 7 (upper) depicts the (box-wise split) inventory on the material storage (*rawBB* and *rawSB*), the building buffers (*inBB*, *inSB*, *inFBB*, and *inFSB*) and the finished *goods* storage during the push simulation.

Figure 7 (lower) shows the pull simulation run's results. The interim storage places are much less utilized as only those items are put into the assembly lines that are demanded by downstream stations. The decrease in raw material stocks is very uniform. Both is as expected, as it corresponds to the main goal behind just-in-time production schedules.

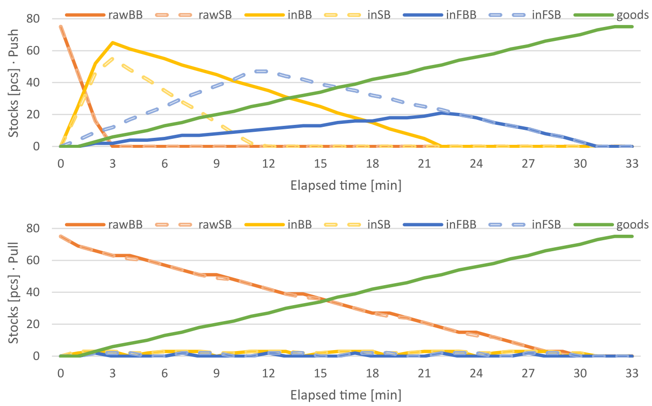


Fig. 7. Stocks in the clock pulse models for (upper) push and (lower) pull.

Figure 8 (upper) presents the accumulated costs of interim storage in the push model, while Figure 8 (lower) depicts the same for the pull model. Also shown are the total costs of all interim storage places. Be mindful of the scale difference of the y-axis: The accumulated interim storage costs differ by an order of magnitude when using pull instead of push principles!

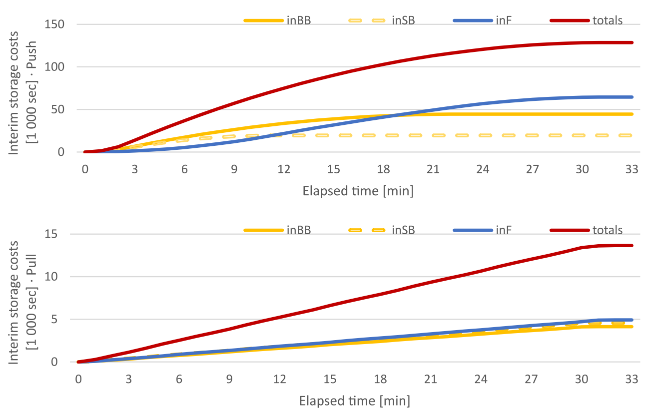


Fig. 8. Accumulated inventory costs per interim storage and totals in the clock pulse models for (upper) push and (lower) pull.

The finished *goods* storage is omitted as throughput is the same in both models, leading to the same costs on this storage. The *material* warehouse, otherwise, is put aside because the difference between the accumulated totals equals to the possible savings when externalizing this storage.

By implementing pull principles, the stock costs on the interim storage places plummet. Thus, the advantages of a just-in-time production and a smaller main storage become obvious. The pull principle allows for exactly this.

What is not accounted for in these models are for example costs of transportation, as smaller batch sizes usually correspond to higher transportation costs, leading to familiar knowledge: decreasing one muda typically increases other muda. Hence, batch size 1, which is optimal for the interim storage cost, is not necessarily the globally optimal solution.

VII. Event Triggered Simulation Models

The second modeling approach examined aims at developing an event triggered simulation model, again with the main goal to determine the total costs of all involved stocks and to explain the applied modeling technique. Thus, the setting remains the same but the modeling principles change. While the clock pulse models allow for observation of storage utilization in real-time, event triggered models aim at computing simulation results as fast as possible.

Hence, the now presented Petri net models mainly consist of places to examine the stocks. The remainder of the *Box Game*'s functionality is implemented as transitions and arcs. Specifically, there are no places for the working stations.

Since an ideal production flow is one with minimal stocks and short throughput times, the individual boxes are passed from one production step directly to the following, effectively establishing a batch size of 1. This is in accordance with the handling times of the clock pulse models that were set to *0* with the exception of the initial dispatch. Other batch sizes are implemented accordingly [5].

A. The Push Model

Figure 9 shows the push version of the event triggered model. The *timer* establishes one possibility to track elapsed times (another one is presented in the following Section VII-B). The initial allocation with four tokens can be seen in Table XI. The used record set *RTimer* stores the *type*, i. e., the associated storage place, and the corresponding *elapsed* time value of last access on this place. The tokens are initialized with *"00:00"* the buffers have not been in use, yet.

TABLE XI. INITIAL ALLOCATION OF TOKENS ON THE PLACE *timer* IN THE EVENT TRIGGERED PUSH MODEL AS DEPICTED IN FIGURE 9

type	elapsed
BB	00:00
SB	00:00
FB	00:00
QA	00:00

The place *material* is typed with a record set *RMat* that consists of the item *type*, the *elapsed* time, and two attributes *boxID* and *count*. While *boxID* determines the next box's id value, *count* shows the amount of the remaining stock. The attribute *type* is self-explanatory and *elapsed* tracks the time of an item's state change.

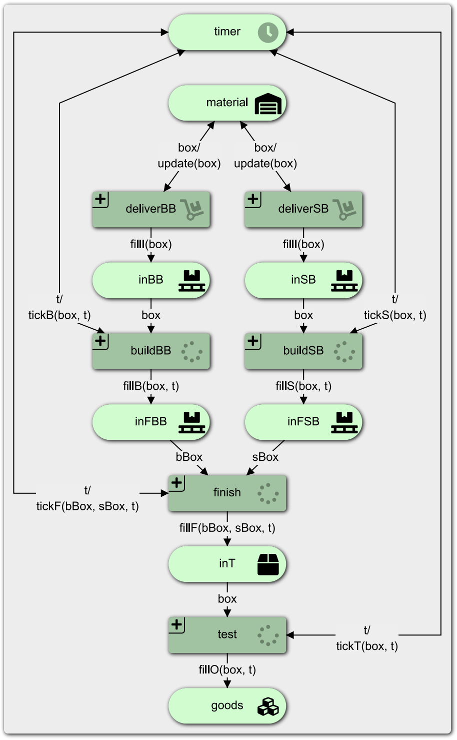


Fig. 9. Event triggered push model after 4 ticks, the first completed box having just arrived on the finished *goods* warehouse.

The initial allocation for *material* is shown in Table XII: There are 75 big and small boxes, respectively. Both (big and small box) tokens that are the next to be created from this information will have the *id 1*.

TABLE XII. INITIAL ALLOCATION OF TOKENS ON THE PLACE *material* IN THE EVENT TRIGGERED PUSH MODEL AS DEPICTED IN FIGURE 9

boxID	type	elapsed	count
1	BB	00:00	75
1	SB	00:00	75

Left and right sides of the preassembly work correspondingly, thus, an explanation of the big boxes' side will suffice. Two *conditions* on *deliverBB* ensure usage of the *BB*-typed token from *material* while there are items left as indicated by the token's *count*. The information gets adapted by the function *update(box)*: *boxID* is incremented, *count* is decremented, and *elapsed* receives a time stamp for the last access. Then, the token is put back on *material* and a newly created *RBox*-typed token is put on *inBB*.

With creation of a box token, *boxID* actually becomes the box's *id* while *count* as attribute on a box becomes superfluous. Thus, *RBox* is adapted as a triplet *id*, *type*, and *elapsed*. Using the *id*, the items can be tracked in this model's tables.

The progression of token allocations on *timer*, *material* warehouse, and the buffers *inBB* and *inSB* can excerpted be seen in Tables XIII, XIV, and XV, respectively.

TABLE XIII. ALLOCATIONS OF TOKENS ON THE PLACE *timer* IN THE EVENT TRIGGERED PUSH MODEL AS DEPICTED IN FIGURE 6 FOR THE FIRST AND LAST FIVE SYSTEM STATES EACH

state	type	elapsed
0	BB|SB|FB|QA	00:00|00:00|00:00|00:00
1	BB|SB|FB|QA	00:00|00:00|00:00|00:00
2	BB|SB|FB|QA	00:20|00:11|00:00|00:00
3	BB|SB|FB|QA	00:38|00:20|00:45|00:00
4	BB|SB|FB|QA	00:56|00:29|01:10|00:52
⋮	⋮	⋮
74	BB|SB|FB|QA	21:56|10:59|30:20|30:02
75	BB|SB|FB|QA	22:14|11:08|30:45|30:27
76	BB|SB|FB|QA	22:32|11:17|31:10|30:52
77	BB|SB|FB|QA	22:32|11:17|31:35|31:17
78	BB|SB|FB|QA	22:32|11:17|31:35|31:42

TABLE XIV. ALLOCATIONS OF TOKENS ON THE PLACE *material* IN THE EVENT TRIGGERED PUSH MODEL AS DEPICTED IN FIGURE 6 FOR THE FIRST AND LAST FIVE SYSTEM STATES EACH

state	boxID	type	elapsed	count
0	1|1	BB|SB	00:00|00:00	75|75
1	2|2	BB|SB	00:00|00:00	74|74
2	3|3	BB|SB	00:00|00:00	73|73
3	4|4	BB|SB	00:00|00:00	72|72
4	5|5	BB|SB	00:00|00:00	71|71
⋮	⋮	⋮	⋮	⋮
74	75|75	BB|SB	02:28|02:28	1|1
75	76|76	BB|SB	02:30|02:30	0|0
76	76|76	BB|SB	02:30|02:30	0|0
77	76|76	BB|SB	02:30|02:30	0|0
78	76|76	BB|SB	02:30|02:30	0|0

TABLE XV. ALLOCATIONS OF TOKENS ON THE PLACES *inBB* AND *inSB* IN THE EVENT TRIGGERED PUSH MODEL AS DEPICTED IN FIGURE 6 FOR THE FIRST AND LAST FIVE SYSTEM STATES EACH

	inBB			inSB		
state	id	type	elapsed	id	type	elapsed
0						
1	1	BB	00:02	1	SB	00:02
2	2	BB	00:04	2	SB	00:04
3	3	BB	00:06	3	SB	00:06
4	4	BB	00:08	4	SB	00:08
⋮	⋮	⋮	⋮	⋮	⋮	⋮
74	74	BB	02:28	74	SB	02:28
75	75	BB	02:30	75	SB	02:30
76						
77						
78						

After the token has arrived on *inBB*, *buildBB* is enabled. It receives tokens from *inBB* and *timer* and transmits tokens to *inFBB* and back to *timer*. The token from *timer* is updated by the function *tick(box,t)* such that *t.elapsed=max(box.elapsed,t.elapsed)+"00:18"*: To the current time, as computed by the maximum function, 18 seconds are added for the preassembly. Thus, the *timer* token always carries the time of last access on. In the system's base state, this token is initialized with *"00:00"* as the workplaces have not been accessed, yet. However, because the delivery takes 2 seconds, the folding must not start earlier. Again, the further transportation steps are assumed to consume no time.

Using a generic function with a fitting parameter for all arcs instead of a unique one for each would have led to larger arc labels, convoluting the visualization. On the other hand, the processing times would have been visible directly.

The function *fillB(box,t)* adapts the principle behind *tickB(box,t)* to a box item where the attribute *elapsed* stores the time of last change. Allocations of tokens on the two buffers *inFBB* and *inFSB* just before the final assembly can be examined in Table XVI. The first folded big box arrives after 20 seconds. At this time, the second folded small box arrives in the other buffer, already. Du to the final assembly requiring both box types, the buffer get cleared simultaneously, though the small boxes show much longer waiting times.

TABLE XVI. ALLOCATIONS OF TOKENS ON THE PLACES *inFBB* AND *inFSB* IN THE EVENT TRIGGERED PUSH MODEL AS DEPICTED IN FIGURE 6 FOR THE FIRST AND LAST FIVE SYSTEM STATES EACH

	inFBB			inFSB		
state	id	type	elapsed	id	type	elapsed
0						
1						
2	1	BB	00:20	1	SB	00:11
3	2	BB	00:38	2	SB	00:20
4	3	BB	00:56	3	SB	00:29
⋮	⋮	⋮	⋮	⋮	⋮	⋮
74	73	BB	21:56	73	SB	10:59
75	74	BB	22:14	74	SB	11:08
76	75	BB	22:32	75	SB	11:17
77						
78						

Both *tickF(bBox,sBox,t)* and *fillF(bBox,sBox,t)* work in the same fashion, though they need both box tokens as input because the big boxes arrive later than the small ones: the maximum of three possible values (including the *timer*) determines the availability to the next processing step.

Table XVII shows the allocations on the last buffer *inT* and the finished *goods* storage. Unsurprisingly, the values match the ones from the clock pulse push model.

TABLE XVII. ALLOCATIONS OF TOKENS ON THE PLACES *inT* AND *goods* IN THE EVENT TRIGGERED PUSH MODEL AS DEPICTED IN FIGURE 6 FOR THE FIRST AND LAST FIVE SYSTEM STATES EACH

	inT			goods		
state	id	type	elapsed	id	type	elapsed
0						
1						
2						
3	1	FB	00:45			
4	2	FB	01:10	1	FB	00:52
⋮	⋮	⋮	⋮	⋮	⋮	⋮
74	72	FB	30:20	1|...|71	FB|...|FB	00:52|...|30:02
75	73	FB	30:45	1|...|72	FB|...|FB	00:52|...|30:27
76	74	FB	31:10	1|...|73	FB|...|FB	00:52|...|30:52
77	75	FB	31:35	1|...|74	FB|...|FB	00:52|...|31:17
78				1|...|75	FB|...|FB	00:52|...|31:42

Again, all these tables represent the entire reachability set of the Petri net model, although the combined table is much smaller in comparison to the clock pulse push model: It contains only 24 columns and 78 data sets or rows. However, storage costs need to be computed separately as a dashboard or tracking functionality was not implemented.

B. The Pull Model

The event triggered pull version of the *Box Game* as depicted in Figure 10 is partly more complicated, as supplementary elements are needed to implement pull requests. Otherwise, as time logging can be integrated by tracing token information in the net forward and backward, the *timer* and corresponding arcs may be dropped: instead of sending time information from and to a dedicated *timer* place, it can be integrated directly in the tokens and updated as needed.

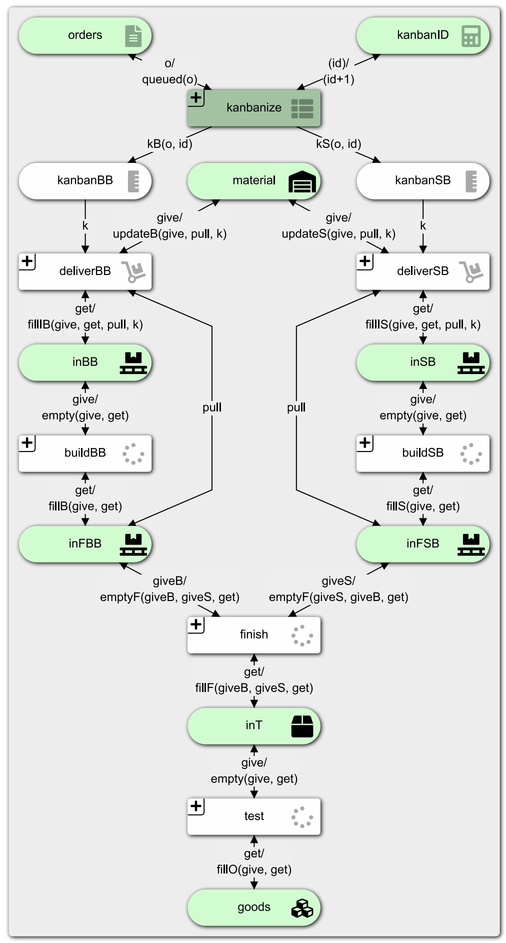


Fig. 10. Event triggered pull model after 5 ticks, the first completed box having just arrived on the finished goods warehouse.

As mentioned in Section II-A, information flows from the dispatch warehouse upstream. Thus, the kanban chain should start at *goods* instead of where it is modeled. However, kanban elements that are theoretically necessary for the model may partly be omitted for simplification.

Skipping these connections is feasible for the same reasons as in the clock pulse version: all pulls beside the modeled ones arise from working steps that require less time than *finish*, so their storage places are empty by specification at the relevant times, plus the omission eases readability of the visualization. For demonstration purposes, the batch size is set to *1*. Modeling the full kanban chain becomes necessary with larger batch sizes, i. e., when the buffers are actually filled.

The first additional elements in the event triggered pull model are the places *orders* and *kanbanID*, the latter being an integer to be used as *id* for single kanban requests. The *orders* place is typed with *ROrder*, consisting of the target *type*, the total ordered *volume*, the *batchSize*, and the amount of already *queued* items. The initial allocation of *orders* can be seen in Table XVIII. For this model, there is only one order placed for a total of 75 items. If there were more orders, an additional attribute *orderID* would become necessary to be tracked all through the net. The *kanbanID* is a sequential number initialized with *0*.

TABLE XVIII. INITIAL ALLOCATION ON THE PLACE *orders* IN THE EVENT TRIGGERED PULL MODEL AS DEPICTED IN FIGURE 10

type	volume	batchSize	queued
FB	75	1	0

Two places, *kanbanBB* and *kanbanSB*, are not initialized. They are typed with the record set *RKanban*. A token is created for each kanban request when the transition *kanbanize* fires. The *id* is the one as supplied by *kanbanID*, *type* defines the necessary intermediate product (that could be stored on a place in a bill of materials, but is hard-coded in this model), and the *batchSize* that is directly copied from the *order*.

As a first step, tokens on *orders* are transposed into kanban tokens that adhere to the given batch size. To this end, the transmission *o* supplies the order that gets updated by *queued(o)* (adding the value of *batchSize* to the current value of *queued*) and put back on its original place. The information is also used to create kanban tokens on *kanbanBB* and *kanbanSB* via the function calls *k∗(o,id)*, integrating the corresponding kanban *id*. The kanban *id* itself is incremented and also put back on its origin. Table XIX gives an overview of the token allocations on the places *orders* and *kanbanID*.

TABLE XIX. ALLOCATIONS OF TOKENS ON THE PLACES *orders* AND *kanbanID* IN THE EVENT TRIGGERED PULL MODEL AS DEPICTED IN FIGURE 10 FOR THE FIRST THREE AND AROUND THE LAST TOKEN CREATING SYSTEM STATE

	orders				kanbanID
state	type	volume	batchSize	queued	id
0	FB	75	1	0	0
1	FB	75	1	1	1
2	FB	75	1	2	2
⋮	⋮	⋮	⋮	⋮	⋮
74	FB	75	1	74	74
75	FB	75	1	75	75
76	FB	75	1	75	75

The *k∗(o,id)* functions create tokens as excerpted in Table XX. The kanban tokens slowly keep accumulating on their respective places *kanbanBB* or *kanbanSB*.

TABLE XX. ALLOCATIONS OF TOKENS ON THE PLACES *kanbanBB* AND *kanbanSB* IN THE EVENT TRIGGERED PULL MODEL AS DEPICTED IN FIGURE 10 FOR THE FIRST AND LAST FIVE SYSTEM STATES EACH WHERE THESE PLACES CARRY TOKENS

	kanbanBB			kanbanSB		
state	id	type	batchSize	id	type	batchSize
1	1	BB	1	1	SB	1
2	2	BB	1	2	SB	1
3	2|3	BB|BB	1|1	2|3	SB|SB	1|1
4	2|3|4	BB|BB|BB	1|1|1	2|3|4	SB|SB|SB	1|1|1
5	3|4|5	BB|BB|BB	1|1|1	3|4|5	SB|SB|SB	1|1|1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
219	74|75	BB|BB	1|1	74|75	SB|SB	1|1
220	74|75	BB|BB	1|1	74|75	SB|SB	1|11
221	75	BB	1	75	SB	1
222	75	BB	1	75	SB	1
223	75	BB	1	75	SB	1

As seen comparably in other models, *material* is initialized with two tokens, one for each type. However, in this model the record set is used differently: Instead of ”moving tokens through the net”, several places are equipped with tokens and information moves in the net while the tokens themselves effectively stay on their original places. The initial allocation for *material* is presented in Table XXI, also showing the structure of the record set *RMat*: item *type*, *elapsed* time denoting last access, and *count* of available stocks.

TABLE XXI. INITIAL ALLOCATION OF TOKENS ON THE PLACE *material* IN THE EVENT TRIGGERED PULL MODEL AS DEPICTED IN FIGURE 10

type	elapsed	count
BB	00:00	75
SB	00:00	75

The remaining places *inBB*, *inSB*, *inFBB*, *inFSB*, *inT*, and *goods* are typed with the record set *RBox*, which expands *RMat* by the batch size, *orderID*, and *kanbanID*. For clarity of presentation, both *orderID* and *kanbanID*, though being used for *RBox*, is not shown in any of the tables! The initialization of these places is shown in Table XXII. The *type* denotes the kind of (intermediate) product that is allowed on the corresponding place.

TABLE XXII. INITIAL ALLOCATION OF TOKENS ON THE PLACES *inBB*, *inSB*, *inFBB*, *inFSB*, *inT*, AND *goods* IN THE EVENT TRIGGERED PULL MODEL AS DEPICTED IN FIGURE 10

place	type	elapsed	count	batchSize
inBB	BB	00:00	0	0
inSB	SB	00:00	0	0
inFBB	BB	00:00	0	0
inFSB	SB	00:00	0	0
inT	FB	00:00	0	0
goods	FB	00:00	0	0

Again, big box and small box preassemblies work the same way, thus only one side is explained in detail. The delivery transition *deliverBB* is enabled when the adjacent places carry fitting tokens, i. e., when following conditions are met:

1. There must not be an item on *inFBB*, i. e., *pull.count=0*, otherwise there would not be the need for a *pull* request. Also, the *pull* should be for the item *type* the delivery transition is actually relevant for, here *pull.type="BB"*.

2. There must not be an item on *inBB*, i. e., *get.count=0* as this would effectively lead to an item on *inFBB*, again rendering the *pull* unnecessary.

3. The fitting items must be supplied by the *material* warehouse as indicated by the kanban request, i. e., *give.type=k.type*. Additionally, the corresponding stock must meet the batch size, i. e., *give.count>=k.batchSize*.

Upon firing *deliverBB*, the corresponding pull request token gets removed from *kanbanBB*. The token on *material* is updated to reflect the withdrawal of items and the access time on the place. A new token is created on *inBB* that contains information about *elapsed* time, the item *count*, and the *batchSize*. The *pull* token is transferred back to its origin without a change. The functions *update∗(give,pull,k)* and *fill∗(give,get,pull,k)* account for the necessary delivery times. Also, the functions on the right side handling the small boxes account for the shorter processing time of those boxes, leading to a later dispatch. Table XXIII gives an overview over the progression of token allocations on *material*.

TABLE XXIII. ALLOCATIONS OF TOKENS ON THE PLACE *material* IN THE EVENT TRIGGERED PULL MODEL AS DEPICTED IN FIGURE 10 AROUND THE FIRST AND LAST FIVE SYSTEM STATES WITH TOKEN CHANGES

state	type	elapsed	count
1	BB|SB	00:00|00:00	75|75
2	BB|SB	00:00|00:09	74|74
3	BB|SB	00:00|00:09	74|74
4	BB|SB	00:00|00:09	74|74
5	BB|SB	00:15|00:24	73|73
⋮	⋮	⋮	⋮
220	BB|SB	29:50|29:59	2|2
221	BB|SB	30:15|30:24	1|1
222	BB|SB	30:15|30:24	1|1
223	BB|SB	30:15|30:24	1|1
224	BB|SB	30:40|30:49	0|0
⋮	⋮	⋮	⋮

Tables XXIV and XXV display some allocations of the buffer places *inBB*, *inSB*, *inFBB*, and *inFSB*. The first big box arrives in the preassembly buffer after 2 seconds, directly being passed into the working station. As the small boxes only need half the processing time, i. e., 9 seconds, the first small box arrives after 11 seconds. Thus, both boxes are ready at the same time for the final assembly. This can be seen as the buffers for *finish* receive the first boxes after 20 seconds. Because of the final assembly taking 25 seconds, these buffers need to hold the next preassembled boxes ready after 45 seconds. Thus, folding for the second big and small boxes should start at 27 and 36 seconds, respectively.

TABLE XXIV. ALLOCATIONS OF TOKENS ON THE PLACES *inBB* AND *inSB* IN THE EVENT TRIGGERED PULL MODEL AS DEPICTED IN FIGURE 10 AROUND THE FIRST AND LAST SIX SYSTEM STATES WITH TOKEN CHANGES

	inBB				inSB			
state	type	elapsed	count	batchSize	type	elapsed	count	batchSize
1	BB	00:00	0	0	SB	00:00	0	0
2	BB	00:02	1	1	SB	00:11	1	1
3	BB	00:02	0	1	SB	00:11	0	1
4	BB	00:02	0	1	SB	00:11	0	1
5	BB	00:27	1	1	SB	00:36	1	1
6	BB	00:27	0	1	SB	00:36	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
222	BB	30:27	0	1	SB	30:36	0	1
223	BB	30:27	0	1	SB	30:36	0	1
224	BB	30:52	1	1	SB	31:01	1	1
225	BB	30:52	0	1	SB	31:01	0	1
226	BB	30:52	0	1	SB	31:01	0	1
227	BB	30:52	0	1	SB	31:01	0	1

TABLE XXV. ALLOCATIONS OF TOKENS ON THE PLACES *inFBB* AND *inFSB* IN THE EVENT TRIGGERED PULL MODEL AS DEPICTED IN FIGURE 10 AROUND THE FIRST AND LAST SIX SYSTEM STATES WITH TOKEN CHANGES

	inFBB				inFSB			
state	type	elapsed	count	batchSize	type	elapsed	count	batchSize
1	BB	00:00	0	0	SB	00:00	0	0
2	BB	00:00	0	0	SB	00:00	0	0
3	BB	00:20	1	1	SB	00:20	1	1
4	BB	00:20	0	1	SB	00:20	0	1
5	BB	00:20	0	1	SB	00:20	0	1
6	BB	00:45	1	1	SB	00:45	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
222	BB	30:45	1	1	SB	30:45	1	1
223	BB	30:45	0	1	SB	30:45	0	1
224	BB	30:45	0	1	SB	30:45	0	1
225	BB	31:10	1	1	SB	31:10	1	1
226	BB	31:10	0	1	SB	31:10	0	1
227	BB	31:10	0	1	SB	31:10	0	1

Due to the *pull* requests, both unfolded boxes are supplied at the right time in the preassembly buffers. The remainders of the tables shows the allocations for the last system states.

Table XXVI examines the allocations for the last two places, the test buffer *inT* and the finished *goods* store.

TABLE XXVI. ALLOCATIONS OF TOKENS ON THE PLACES *inT* AND *goods* IN THE EVENT TRIGGERED PULL MODEL AS DEPICTED IN FIGURE 10 AROUND THE FIRST AND LAST FIVE SYSTEM STATES WITH TOKEN CHANGES

	inT				goods			
state	type	elapsed	count	batchSize	type	elapsed	count	batchSize
3	FB	00:00	0	0	FB	00:00	0	0
4	FB	00:45	1	1	FB	00:00	0	0
5	FB	00:45	0	1	FB	00:52	1	1
6	FB	00:45	0	1	FB	00:52	1	1
7	FB	01:10	1	1	FB	00:52	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
223	FB	31:10	1	1	FB	30:52	73	1
224	FB	31:10	0	1	FB	31:17	74	1
225	FB	31:10	0	1	FB	31:17	74	1
226	FB	31:35	1	1	FB	31:17	74	1
227	FB	31:35	0	1	FB	31:42	75	1

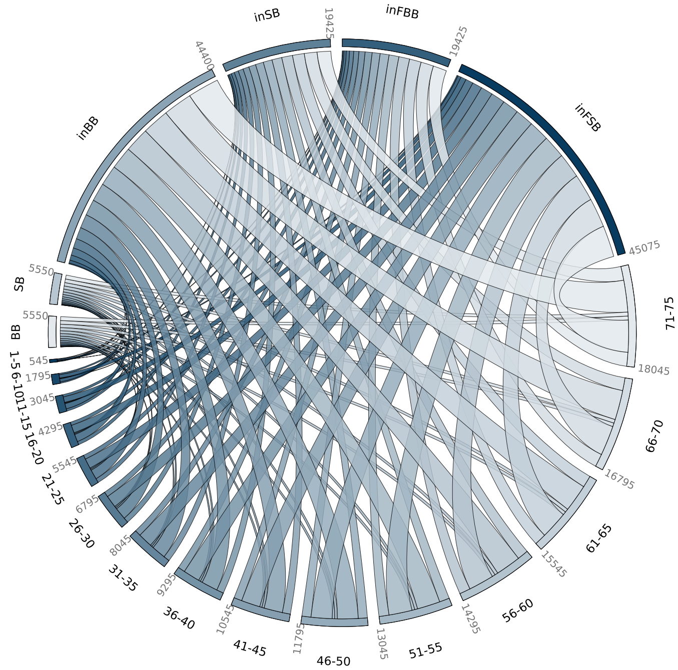


Fig. 11. Accumulated waiting times [in seconds]
in the event triggered push model:
(upper half circle) named storage places
(lower half circle) clusters of five consecutive boxes
(circular area) distribution of accumulated waiting times
of five-box-clusters to individual storage places.

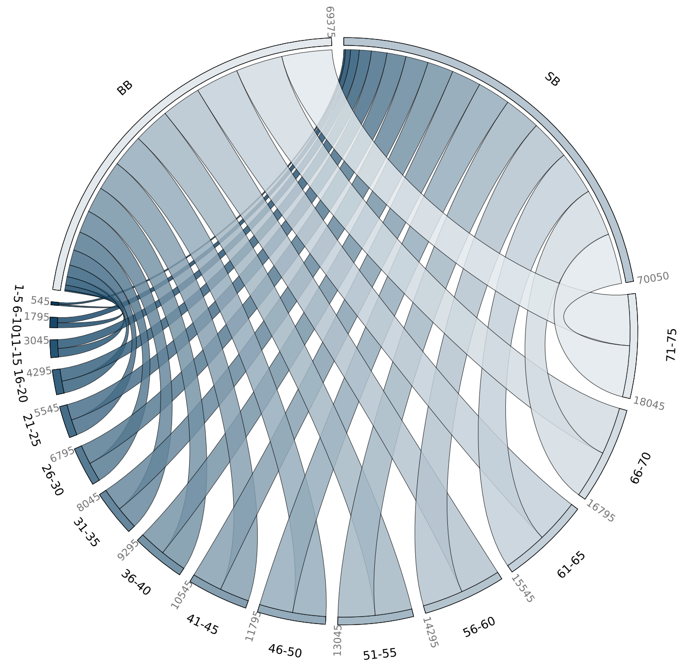


Fig. 12. Accumulated waiting times [in seconds]
in the event triggered pull model:
(upper half circle) named storage places
(lower half circle) clusters of five consecutive boxes
(circular area) distribution of accumulated waiting times
of five-box-clusters to individual storage places.

The first finished box arrives at *inT* after 45 seconds. The quality test starts immediately and the checked box is passed to the outgoing store after a total elapsed time of 52 seconds.

As is the case for the event triggered push model, storage costs are not directly visible in this result set and have to be processed separately. However, the tokens on the finished *goods* warehouses from both models resemble each other.

C. Simulation Results

As expected, the simulation of the two models show no difference in total processing or idle times for single workplaces. Although, differences on storage places become evident again.

Because the distribution of waiting times behind *finish* is the same for both models, only the first five storage places need to be analyzed. Figure 11 shows the allocation of the boxes' waiting times to single stocks - and hence the distribution of storage costs - in the push model while Figure 12 shows the same for the pull model.

The stocks on *material* are split by type (*BB* and *SB* in the figures). The boxes themselves are aggregated into five-box-clusters for clarity. The trend to successively longer waiting times is clearly visible.

The push model's *material* gets cleared as fast as possible. This, however, leads to large buffers (*inBB*, *inSB*) just before the first concurrent production steps. As the preassembly requires less time than the final assembly, the two incoming buffers for *finish* (*inFBB*, *inFSB*) are also highly occupied.

As anticipated, the pull principle leads to zero interim buffer inventory throughout the stocks under consideration. This is clearly visible as, although the total accumulated waiting times for both models are equivalent, the pull model unites them on the main storage *material* only.

This opens up the possibility to externalize costs. One such alternative is to use consignment stores: suppliers maintain warehouses inside the customer's facility and, thus, must keep the material in their balance sheet. Another common possibility is just-in-time delivery. With high process stability and fast response times, even the use of a kanban cycle with the supplier is conceivable [6]. These options may lead to partial or total eradication of stocks.

The final remark of the former section regarding muda and friction between local vs. global optima still holds true.

VIII. Conclusion and Future Work

The advantage of simulating the *Box Game* using clock pulse models is that the fluctuation of stocks can be observed in a very illustrative way while the simulation is running. The amount of items in each storage is clearly visible for every second. Since in the *P-S.C* the color of and the symbol on places can change depending on the number of tokens on the place, the advantages and disadvantages of push and pull can be demonstrated very figuratively. As a consequence, the simulation is a demonstrative extension of the personal experience for the students in the logistics laboratory.

Event triggered models on the other hand emphasize the end results over the visualization during runtime. As such, they show advantages in computation times.

A. Computational Times

Using either of the clock pulse models, the presented scenario runs for 1902 discrete time steps that represent seconds in the *Box Game*. This equals to almost 32 minutes. Although this is a much longer period than one that can be played by students - because concentration levels decrease after about 5 minutes - it seems unnecessary as steps are computed at which no change of the system's state occurs.

Even though, calculating these steps takes time. In Chrome on an iMac (4 GHz Quad-Core Intel Core i7, 16 GB RAM) the full simulation takes on average 8234 milliseconds. The duration for simulating a working day increases linearly.

However, this effort can be reduced drastically if the concrete result is more important than the runtime visualization. In this case, it would be sufficient to calculate new states only in the moments of state changes. Using event triggered models, the simulation for the given scenario can be reduced to 79 steps for the push and 228 steps for the pull version. The difference is due to the kanban calculations needed during runtime. On the aforementioned computer, the simulations only run for 315 and 923 milliseconds on average, respectively. The advantages of event triggered simulation increase more if the demand interval in which changes occur vary strongly from one part of the model to the other since it takes into account local state changes. Hence, the latter approach scales better, making it beneficial for industrial applications.

B. Result Presentation

The primary presentation advantage of the clock pulse models - visualization of state changes during runtime - cannot inherently be conveyed without animation.

Generally, the simulation results of Petri net models can be deducted from the nets' reachability sets. The *P-S.C* provides functionalities for completely or partially logging the actually reached system states, i. e., the simulation results, and for exporting this information in form of comma separated values. For the diagrams presented in Sections VI and VII, such data was used to generate the visualizations in external programs in an individual working step.

Incidentally, this lead to the question as to how to present data to the target audience. However, this audience and their information needs have to be determined in the first place. Modeling and domain expertise often exist disparately, so modelers first need to understand which goals the domain experts want to achieve and which indicators can be used for reaching them.

The cumbersome external processing step and the requirement for an audience orientation combined show the necessity for an internal visualization solution. The goal is to supply modelers with fitting tools in form of an adaptable, integrated dashboard. As a first step, a corresponding research-agenda has been established [36].

C. Key Takeaways

The challenges that had to be overcome for implementing the presented models led to some new insights in the development of conceptual models for discrete time-dependent systems. Eventually, the authors found their personal best practice that consists of the following steps:

1. Define data types for the different stocks and other data objects. Then, initialize the corresponding places in accordance with the starting condition.
2. Augment the model by transitions for beginning and ending specific tasks like delivering raw materials, building or quality testing.
3. Identify the next item to be taken and the moment this will occur. This also allows for implementation of different prioritization strategies.
4. Start with modeling the simpler push principle and augment this model by pull principles.
5. Look for a proper visualization of the simulation results.

Moreover, at this time the development of a clock pulse model may be seen as a preliminary for the development of an event triggered model as some perceive the implementation of the event triggered models as more difficult in comparison. If an event triggered model is needed, the following steps - especially the second one - might be helpful:

1. Develop the clock pulse model first.
2. Observe the reasons for state changes with the aid of the clock pulse model and derive the event triggered model from these observations.
3. Look for a proper visualization of the simulation results.

From personal observations, the authors derive the following suggestion on when to use which modeling approach:

Use a clock pulse simulation if either a clock pulse visualization of the system's states is needed or if the computer is fast enough for the few simulations that must be run for the modeled system.

Use an event triggered simulation if high simulation speed is necessary due the complexity of the modeled system, if fast answers are needed in production, or if a large number of variations of the production schedule or input data has to be compared. In general, the more often a specific model needs to be simulated, the more it is worthwhile to develop an event triggered model instead of a clock pulse model.

In order to ease the step from a clock pulse model to an event triggered model, in the future the *P-S.C* will receive an extension such that users are supported in finding the relevant moments of state changes.

D. What else?

The biggest impression on the authors was the realization that what can be modeled and simulated with the aid of Petri nets is only restricted by the modelers imagination and the ability of the used tool. In opposite to other out of the box modeling environments for logistics, a user is free to lay the focus on any parameter they are interested in most.

However, two further challenges exist: Modelers must be able to develop sophisticated, abstract models and they must find a way to visualize the results. The authors hope to have given a possible answer to the first challenge and see major future work concerning the second one.

The *P-S.C* itself and the lessons learned regarding modeling are used for theoretical and practical research.

The bullwhip effect, for example, is a phenomenon in logistics that is widely known, but rarely examined from a simulation perspective. However, using the *P-S.C* such a simulation becomes manageable, giving insight into the basic mechanism and possibilities to prevent the effect [37].

On the practical side, the authors work on simulating and evaluating the processes used in a large hazardous goods warehouse that is currently be built. The materials in question will be transferred from an older warehouse, however, the established processes can not be adopted due to scale, handling and legal aspects. Thus, it is necessary to examine the system's behavior for problems and anomalies.

Teaching aspects, as shown in the *Box Game*, theoretical and practical research cross-fertilize each other. In effect, this leads to better tools, improved modeling competencies, and more wide-spread usage of simulation and its benefits.

References

[1] S. Haag, L. Zakfeld, C. Simon, and C. Reuter, "Event Triggered Simulation of Push and Pull Processes," in *SIMUL 2020: The Twelfth International Conference on Advances in System Simulation*, L. Parra, Ed., Porto (Portugal), 2020, pp. 68–73.
[2] C. Simon, S. Haag, and L. Zakfeld, "Clock Pulse Modeling and Simulation of Push and Pull Processes in Logistics," in *SIMMaApp: Special Track at SIMUL 2020: The Twelfth International Conference on Advances in System Simulation*, F. Herrmann, Ed., Porto (Portugal), 2020, pp. 31–36.
[3] M. Zhou and K. Venkatesh, *Modeling, Simulation, and Control of Flexible Manufacturing Systems - A Petri net Approach*, ser. Intelligent Control and Intelligent Automation. Singapore, New Jersey: World Scientific, 1999, vol. 6.
[4] U. Dombrowski and T. Mielke, *Ganzheitliche Produktionssysteme: Aktueller Stand und zukünftige Entwicklungen*. München: Springer Vieweg, 2014, German, transl. *Holistic production systems: Current status and future developments.*
[5] J. Gottmann, *Produktionscontrolling: Werströme und Kosten optimieren*, 2nd ed. Wiesbaden: Springer Gabler, 2019, German, transl. *Production controlling: Optimizing value flows and costs.*
[6] H. Wildemann, *Kanban-Produktionssteuerung*, 28th ed. München: TCW, 2020, German, transl. *Kanban production control.*
[7] L. Recalde, M. Silva, J. Ezpeleta, and E. Teruel, *Lectures on Concurrency and Petri Nets: Advances in Petri Nets*. Berlin: Springer, 2004, ch. Petri nets and manufacturing systems: An examples-driven tour, pp. 742–788.
[8] W. Reisig, *Understanding Petri Nets*. Berlin: Springer, 2013.
[9] H. J. Genrich and K. Lautenbach, "System Modelling with High-Level Petri Nets," *Theoretical Computer Science*, vol. 13, 1981.
[10] K. Jensen, *Coloured Petri-Nets*, 1st ed. Berlin: Springer, 1992.
[11] M. Montali and A. Rivkin, "From DB-nets to Coloured Petri Nets with Priorities (Extended Version)," *CoRR*, vol. abs/1904.00058, 2019.
[12] C. Ramchandani, "Analysis of Asynchronous Concurrent Systems by Timed Petri Nets," MIT, Project MAC, Technical Report 120, 1974.
[13] P. Merlin, "The Time-Petri-Net and the Recoverability of Processes," University California, Irvine, Tech. Rep., 1974.
[14] J. Sifakis, "Use of petri nets for performance evalutation," in *Measuring, modelling and evalutating computer systems*, ser. IFIP, H. Beilner and E. Gelenbe, Eds., North Holland Publ. Co., 1977, pp. 75–93.
[15] H.-M. Hanisch, *Petri-Netze in der Verfahrenstechnik*. München: Oldenbourg, 1992, German, transl. *Petri nets in process engineering.*
[16] H.-M. Hanisch, "Dynamik von Koordinierungssteuerungen in diskontinuierlichen verfahrenstechnischen Systemen," in *Petrinetze in der Automatisierungstechnik*, E. Schnieder, Ed. München, Wien: Oldenbourg Verlag, 1992, German, transl. *Dynamics of coordination controls in discontinuous process engineering systems.*
[17] R. König and L. Quäck, *Petri-Netze in der Steuerungs- und Digitaltechnik*. München, Wien: Oldenbourg Verlag, 1988, German, transl. *Petri nets in control and digital technology.*
[18] C. Ghezzi, D. Mandrioli, S. Morasca, and M. Pezzè, "A unified high-level petri net formalism for time-critical systems," *IEEE Transactions On Software Engineering*, vol. 17, no. 2, pp. 160–172, 1991.
[19] H.-M. Hanisch, K. Lautenbach, C. Simon, and J. Thieme, "Timestamp Nets in Technical Applications," in *IEEE International Workshop on Discrete Event Systems*, San Diego, CA, 1998.
[20] C. Simon, "Developing Software Controllers with Petri Nets and a Logic of Actions," in *IEEE International Conference on Robotics and Automation, ICRA 2001*, Seoul, Korea, 2001.
[21] K. Jensen, "High-Level Petri Nets," *Informatik-Fachberichte*, vol. 66, pp. 166–180, 1983.
[22] L. Popova-Zeugmann, *Time and Petri Nets*. Berlin: Springer, 2013.
[23] C. E. Knoeppel, *Installing Efficiency Methods*. The Engineering Magazine, 1915.
[24] T. Ohno, *Toyota Production System*. Milton Park, UK: Taylor & Francis, 1988.
[25] BPMI, "BPMN 1.0 - Business Process Model and Notation," https://www.omg.org/spec/BPMN/ (last accessed 2021.11.20), 2004.
[26] OMG, "BPMN 2.0 - Business Process Model and Notation," http://www.bpmn.org/ (last accessed 2021.11.20), 2011.
[27] C. Simon, "Web-Based Simulation Of Production Schedules With High-Level Petri Nets," in *32rd International ECMS Conference on Modelling and Simulation (ECMS 2018)*, L. Nolle, A. Burger, C. Tholen, J. Werner, and J. Wellhausen, Eds. Wilhelmshaven, Germany: SCS Europe, 2018, pp. 275–281.
[28] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research," *MIS Q.*, vol. 28, no. 1, pp. 75–105, mar 2004.
[29] C. Simon and S. Haag, "Simulatable Reference Models To Transform Enterprises For The Digital Age – A Case Study," in *ECMS 2020: 34th International ECMS Conference on Modelling and Simulation*, M. Steglich, C. Müller, G. Neumann, and M. Walther, Eds., 2020, pp. 294 – 300.
[30] C. Simon and S. Haag, "A Case-Study to Teach Process-Aware Information Systems," *EMISA Forum: Proceedings of the SIG Enterprise Modelling and Information Systems Architectures of the German Informatics Society*, vol. 40, pp. 9–10, 2020.
[31] S. Haag and C. Simon, "Simulation of Horizontal and Vertical Integration in Digital Twins," in *ECMS 2019: 33rd International ECMS Conference on Modelling and Simulation*, M. Iacono, F. Palmieri, M. Gribaudo, and M. Ficco, Eds., 2019, pp. 284 – 289.
[32] C. Simon and S. Haag, "Simulation vertikaler Integration: Vom Top-Floor zum Shop-Floor und zurück," in *Tagungsband AKWI*, T. Barton, F. Herrmann, V. G. Meister, C. Müller, C. Seel, and U. Steffens, Eds., 2018, pp. 104 – 113, German, transl. *Simulation of vertical integration: from top floor to shop floor and back again.*
[33] C. Simon, "Eine Petri-Netz-Programmiersprache und Anwendungen in der Produktion," in *Tagungsband AKWI*, T. Barton, F. Herrmann, V. G. Meister, C. Müller, and C. Seel, Eds., 2017, pp. 61–70, a Petri net programming language and applications in production.
[34] C. Simon, S. Haag, and L. Zakfeld, "Simulation taktgesteuerter Modelle von Push- und Pull-Prozessen in der Logistik," *Anwendungen und Konzepte der Wirtschaftsinformatik*, vol. 13, pp. 27–33, 2021, German, transl. *Simulation of clock-controlled models of push and pull processes in logistics.*
[35] Petri Nets World, "Petri Nets Tools Database Quick Overview," https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html (last accessed 2021.11.20), 2020.
[36] C. Simon, S. Haag, and L. Zakfeld, "Research-Agenda for Process Simulation Dashboards," in *ECMS 2021: 35th International ECMS Conference on Modelling and Simulation*, 2021, pp. 243–249.
[37] C. Simon, L. Zakfeld, C. E. Jensen, D. Klietsch, and M. Montag, "Can simulation prevent companies from the bullwhip trap? New approaches to model the bullwhip effect with the aid of Excel and high-level Petri nets," in *SIM-SC : Special Tack at SIMUL 2021 : The Thirteenth International Conference on Advances in System Simulation*, F. Herrmann, M. Popescu, and M. Audette, Eds., 2021, pp. 31–37.

Methodology for Extracting Knowledge from a Gaming Simulation Using Data Envelopment Analysis

Akinobu Sakata
Department of Computational Intelligence and System Science
Tokyo Institute of Technology
Yokohama, Japan
e-mail: sakata.a.aa@m.titech.ac.jp

Takamasa Kikuchi
Graduate School of Business Administration
Keio University
Yokohama, Japan
e-mail: takamasa_kikuchi@keio.jp

Ryuichi Okumura
Career Innovation Division
Mitsubishi Research Institute,Inc.
Tokyo, Japan
e-mail: okuty@mri.co.jp

Masaaki Kunigami
Department of Computer Science
Tokyo Institute of Technology
Yokohama, Japan
e-mail: mkunigami@gakushikai.jp

Atsushi Yoshikawa
Department of Computer Science
Tokyo Institute of Technology
Yokohama, Japan
e-mail: at_sushi_bar@dis.titech.ac.jp

Masayuki Yamamura
Department of Computer Science
Tokyo Institute of Technology
Yokohama, Japan
e-mail: my@c.titech.ac.jp

Takao Terano
Chiba University of Commerce
Ichikawa, Japan
e-mail: terano@cuc.ac.jp

***Abstract*—In this study, we propose a methodology for extracting knowledge about which play logs are superior (inferior) to other play logs under certain criteria from the results of a gaming simulation. Previous research has enabled facilitators to know where players' play logs output from gaming simulations are positioned in all possible scenarios. However, facilitators have no valid solution to encourage players to change their behavior in gaming simulations. The proposed methodology enables a facilitator to identify the players who show similar behavior and performance to the target player under certain criteria, and to present to the play logs which show superior performance than the target player's play log to the target player. In order to achieve our research objective, we created several software agents instead of human players to play a gaming simulation for career education, and analyzed the output play logs using data envelopment analysis. As a result, the desired knowledge was extracted. We argue that the extracted knowledge should be applied for debriefing. The proposed methodology is flexible enough to work under both conditions where all players are human and where human and machine agents are mixed as players.**

***Index Terms*—Gaming simulation; data envelopment analysis; knowledge extraction; debriefing; facilitation.**

I. INTRODUCTION

The objective of this study is to propose a methodology to extract knowledge about which play logs are superior (or inferior) to other play logs under certain criteria from the results of a gaming simulation. In order to achieve our research objective, we use a gaming simulation that is a modified version of the Shin-Life Career Game [1] presented at eKNOW 2021, and the Data Envelopment Analysis (DEA) [2]. A gaming simulation is a simulation in which humans participate in the simulation situation as players and are influenced by the decisions of those players [3]. DEA is a method that uses linear programming to measure the efficiency of a decision-making unit (DMU) by enveloping the observed input-output vectors as tightly as possible [4].

Facilitation and debriefing are important in gaming simulations [5]. Facilitation is that a facilitator takes care of everything related to the progress of the game [6]. Debriefing is learning by reflecting on the simulation experience [7]. However, it is said that the effectiveness of facilitation and debriefing depends largely on the experience and skills of a facilitator [6].

According to Kikuchi et al. [8], in order to evaluate a player's behavior in a gaming simulation, it is common to focus on individual play logs, track a player's decisions and actions, and observe them in detail (e.g., [9], [10]) . The approach of analyzing individual play logs requires a great deal of effort and cost. It is therefore difficult to compare and evaluate the play logs of a large number of different players.

In contrast to the approach of analyzing individual play logs, there is an approach of comprehensively analyzing the results of gaming simulations. Kikuchi et al. point out the following challenges in comprehensively analyzing the results of business games [8]: (1) the collection of a large number of play logs, (2) the possibility of a biased sample, and (3) the difficulty of creating a list that covers all possibilities. Business games are a type of gaming simulation. Business games are a common means of studying business and management principles under controlled and virtual situations [11]. Kikuchi et al. point out that because of the difficulty in overcoming issues raised above, methods for evaluating gaming outcomes based on game wins and losses are overused.

Based on these arguments, Kikuchi et al. proposed a framework for properly evaluating the behavior of players in computer-based business games [8]. In the framework proposed by Kikuchi et al., the following steps are taken: (1) Agent-based Model is constructed based on the target business game, computer simulations are run thoroughly, and the logs are classified. (2) Identify the behavior of players in gaming by locating the experimental results (play logs) generated by players in the classified computational results. Kikuchi et al. argue that by following these steps of analysis, it is possible to visualize and present to the players and facilitators the position of the players in the possible outcomes of business games.

The analytical framework proposed by Kikuchi et al. is applicable to all gaming simulations. This is because the arguments of Kikuchi et al. are derived from a structure that is common to all gaming simulations, not just business games. By applying the framework proposed by Kikuchi et al. to a gaming simulation, players can know in which pattern their behavior in their playing experience is included in all possible gaming scenarios. This method provides an opportunity for players to recognize what differences there are between their own behavior and that of other players.

However, previous approaches are unlikely to encourage players to improve their play. This is because even if players know the position of their gaming outcome in possible scenarios, they do not know what they can refer to for changing their behavior in their gaming.

Based on the above, we propose a methodology for acquiring knowledge that facilitates a player to change his/her behavior in a gaming simulation. Specifically, the following steps are taken: (1) have players play the gaming simulation and collect play logs, (2) classify the play logs, and (3) evaluate the superiority or inferiority of the play logs under the criteria specific to each group. By following these steps, we can acquire knowledge, which play logs are superior (inferior) to others under the same criteria. As a result, a facilitator will be able to use the acquired knowledge to narrow down the play log that can serve as a model for the players to be instructed, and to give appropriate advice to the players. In addition, players can learn effectively and efficiently by referring to the facilitator's advice. The above analysis procedures are also valid if some of the players are replaced by machine agents.

The differences between our analytical approach and that proposed by Kikuchi et al. [8] are as follows: (1) the players in the gaming simulation can be either human-only or a mixture of human and machine agents, (2) when the players are human-only (which limits the number of experiments), the superiority or inferiority among play logs among a limited number of play logs can be helpful to the players. Of course, in conditions where both humans and software agents are mixed, a large number of play logs should be collected by computer simulation, as in the study by Kikuchi et al..

In this study, we demonstrate our proposed methodology with solving a example problem. Specifically, we assume software agents to be human players, have them play a gaming simulations, collect play logs, classify the play logs using Charnes-Cooper-Rhodes model (CCR model) [2], which is the basic model of DEA, and extract knowledge about which play logs are better than other play logs under a specific objective function, or which play logs are worse than other play logs under the same objective function. The gaming simulation to be used in this demonstration is the Shin-Life Career Game version 2 developed for career education [12]. The Shin-Life Career Game version 2 is a kind of typical life game that allows players to experience a virtual life as a worker. The Shin-Life Career Game version 2 consists of multiple rounds, and in each round, players are forced to make life choices and solve resource allocation problems according to their own will. The details of this gaming simulation are described in Section 2.

The rest of this paper is organized as follows: First, related research (Sections 2 and 3) is described. Section 2 provides a detailed description of the Shin-Life Career Game version 2 used in this research and Section 3 describes DEA. Sections 4 through 6 provide an example of applications of the proposed methodology and finally, a summary and conclusions are provided.

II. The Shin-Life Career Game

In this study, a gaming simulation named the Shin-Life Career Game version 2 [12] is used in the experiments. This game was developed by adding new functional elements to the original Shin-Life Career Game [1]. In the following, we first describe how and why the original Shin-Life Career Game was developed, and then describe the Shin-Life Career Game version 2.

A. The Shin-Life Career Game (Original Version)

The Shin-Life Career Game is an updated version of the Life Career Game developed by Boocock [13][14] that reflects modern work elements. To overcome the discrepancy between the career world as seen by secondary school youth and the career world as seen by adults, Boocock developed the Life Career Game, which plays much like the original Life Game [15]. The players of the game experience a hypothetical life, playing various roles and spending their resources (money, time, etc.) on various activities with the aim of maximizing their present satisfaction and the possibility of a good life in the future. As a result, they acquire knowledge related to career development and develop understanding and confidence. These characteristics have been partially inherited by serious games for career education that have been developed since then (e.g., [1], [16] [17] [18]).

Resource Variable	Conventional Games	The Shin-Life Career Game
Number of labors a player can have at the same time	One	Multiple
Diversity of labor forms	Low	High

TABLE I: Difference in characteristics between the Shin-Life Career Game and traditional gaming simulations (Reprinted from [1])

The situation that the Life Career Game developed by Boocock offers to players is somewhat unrealistic in today's working society. In the game, players are required to make a living from only one job, and they cannot choose the type of work they do when they enter the profession. It is natural for a modern worker to have a variety of options, such as working as an employee or as a freelance worker.

The Shin-Life Career Game was developed to solve this problem. It is a modern version of the Life Career Game in which players can choose any number of activities such as work and skill development at the same time, as long as the constraints are satisfied. Specifically, in the Shin-Life Career Game, players can choose from various types of work, such as permanent work, freelance work, and simple work. And, they can also choose learning to develop their ability.

In the Shin-Life Career Game, players are asked to live out their lives as virtual workers through game play. In the course of their virtual lives, players determine their own goals in life, decide what and how to spend their resources: earn money, improve their abilities, and sometimes take a break. Just as in real life, players may lose money or get into trouble. A table describing the differences in characteristics between the Shin-Life Career Game and previous career education games from the study [1] is reproduced above (see Table I).

Munsen et al. is skeptical that the experiences of players who participated in gaming simulations that imitate the life of a worker adequately reflect real-life [19]. In response to these negative views, Duke, the founder of the International Society for Simulation and Gaming, said the following [20]: gaming simulation is a useful method for representing dynamic models that abstract complex realities; gaming simulation is appropriate for gaining a holistic understanding of complex situations, and; it helps to deepen the contemplation of multiple futures by considering multiple alternatives in a gaming simulation. We have supported Duke's position and developed the Shin-Life Career Game (and version 2). Players who participate in the Shin-Life Career Game will be assigned the role of a worker, experience a virtual life from the perspective of a worker, and interpret the life of a worker from their perspectives in the course of the experience, gaining new knowledge and perspectives through the exchange of opinions among the players and facilitators.

B. The Shin-Life Career Game version 2

In this study, we use the Shin-Life Career Game version 2 [12], which is an extension of the original Shin Life Career Game described above. In the Shin-Life Career Game, players have a total of four types of resources (money, ability, time, and health), which they can freely distribute among five types of activities (permanent work, freelance work, simple work, learning, and leisure). The main differences between Shin-Life Career Game version 2 and the original Shin-Life Career Game are: (1) the number of resource variables the player has been increased by one (the health resource variable has been added), (2) the number of activities that can be selected has been increased by one (Leisure has been added), and (3) the health resource variable affects the performance of work and learning activities. With these new elements added to the game model, the decision-making problem regarding the allocation of resources handled by players becomes more complex, and the behavior of the game system becomes more dynamic. The differences in the specifications of the two games are summarized in Table II.

Figure 1 depicts a model that represents which resources are gained (or lost) when a player allocates certain resources for his activities in the Shin-Life Career Game version 2. This model, called MATH model, describes the four basic resource variables of an agent: monetary assets (M: money), ability (A: ability), time (T: time), and health and fitness (H: health), in N^2 diagram (duality) form, with each variable as input and output [21]. This is a logical model in which the agent's activity is described as a region that crosses the corresponding rows and columns [21].

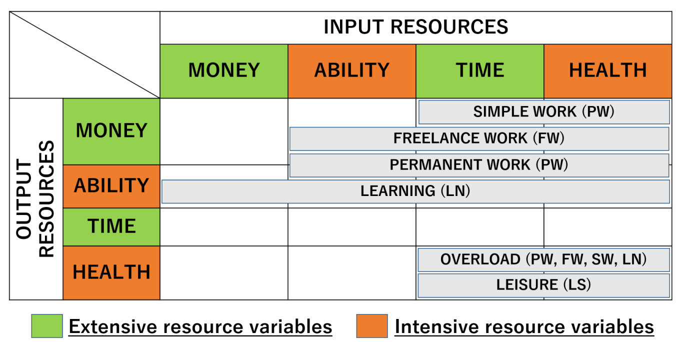


Fig. 1: This figure shows the relationship between the resources input to each activity and the resources output from each one in the Shin-Life Career Game version 2. For example, players input time resources and health ones to SW, and they acquire money resources. In a given round, if a player inputs extensive resources, such as money and time, into multiple activities, the resources decrease with each resource input because extensive resource is allocated competitively and additively. Also, in a given round, if a player inputs intensive resources, such as ability and health, into multiple activities, the same amount of the resources are allocated to the target activities because intensive resource is non-competitively and multiplicatively. The amount of extensive and intensive resources increase, decrease, or remain the same depending on a result of a target activity. A detailed description of the properties of resource variables and activities can be found in "1) Resource Variable" and "2) Activity", respectively.

Category	Elements	The Shin-Life Career Game†	
		Original Version	Version 2
Resources	Money	◯	◯
	Ability	◯	◯
	Time	◯	◯
	Health	×	◯
Labors	Permanent work (PW)	◯	◯
	Freelance work (FW)	◯	◯
	Simple work (SW)	◯	◯
	Learning (LN)	◯	◯
	Leisure (LS)	×	◯
Efficiency‡	Money	×	◯
	Ability	×	◯

† ”◯” means that the element is integrated into the game model. ”×” means that the element is not integrated into the game model.
‡ Efficiency of increasing resources through activities.

TABLE II: Difference in composition elements between the Shin-Life Career Game (Original version) and the Shin-Life Career Game version2

This logic diagram is interpreted as follows. When ability resources, time resources, and health resources are input to permanent work (PW), freelance work (FW), and simple work (SW), money resources are output, and the player obtains money resources. For permanent work (PW), ability resources are also output, and the player's ability increases. When ability resources, time resources, and health resources are input to learning (LN), ability resources are output, again increasing the player's ability. When time and health resources are input to leisure (LS), health resources are output, and the player's health is restored. The last resource input/output to overload (OL) means that when time and health resources are input to permanent work, freelance work, simple work, and learning, health resources are reduced. A detailed description of each resource variable and each activity is left to the subsequent sections.

1) Resource Variable: In the following, we explain the resource variables according to the studies [12][21]. The resource variable has two properties.

The first property is whether a resource variable is extensive or intensive. Extensive resource variables are additive and competitively allocated in resource allocation. If a player allocates his/her extensive resources for one objective, the resources reduce. And then, there are fewer resources available to allocate for other objectives. On the other hand, intensive resource variables are multiplicative and allocated non-competitively in resource allocation. If a player allocates his/her intensive resources for multiple objectives at the same time, unlike extensive resources, his/her intensive resources have the same effect on all objectives.

The second property is whether or not there is an upper bound on the value of a resource variable. If there is no upper limit to the value of a resource variable, it means that the resource is able to increase without limit. On the other hand, if there is an upper limit to the value of a resource variable, it means that the resource is not able to increase beyond a certain level. Here, a resource variable with no upper limit on its value is called an unbounded resource variable, while a resource variable with an upper limit on its value is called a bounded resource variable.

In the Shin-Life Career Game version 2, a player has a total of four types of resources (money resource, ability resource, time resource, and health resource). Money resource corresponds to an asset in the real life. Human beings use monetary assets to support their lives. Money resource is extensive and unbounded. Ability resource corresponds to knowledge and skills for work in the real life. Human beings use their skills and knowledge to engage in labor and get paid for it. Ability resource is intensive and unbounded. Time resource corresponds to time for real life. Human beings live their lives by spending their time in a variety of activities. Time resource is extensive and bounded. Health resource corresponds to health or physical fitness for real life. Health and physical fitness are the foundation of real life. Health resource is intensive and bounded. Table III summarizes the characteristics of these resource variables.

2) Activity: In the following, we describe the activities of the agents according to the cited research [12]. In this game, the player allocates his resources to several activities (permanent work (PW), freelance work (FW), simple work (SW), learning (LN), and leisure (LS)) in each round according to the constraints of the game and his own will. The individual activities are described below. Table IV summarizes the definitions and characteristics of the various types of activities in the Shin-Life Career Game version 2.

Permanent work (PW) is a labor form in which workers are employed by an organization until they reach retirement age and receive remuneration for their labor. The characteristics of permanent work are described below. First, the remuneration for permanent work is stable. In reality, formal workers are

Resource Variables	Properties Extensive / Intensive	Upper limit	Descriptions
Money resource	Extensive	No	The variable of money resource corresponds to an asset in the real life.
Ability resource	Intensive	No	The variable of ability resource corresponds to knowledge and skills for work in the real life.
Time resource	Extensive	Yes	The variable of time resource corresponds to time for real life.
Health resource	Intensive	Yes	The variable of health resource corresponds to health or physical fitness for real life.

TABLE III: Properties of resource variables

Activity	Symbol	Definition
Permanent work	PW	Permanent work (PW) is a way of working in which a person is employed by an organization until retirement and receives remuneration for providing labor power.
Freelance work	FW	Freelance work (FW) is a form of work in which a worker is independent of a particular organization and is paid for providing specialized knowledge and skills to a contractor.
Simple work	SW	Simple work (SW) is a form of work in which a worker provides labor time to an employer or contractor and is paid for it.
Learning	LN	Learning (LN) is the act of taking extra time to develop one's skills in order to develop one's ability to do a job.
Leisure	LS	Leisure (LS) is an activity that workers set aside to restore their physical and mental health.

TABLE IV: Definition of activities implemented in the Shin-Life Career Game Version2

less likely to experience unemployment due to the effects of the economy than non-formal workers [22]. Second, the remuneration for formal labor is higher than that for simple labor. In reality, the income of full-time workers is much higher than that of part-time workers [23]. Third, as a player's ability resources increase, his/her rewards of permanent work become higher. In the personnel evaluation system of a company, it is customary to reflect the medium- to long-term accumulation of capabilities by permanent workers in the increase of their basic salary and the promotion of their qualification grade [24]. Fourth, engagement in permanent work increases a player's ability resources. Firms provide formal workers with education or special jobs that encourage their growth [25]. Fifth, a player needs to provide a certain amount of time resources for permanent work. In general, the law sets minimum and maximum working hours for permanent workers. Sixth, a player cannot decide the amount of time resources to be allocated to permanent work at will. In general, in formal employment, workers are obligated to engage in overtime and holiday work according to the orders of their supervisors.

Freelance work (FW) is a labor form in which workers are independent of a particular organization and is paid by providing expertise and skills to a contracted party. Freelance labor has the following characteristics. First, the income of workers who engage in freelance work is unstable. The income of actual freelance workers is unstable as well [26]. Second, as a player's ability resources increase, the reward for freelance labor increases. In the online job brokerage services used by real freelance workers, workers with high value-added skills have a chance to get high-paying jobs [27]. Third, a player's ability resources do not increase when the player engages in freelance labor. Employers spend less time on training self-employed workers than employees do [28]. Our model have not been designed to take into account the experience through freelance work based on the views obtained from interviews with a real freelance worker. But, it is debatable whether the experience of engaging in freelance work should be included in ability resource. Fourth, the player is free to decide the amount of time resources to be allocated to freelance labor. Since freelance workers do not have an employment contract, they do not have the obligation or responsibility to have their working hours controlled by others [29].

Simple work (SW) is a labor form in which workers provide their time to their employers or contractors and are paid for it. In this game, simple work is the kind of work that manual workers do in the real world, such as part-time jobs, day labor, and gig work, which do not require any special skills or qualifications. Simple work has the following characteristics. First, the reward (the money resource) increases in proportion to the amount of time resource the player allocates to simple work. In general, the wages of part-timers are determined by the length of time they work. Similarly, the wages of day laborers and gig workers are determined by the unit cost and number of jobs that can be completed in a relatively short period of time. Regardless of the type of labor engaged in, the longer the time spent in labor, the more the worker's income is expected to increase roughly proportionally. Second, the amount of players' ability resources does not affect the amount of compensation for simple work. In fact, many managers do not require part-time workers to have special job performance skills [30]. Third, the income of players who engage in simple labor is unstable. When economic conditions are favorable, the income of simple workers is stable, but when economic conditions worsen, simple workers' jobs are reduced or they are laid off [31]. Fourth, players' ability resources do not increase when the players engages in simple work. Many firms keep the training costs they pay for part-timers to a minimum [30]. Finally, workers have the flexibility to adjust their working hours to engage in simple labor. Workers who earn money from one-time jobs, such as on-call work or gig work, and part-time workers can flexibly manage their working hours [32].

Learning (LN) is the act of taking extra time to develop competencies in order to nurture one's work capacity. People who hold core positions in organizations and those who do business with expertise as freelancers engage in lifelong learning to develop the professional skills needed to maintain their employment status [33]. Self-employed individuals with unstable incomes devote more time to work-related learning than employees of firms [28].

Overload (OL) is an activity that involves an involuntary phenomenon in which agents automatically lose health resources in return for choosing to engage in these activities (PW, FW, SW, and LN). In general, overwork in labor and study impairs human health, which in turn leads to lower labor productivity and reduced effectiveness in learning. OL is not an activity to which the player directly allocates resources, but executes automatically as a consequence of the allocation of resources to activities (PW, FW, SW, and LN).

Leisure (LS) is an activity that workers engage in to restore their physical and mental health. In general, workers use their leisure time for rest and recuperation to maintain their health condition.

Based on the characteristics of each activity described above, equations (1) through (8) were designed to describe the relationship between input and output resources for the activities in the Shin-Life Career Game version 2 [12]. The following sections describe the characteristics of the changes in each output.

The amount of remuneration for permanent work is considered to increase monotonically in proportion to the product of the length of working hours, the amount of basic wage, and the level of a worker ' s ability (corresponding to Equation (1)). We set up Equation (1) from the following idea: the higher the basic wage, the more a employer pay permanent workers in his/her organization; the higher the level of permanent workers' professional competence, the more they earn; the longer permanent workers work, the higher their income is; and permanent workers are less likely to be dismissed if their health suffers unlike other types of work (as they are protected by law, and therefore their health status does not affect their income).

The ability of permanent workers is considered to increase monotonically in proportion to the product of the length of the working hours, the level of ability, the learning effect per working time, and the degree of influence from health conditions (corresponding to Equation (2)). We set up Equation (2) from the following idea: the better permanent workers are at learning through engagement in their work, the faster they grow; the higher the competence level of permanent workers, the more they grow on the job; the longer permanent workers are on the job, the more experience and educational opportunities they have, and the more they grow; and the worse permanent workers' health, the slower their growth.

The amount of remuneration for freelance work increases monotonically in proportion to the product of the length of working hours, the degree of influence from the worker's ability, the amount of the basic compensation per working time, health condition, but there is uncertainty in the income side. (corresponding to Equation (3)). We set up Equation (3) from the following idea: freelance workers earn more if they take on jobs with higher base compensation; if freelance workers have higher job skills, they can get higher-paying jobs and their income goes up; the more time freelance workers spend on a job, the more income they earn; freelance workers may not earn enough money due to unforeseen problems at work, or maybe lucky enough to get a job that pays well; and when freelance workers' health deteriorates, they cannot work efficiently and their income decreases.

The amount of remuneration for simple work is considered to increase monotonically in proportion to the product of hourly pay rates, the length of working hours, and the degree of influence from the state of health (corresponding to Equation (4)). We set up Equation (4) from the following idea: simple workers earn income more by working with a good basic wage; simple workers earn more if they work longer; and when simple workers' health deteriorates, they cannot work efficiently and their income decreases.

The degree of growth of a worker's ability is considered to increase monotonically in proportion to the product of the length of learning time, the learning effect per learning time, n power of learning cost, the level of working ability, and the degree of influence from health conditions (corresponding to Equation (5)). We set up Equation (5) from the following idea: the efficiency of workers' skill development depends on whether they are good at it or not; money supports the growth of workers (For example, it would be more effective to spend the same amount of time at a preparatory school and receive instruction from a professional teacher than to study for a qualification by oneself. However, the higher the amount invested in education, the more effective the education becomes, but at the same time, the worse the cost effectiveness becomes. The quality of lectures offered by prestigious universities that require students to pay very high tuition fees is not necessarily better than the quality of lectures offered by ordinary universities.); it is easier to grow if a worker's ability at the time of skill development is high (For example, if an expert software engineer and a beginner spend the same amount of time studying, the former will acquire much more knowledge and skills than the latter.); the more time a worker spends on skill development, the more he/she grows; and health status generally affects the efficiency of a worker's performance.

The degree of deterioration in health status is considered to increase monotonically in proportion to the time of the activity (corresponding to Equation (6)). The more a worker devote themselves to work and learning, the worse the worker ' s health becomes. This is expressed in Equation (6).

The degree of recovery of the health state is considered to increase monotonically in proportion to leisure time (corresponding to Equation (7)). The more time a worker take to rest, the better the worker's health becomes. This is expressed in Equation (7).

The degree of influence from the health condition will be influenced by the past health condition after a time delay (corresponding to Equation (8)). It takes a certain amount of time for ordinary people without medical expertise or skills to become aware of the deterioration of their own health condition. Equation (8) expresses this.

The values of each resource variable are updated in Equations (9) through (12). The variables and constants used in the equations are summarized in Table V. Range of values for resource variables is listed in Table VI. Each equation is explained the followings.

The amount of money resource in the latest round is determined by adding the amount of money resource obtained in the latest round to the amount of money resource in the previous round and subtracting the amount of money resource used in the latest round (see Equation (9)). A person's wealth can change depending on his/her financial situation.

The amount of ability resource is determined by adding the increase in ability resources in the latest round to ability resources in the previous round (see Equation (10)). Everyone grows after he/she has learned.

There is an upper limit to the time a worker can spend on activities (see Equation (11)). For any human being, time is finite.

The amount of health resource in the latest round are determined by subtracting the amount of decreasing health resource in the latest round from the amount of health resource in the previous round and adding up the amount of increasing health resource in the latest round (see Equation (12)). Real workers take care of the negative health effects of high-impact activities such as work by resting and treating illness.

$$I_{PW}(t) = c_{PW} \times A(t-1) \times T_{PW}(t) \tag{1}$$

$$G_{PW}(t) = \gamma_H(t) \times cg_{PW} \times T_{PW}(t) \times A(t-1) \tag{2}$$

$$I_{FW}(t) = \gamma_H(t) \times \epsilon_{FW} \times c_{FW} \times T_{FW}(t) \times A(t-1) \tag{3}$$

$$I_{SW}(t) = \gamma_H(t) \times c_{SW} \times T_{SW}(t) \tag{4}$$

$$G_{LN}(t) = \gamma_H(t) \times cg_{LN} \times T_{LN}(t) \times M_{LN}(t)^n \times A(t-1) \tag{5}$$

Symbol	Description
$M(t)$	Player's money resource as of round t
$A(t)$	Player's ability resource as of round t
$T(t)$	Player's time resource as of round t
$H(t)$	Player's health resource as of round t
$I_{PW}(t)$	Reward for PW in round t (money resource)
$I_{FW}(t)$	Reward for FW in round t (money resource)
$I_{SW}(t)$	Reward for SW in round t (money resource)
$G_{PW}(t)$	Reward for PW in round t (ability resource)
$G_{LN}(t)$	Reward for LN in round t (ability resource)
$H_{RCV}(t)$	Reward for LS in round t (health resource)
$H_{BRDW}(t)$	Penalty for activities (except LS) in round t (health resource)
$M_{LN}(t)$	Money allocated to LN in round t by a player (money resource)
$T_{PW}(t)$	Time spent working as a permanent worker in round t (time resource)
$T_{FW}(t)$	Time spent working as a freelance worker in round t (time resource)
$T_{SW}(t)$	Time spent working as a simple worker in round t (time resource)
$T_{LN}(t)$	Time spent developing ability in round t (time resource)
$T_{LS}(t)$	Time spent recovering health in round t (time resource)
$\epsilon_{FW}(t)$	A random number generated according to a continuous distribution whose probability density function is constant on a finite interval (α,β) and zero outside the interval.
$\gamma_H(t)$	Influence of health status on performance of each activity ($0 \leq \gamma_H(t) \leq 1$)
T_{MAX}	Initial value of a player's time resource (time resource, constant)
H_{MAX}	Initial value of a player's health resource (health resource, constant)
c_{PW}	Reward per unit time for PW (constant)
c_{FW}	Reward per unit time for FW (constant)
c_{SW}	Reward per unit time for SW (constant)
cg_{PW}	Growth per unit time for PW (constant)
cg_{LN}	Growth per unit time for LN (constant)
c_{BRDW}	Amount of health resources lost per unit of time (constant)
c_{RCV}	Amount of health resources recovered per unit time (constant)

TABLE V: Variables and constants composing Equations (1)-(12) (Reprinted from [12] and modified the table)

$$H_{BRDW}(t) = c_{BRDW} \times (T_{PW}(t)+T_{FW}(t)+T_{SW}(t)+T_{LN}(t)) \tag{6}$$

$$H_{RCV}(t) = c_{RCV} \times T_{LS}(t) \tag{7}$$

$$\gamma_H(t) = \begin{cases} 1 & (1 \leq t \leq 2) \\ \frac{H(t-2)}{H_{MAX}} & (2 < t \leq 40) \end{cases} \tag{8}$$

$$M(t) = M(t-1) + I_{PW} + I_{FW} + I_{SW} - M_{LN} \tag{9}$$

$$A(t) = A(t-1) + G_{PW} + G_{LN} \tag{10}$$

Range of values
$0 \leq M(t)$
$0 \leq A(t)$
$0 \leq T(t) \leq T_{MAX}$
$0 \leq H(t) \leq H_{MAX}$

TABLE VI: Range of values for resource variables

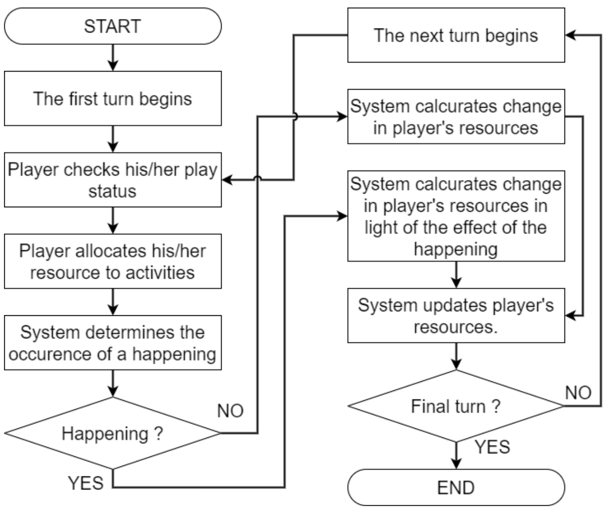


Fig. 2: Flowchart for gameplay (Reprinted from [12])

$$T(t) = T_{MAX} - (T_{PW} + T_{FW} + T_{SW} + T_{LN} + T_{LS}) \quad (11)$$

$$H(t) = H(t-1) + H_{RCV}(t) - H_{BRDW}(t) \quad (12)$$

3) Unexpected Happenings: The Shin-Life Career Game version 2 implements an event called an economic depression event. An economic depression event is a phenomenon designed based on real-world global financial crises such as the Lehman Bankruptcy and the Pandemic Shock. A player may experience an economic depression event. The occurrence of an economic depression event is unpredictable and occurs randomly in each round with a predetermined probability. When an economic depression event occurs, it affects the amount of resources the player allocates to various types of labor. The amount of time resources allocated to PW, FW, and SW is forcibly reset to the minimum amount, and all surplus time resources are allocated to LS. In this study, the minimum amount of time resources that a player can allocate to PW is seventy, and the minimum amount of time resources that a player can allocate to FW and SW is zero.

4) Instructions for Gameplay: In this section, we describe the procedure for playing the game (see Figure 2), referring to the description of the Shin-Life Career Game version 2. In each round, a player make decisions and allocate resources to activities. First, when a round starts, the player has the opportunity to check the game information (history of the amount of each resource, of the amount of resources allocated, and of the occurrence of economic recession events). At this phase, the player considers the policy of resource allocation for the current round and subsequent rounds. Next, the player allocates resources to each activity. After completing the screen operations for resource allocation, the game system updates the game information. At this time, if an economic depression event occurs, the effects of the event are reflected in the game information. Finally, the player is notified of the updating game information, and the round ends. Thereafter, the above procedure is repeated until the player has experienced all rounds.

III. DEA

DEA, proposed by Charnes, Cooper, and Rhodes in 1978, is an approach comparing the efficiency of organizational units such as local authority departments, schools, hospitals, shops, bank branches, and similar instances where there is a relatively homogeneous set of units[4]. In an analysis using the DEA, the observed input-output vectors are enveloped as tightly as possible using linear programming to measure the efficiency of decision-making units (DMUs) and compare the efficiency of DMUs with each other. As a result, each DMU is characterized by a reference set consisting of DMUs with more optimal efficiency. In addition, the envelope generated by connecting the reference sets reveals the relative positions of each DMU. In addition to empirical studies of organizational efficiency, the DEA has been applied to simulation studies such as optimization of production systems (e.g., [34], [35], [36], [37]).

In the following, we describe CCR model, which is the basic model of the DEA. In the analysis using CCR model, the following procedure is followed: first, the input and output of each DMU is observed and the data is collected; second, the weights of the input and output vectors of each DMU are optimized to obtain the efficiency; thirdly, which is then compared among DMUs; and finally, the inefficient DMUs are characterized by a reference set of DMUs with higher efficiency. One of the advantages of the DEA is that it automatically solves the problem of weighting when measuring the efficiency of multi-input multi-output systems.

In the following explanation, we denote the N DMUs to be evaluated by $DMU_j (j = 1, 2, \ldots, N)$. DMU_j has M input variables $\mathbf{x_{mj}}$ and L output variables $\mathbf{y_{lj}}$ that have already been observed. When these are described by matrix equations, they can be regarded as data stored in an M by N matrix X and an L by N matrix Y (see equations (13) and (14)).

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{bmatrix} \quad (x_{mj} > 0 \quad (\forall j, \forall m)) \quad (13)$$

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1N} \\ y_{21} & y_{22} & \cdots & y_{2N} \\ \vdots & \vdots & & \vdots \\ y_{L1} & y_{L2} & \cdots & y_{LN} \end{bmatrix} \quad (y_{lj} > 0 \quad (\forall j, \forall l)) \quad (14)$$

First, we briefly define the efficiency of the simplest input-output system with one type of input data and one type of output data; the efficiency when $M = 1$ and $L = 1$

is expressed by the following equation, and the system is evaluated as efficient in the order of increasing efficiency value.

$$Efficiency = \frac{output}{input}$$

More general input-output systems are characterized by multiple input terms, output terms, or both. The efficiency under such complex conditions is defined as follows

$$Efficiency = \frac{weighted\ sum\ of\ outputs}{weighted\ sum\ of\ inputs}$$

In the above definition, a set of weights should be defined, but this is difficult. In contrast, in CCR model, the efficiency of DMU_j, Θ_j, is formulated as the following fractional programming problem. By solving this fractional programming problem, the efficiency score of DMU_j can be easily obtained.

$<FP_j>$

$$\text{maximize} \quad \Theta_j = \frac{\sum_{l=1}^{L} u_l y_{lj}}{\sum_{m=1}^{M} v_m x_{mj}} \tag{15}$$

$$\text{subject to:} \quad \frac{\sum_{l=1}^{L} u_l y_{lj}}{\sum_{m=1}^{M} v_m x_{mj}} \leq 1 (j = 1, 2, \ldots, N) \tag{16}$$

$$v_m \geq 0 \quad (m = 1, 2, \ldots, M) \tag{17}$$

$$u_l \geq 0 \quad (l = 1, 2, \ldots, L) \tag{18}$$

In this case, since the weights $\mathbf{u_j}$ and $\mathbf{v_j}$ can be moved in any way, the value of the efficiency Θ_j also changes depending on the values of the weights $\mathbf{u_j}$ and $\mathbf{v_j}$, and is not uniquely determined. Therefore, after commonizing the upper limit of efficiency Θ_j at 1 ($\Theta_j \leq 1$), we select the weights $\mathbf{u_j}$ and $\mathbf{v_j}$ that maximize the value of efficiency Θ_j for each DMU_j. If DMU_j is efficient compared to other $DMUs$, the value of Θ_j is 1; conversely, if it is not efficient, it takes a value smaller than 1.

Here suppose that DMU_o is inefficient. Let the optimal weight of the input of DMUo is v^*, the one of the output is u^*, and the input-output efficiency calculated from v^* and u^* is Θ^*. If DMU_j ($j \neq o$) for which the left and right sides of the equation (19) are equal exists, DMU_j is called the reference set of DMU_o. This reference set E_o can be used as a reference to improve the efficiency of DMU_j.

$$E_o = \left\{ j \middle| \sum_{l=1}^{L} u_l^* y_{lj} = \sum_{m=1}^{M} v_m^* x_{mj}, j = 1, 2, \ldots, N, j \neq o \right\} \tag{19}$$

The optimization of efficiency in DEA that we have described so far is formulated as a fractional programming problem (see equations (15) through (18)). However, in practice, it is solved by converting it into an equivalent linear problem (see equations (20) through (24)).

$<LP_j>$

$$\text{maximize} \quad \Theta_j = \sum_{l=1}^{L} u_l y_{lj} \tag{20}$$

$$\text{subject to:} \quad \sum_{m=1}^{M} v_m x_{mj} = 1 \tag{21}$$

$$\sum_{l=1}^{L} u_l y_{lj} \leq \sum_{m=1}^{M} v_m x_{mj} (j = 1, 2, \ldots, N) \tag{22}$$

$$v_m \geq 0 \quad (m = 1, 2, \ldots, M) \tag{23}$$

$$u_l \geq 0 \quad (l = 1, 2, \ldots, L) \tag{24}$$

One of the advantages of using the DEA is that the problem of how to determine weights when comparing the efficiency of multiple-input multiple-output systems is automatically solved in the optimization process described above.

In the following, we will discuss DEA as a data classification method. As a methodology for data classification, DEA takes a different approach from distance-based classification methods such as cluster analysis.

In DEA, the reference set E of an inefficient DMU is a target that should be referred to in order to improve the efficiency of the DMU. Also, the reference set of an efficient DMU is itself. If we draw a scatter plot of DMUs in the space of input/output variables, the envelope of DMUs is formed by all the efficient DMUs (see Figure 3). The smallest convex region containing the efficient DMUs and the origin also contains the inefficient DMUs whose reference set is the efficient DMUs. Therefore, this set of DMUs is considered to share the same reference set and is classified into the same group.

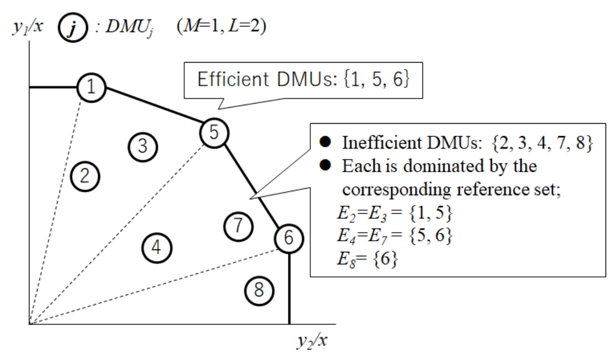


Fig. 3: Dividing the envelope by efficient DMUs (reference set) allows classification of inefficient DMUs by common reference sets. (Reprinted from the study by Kunigami et al.[38])

Kunigami et al. list two advantages of using the DEA as a classifier [38]. One of the advantages of using the DEA as a classifier is that the superiority and inferiority relationships between DMUs belonging to a certain group can be extracted mechanically and endogenously. When an analyst uses the DEA to optimize the efficiency of each DMU that has the same reference set, a set of DMUs with weight sets that are

Agent ID	Action #	Orientation	Allocated Resources †,*,**					Outcome‡		
			T_{PW}	T_{FW}	T_{SW}	T_{LN}	T_{LS}	M	A	H
A1	0	Money-making	RD^1_{PW}			Rest^2	Rest^2	+	(+)	+/-
A2	1			RD^1_{FW}		Rest^2	Rest^2	+	(+)	+/-
A3	2				RD^1_{SW}	Rest^2	Rest^2	+	(+)	+/-
A4	3		RD^1_{PW},	RD^1_{FW}, or	RD^1_{SW}	Rest^2	Rest^2	+	(+)	+/-
B1	4	Ability Development	RD^1_{PW}			Rest^2	RD^1_{LS}	(+)	+	+/-
B2	5			RD^1_{FW}		Rest^2	RD^1_{LS}	(+)	+	+/-
B3	6				RD^1_{SW}	Rest^2	RD^1_{LS}	(+)	+	+/-
B4	7		RD^1_{PW},	RD^1_{FW}, or	RD^1_{SW}	Rest^2	RD^1_{LS}	(+)	+	+/-
C1	8	Enjoying Leisure	RD^1_{PW}			RD^1_{LN}	Rest^2	(+)	(+)	+/-
C2	9			RD^1_{FW}		RD^1_{LN}	Rest^2	(+)	(+)	+/-
C3	10				RD^1_{SW}	RD^1_{LN}	Rest^2	(+)	(+)	+/-
C4	11		RD^1_{PW},	RD^1_{FW}, or	RD^1_{SW}	RD^1_{LN}	Rest^2	(+)	(+)	+/-

† RD_{PW}, RD_{FW}, RD_{SW}, RD_{LN}, and RD_{LS} are randomly determined in increments of 10, from 70 to 100, 50 to 100, 90 to 100, 50 to 100, and 50 to 100 respectively.

‡ ”M” stands for money resource, ”A” for ability resource,and ”H” for health resource.”+” means resource increase, ”-” means resource decrease, and ”+/-” means one of the two can happen. ”(+)” means resource may increase.

* ”Rest” is the maximum value of time resources available to the player in each round minus the total value of time resources allocated to other activities. For example, if software agent A1 first decides the amount of time resources to allocate to PWs, and then decides to allocate resources to LNs, all remaining time resources are allocated to LNs.

** The player performs resource allocation in two steps. First, the player executes the resource allocation action^1; if there are two actions with ^{1}, the player randomly executes one of them. Next, the player executes resource allocation action^2. If there are two actions marked with ^{2}, the player executes one of them at random.

TABLE VII: List of Resource Allocation Actions to Be Selected by Software Agents

similar in composition to each other is mechanically output. As a result, a relation representing the dominance of efficient DMUs belonging to a certain group (which is the reference set and has the optimal weight set) and other inefficient DMUs is endogenously extracted. This is an advantage in using the DEA as a classifier.

Another advantage of using the DEA as a classifier is that it is easy to determine the similarity of groups. The number of overlapping DMUs between DMUs belonging to two reference sets indicates the high degree of similarity between the two groups. The more reference sets in common between the two groups, the closer the two groups are.

IV. Demonstration

In this section, we explain how to conduct a demonstration experiment.

A. Gaming simulation

For this demonstration experiment, we decided to use the Shin-Life Career Game version 2. This game is a gaming simulation with a multi-input multi-output system. See Section 2 for details of the game.

B. Analysis Tools

To analyze the play logs output from the gaming simulation, we use CCR model, which is the basic model of DEA; for more information on DEA, see Section 3.

C. Participants

We made twelve software agents play the Shin-Life Career Game version 2 in place of human players. Each software agent chooses the same resource allocation rule for all rounds. The characteristics of each software agent’s play policy and resource allocation rules are described below. In addition, Table VII summarizes the characteristics of the behavioral rules of the software agents.

Software agents A1, A2, A3, and A4 always choose money-making oriented behavioral rules. They always allocate resources to labor every round, and allocate the extra time resources to either learning or leisure. A1, A2, and A3 always engage only in a specific type of labor, while A4 randomly chooses one of the three types of labor in each round. Second, software agents B1, B2, B3, and B4 always choose growth-oriented behavioral rules. These software agents always allocate resources to learning each round, and allocate the surplus resources to either labor or leisure. When software agents B1, B2, and B3 choose to allocate resources to labor, they always engage in a specific labor only, while B4 randomly chooses one labor among the three. Finally, the software agents C1, C2, C3, and C4 always choose health-oriented behavioral rules. When the software agents C1, C2, and C3 choose to allocate resources to labor, they always engage in only one specific labor, whereas C4 randomly chooses one labor from among the three. These software agents always allocate resources to leisure each round, and allocate the surplus resources to either

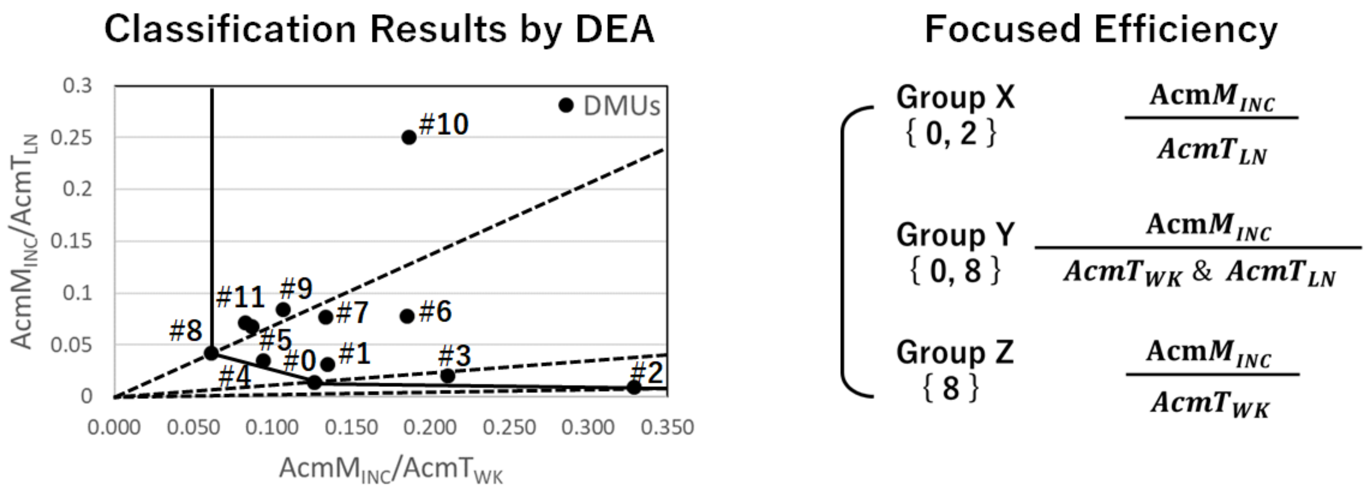


Fig. 4: Classification Results

labor or learning.

D. System

The software for gaming is written in python. The gaming simulation was carried out using Intel(R) core(TM) 4600U CPU @ 2.10GHz PC with 16GB RAM, Windows10 Pro, 64bit OS.

E. Procedures

The software agent was given 40 opportunities (rounds) to make a decision per gaming session. For each gaming session, the incidence of economic recession was fixed at 10 percent. In all gaming rounds, a common random seed and random number generator were used to determine the choice of activities for resource allocation by each software agent, the amount of resources allocated to each activity, whether an economic depression event occurred, and the FW reward correction rate. Each software agent played the game one time. The initial values of the parameters common to all gaming are listed in Table VIII.

Parameter	Initial Value	Parameter	Initial Value
$M(0)$	0	cg_{PW}	2.0E-4
$A(0)$	1	cg_{LN}	2.0E-4
T_{MAX}	100	c_{RCV}	2.8E-1
H_{MAX}	100	c_{BRDW}	5.0E-2
c_{PW}	5.5	n	0.50
c_{FW}	5.5	α	2.5E-1
c_{SW}	5.5	β	2.0

TABLE VIII: Initial values of parameters

V. Results

The input and output data of each software agent are summarized in Table IXa. Basic statistics of input and output are summarized in Table IXb. Note that $AcmT_{WK}$ represents the accumulated value of time resources allocated to labors, $AcmT_{LN}$ represents the accumulated value of time resources allocated to learning, $AcmT_{LS}$ represents the accumulated value of time resources allocated to leisure, and $AcmINC$ represents the accumulated value of money resources acquired. In addition, a DEA tool pyDEA [39][40] was used to generate the following DEA results (See Table IXc).

According to Table IXc, the most efficient software agents (DMUs) were DMU#0 and DMU#8. The DEA also classified the play logs into Group X, Group Y, and Group Z. Group X consisted of only DMU#3 with DMU#0 and DMU#2 as the reference set. Group Y consisted of DMU#1, DMU#4, DMU#6, and DMU#7 with DMU#0 and DMU#8 as the reference set. Group Z consisted of DMU#5, DMU#9, DMU#10, and DMU#11 with DMU#8 as the reference set. The software agents (DMUs) belonging to each group are summarized in Table IXd.

The results of the analysis using DEA (Table IXc) are schematically summarized in Figure 4. As a result of the analysis by DEA, the value of the weight of the accumulated value of time resources allocated to leisure for all play logs was zero (this means that this factor has no effect on the evaluation of the efficiency of each play log.), so the data was plotted on a two-dimensional plane (see Figure 4). In Figure 4, multiple groups consisting of efficient DMUs (reference sets) and inefficient DMUs that refer to them are depicted. The similarity between the groups was easily determined by whether or not the reference sets were shared. Specifically, it was found that the similarity between Group X and Group Y, which share the common reference set DMU#0, is higher than the similarity between Group X and Group Z. Similarly, we found that the similarity between Group Y and Group Z with the common reference set, DMU#8, was higher than the similarity between Group X and Group Z. Therefore, we also found that Group Y is the group that was in the middle of Group X and Group Z.

DMU #	Agent ID	Gaming Result[†]			
		INPUT			OUTPUT
		$AcmT_{WK}$	$AcmT_{LN}$	$AcmT_{LS}$	$AcmINC$
0	A1	3.39.E+03	3.60.E+02	2.50.E+02	2.67.E+04
1	A2	2.62.E+03	6.10.E+02	7.70.E+02	1.94.E+04
2	A3	3.61.E+03	1.00.E+02	2.90.E+02	1.10.E+04
3	A4	3.29.E+03	3.20.E+02	3.90.E+02	1.56.E+04
4	B1	1.89.E+03	7.00.E+02	1.41.E+03	2.01.E+04
5	B2	1.23.E+03	9.50.E+02	1.82.E+03	1.41.E+04
6	B3	1.41.E+03	5.90.E+02	2.00.E+03	7.60.E+03
7	B4	1.32.E+03	7.60.E+02	1.92.E+03	9.86.E+03
8	C1	1.84.E+03	1.28.E+03	8.80.E+02	3.01.E+04
9	C2	1.63.E+03	1.28.E+03	1.09.E+03	1.52.E+04
10	C3	1.31.E+03	1.76.E+03	9.30.E+02	7.04.E+03
11	C4	1.64.E+03	1.41.E+03	9.50.E+02	1.97.E+04

[†] $AcmT_{WK}$ and $AcmT_{LN}$ are the accmulated value of time resources allocated to works (PW, FW, and SW) and learning, respectively. $AcmINC$ is the accumulated value of income (money resource).

(a) Gaming Results (Input-output data)

$Statics$	$AcmT_{WK}$	$AcmT_{LN}$	$AcmT_{LS}$	$AcmINC$
Mean	2.10.E+03	8.43.E+02	1.06.E+03	1.64.E+04
S.D.	8.86.E+02	5.01.E+02	6.16.E+02	7.22.E+03
Variance	7.85.E+05	2.51.E+05	3.79.E+05	5.21.E+07
Median	1.74.E+03	7.30.E+02	9.40.E+02	1.54.E+04
Max	3.61.E+03	1.76.E+03	2.00.E+03	3.01.E+04
Min	1.23.E+03	1.00.E+02	2.50.E+02	7.04.E+03

(b) Basic statistics of input/output data

DMU #	Agent ID	DEA Score Θ	Reference sets	Weights			
				INPUT			OUTPUT
				$v_{T_{WK}}$	$v_{T_{LN}}$	$v_{T_{LS}}$	u_I
0	A1	1.00	{0, 8}	2.38.E-04	5.38.E-04	0.00.E+00	3.74.E-05
1	A2	0.765	{0, 8}	2.50.E-04	5.66.E-04	0.00.E+00	3.93.E-05
2	A3	1.00	{0, 2}	1.21.E-04	5.63.E-03	0.00.E+00	9.12.E-05
3	A4	0.646	{0, 2}	5.51.E-05	2.56.E-03	0.00.E+00	4.15.E-05
4	B1	0.910	{0, 8}	2.88.E-04	6.51.E-04	0.00.E+00	4.53.E-05
5	B2	0.700	{8}	8.13.E-04	0.00.E+00	0.00.E+00	4.97.E-05
6	B3	0.436	{0, 8}	3.64.E-04	8.24.E-04	0.00.E+00	5.73.E-05
7	B4	0.510	{0, 8}	3.29.E-04	7.44.E-04	0.00.E+00	5.18.E-05
8	C1	1.00	{0, 8}	2.11.E-04	4.78.E-04	0.00.E+00	3.32.E-05
9	C2	0.571	{8}	6.13.E-04	0.00.E+00	0.00.E+00	3.75.E-05
10	C3	0.328	{8}	7.63.E-04	0.00.E+00	0.00.E+00	4.67.E-05
11	C4	0.736	{8}	6.10.E-04	0.00.E+00	0.00.E+00	3.73.E-05

(c) Analysis results by DEA

Groups	Reference sets	DMUs
X	{0, 2}	3
Y	{0, 8}	1, 4, 6, 7
Z	{8}	5, 9, 10, 11

(d) List of software agents (DMUs) belonging to each group

TABLE IX: Input-output data and basic statistics, and results of analysis by DEA

VI. Discussions

As a result of DEA analysis of the play logs obtained from the Shin-Life Career Game version 2, the play logs were classified into three groups, and the reference sets and DMUs composition of each group were revealed. The reference sets of each group can be used as a reference to improve the efficiency of inefficient DMUs belonging to the same group. In the following, we will clarify the structure of the evaluation criteria for each group and discuss what methods can be used to improve the evaluation of inefficient DMUs belonging to each group.

First, we discuss the experimental results for Group X. The value of the weights of $AcmT_{WK}$ ($v_{AcmT_{WK}}$) and of $AcmT_{LN}$ ($v_{AcmT_{LN}}$) of DMU#3 (inefficient DMU belonging to Group X) are both positive. However, the value of the weight of the former is about 2% of the one of the latter, and the effect of $v_{AcmT_{WK}}$ on the evaluation criteria is small. Thus, it was found that in Group X, DMUs were evaluated based on the efficiency of the accumulated income amount (output) relative to the accumulated learning time (input).

The criterion for evaluating the efficiency of DMU#2, the one of reference sets of group X, was similar to that of DMU#3. As in the case of DMU#3, the value of $v_{AcmT_{WK}}$ and $v_{AcmT_{LN}}$ in DMU#2 were both positive. However, the former was only about 2% as large as the latter. Therefore, the effect of $AcmT_{WK}$ is very small. In other words, DMU#2 was evaluated by DEA to be optimal in terms of its efficiency under the same evaluation criteria as Group X.

In contrast, the evaluation criteria for DMU#0, the other reference set of Group X, was different from the one for DMU#2. For DMU#0, the value of $v_{AcmT_{WK}}$ was about 44% of $v_{AcmT_{LN}}$. In other words, the effect of $AcmT_{WK}$ on AcmINC for DMU#0 was larger than the one for DMU#2. Thus, DMU#0 was highly valued by DEA as having made a lot of money despite the fact that the player had spent less time on working and learning.

Given the above discussion on Group X, we found that there were two options to improve the efficiency of inefficient DMU#3 belonging to Group X. The first option is to learn from the behavior of DMU#2 that performed better without changing its current policy on resource allocations. The second option is to modify the current policy on resource allocations by referring to DMU#0 and change it to one that takes into account the allocation of resources to both work and learning.

Next, we discuss the experimental results for Group Y. The values of $v_{AcmT_{WK}}$ and $v_{AcmT_{LN}}$ of the DMUs belonging to Group Y were both positive. The value of the former was about 44% of the one of the latter, indicating that the influence of $AcmT_{WK}$ and $AcmT_{LN}$ on $AcmINC$ was not negligible in both factors. Therefore, it was found that in Group Y, DMUs were evaluated based on the efficiency of $AcmINC$ with respect to $AcmT_{WK}$ and $AcmT_{LN}$. In other words, in Group Y, DMUs that earned a lot of money while spending less time working and learning through gaming were highly valued.

In contrast, the efficiency metric of DMU#8, the one of reference sets of Group Y, is different from that of Group Y. For DMU#8, the value of $v_{AcmT_{LN}}$ is zero, while the value of $v_{AcmT_{WK}}$ is positive. Thus, DMU#8 was highly valued because he made a lot of money through gaming, even though he did not spend much time on labor.

Given the above discussion on Group Y, we found that there were two options to improve the efficiency of inefficient DMUs belonging to Group Y. The first option is to learn from the behavior of DMU#0 that performed better without changing the current course of action. The second option is to change the current course of action by referring to DMU#8 and orienting to a more active allocation of resources to labor.

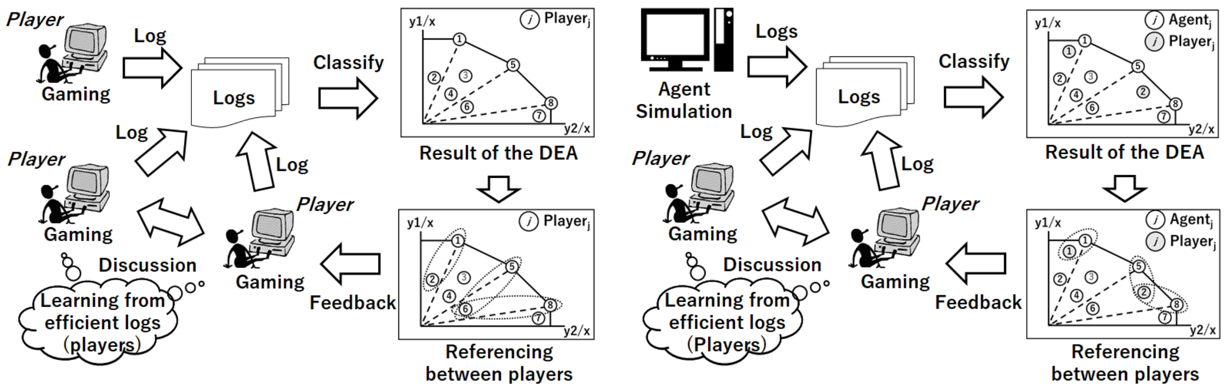
Finally, we discuss the experimental results for Group Z. The value of $v_{AcmT_{WK}}$ for DMUs belonging to Group Z was positive, while the value of $v_{AcmT_{LN}}$ was zero. Therefore, we knew that in Group Z, DMUs were evaluated based on the high efficiency of $AcmINC$ relative to $AcmT_{WK}$. In other words, in Group Z, DMUs that earned a lot of money through gaming, even though they spent less time on labor, were highly valued.

From the above discussion, by using DEA to analyze the 12 play logs output from the gaming simulation, it became clear which play logs were superior to other play logs under the same evaluation criteria. Based on the extracted knowledge, the facilitator can develop instruction that encourages players to improve their behavior. In addition, players can use the facilitator's guidance to narrow down the behaviors and performances of others in the game that they need to learn, and learn more efficiently (See Figure 5a).

As shown in the explanation of the example above, by applying DEA to the analysis of the play logs output from a gaming simulation, multiple play logs are automatically classified under various criteria, and the superiority or inferiority of the play logs within each group is revealed. Based on the above knowledge obtained from the analysis, facilitators will be able to provide specific guidance to players in debriefing to improve their behavior. And at the same time, the players will be able to focus on specific play logs to learn effectively and efficiently (see Figure 5a and Figure 5b). In order to propose a new analysis methodology, we have presented an example of analysis using a gaming simulation, the Shin-Life Career Game version 2, and DEA, but this analysis methodology can be applied to any gaming simulation that its players operate a multiple-input, multiple-output system, and the tools used in the analysis are not limited to DEA. In addition, this analytical methodology can be applied to both gaming simulations consisting only of humans and gaming simulations in which humans and software agents are mixed (see Figure 5a and Figure 5b).

VII. Concluding Remarks

In this study, we proposed a methodology to extract knowledge from the results of a gaming simulation that explains which play logs are superior (inferior) to other play logs under a specific objective function. In order to propose the new methodology, we had multiple software agents play the gaming simulation instead of human players, and analyzed



(a) Gaming simulation with only human players **(b)** Gaming simulation with human players and software agents

Fig. 5: Conceptual diagram of the proposed method: Play logs are collected in a gaming session with a large number of players, and DEA is used to identify the superiority or inferiority among the play logs, so that each player has the opportunity to learn from the play logs of players whose behavior is similar to and superior to his own.

the output multiple play logs with DEA to extract the desired knowledge. As a result, the desired knowledge was obtained. Finally, we proposed supporting players' learning of gaming by facilitators' presenting the extracted knowledge to players in debriefing.

In past research, it was pointed out that life simulation games such as the Life Career Game developed by Boocock and the Shin-Life Career Game developed by us may not express human decision-making problems in real life properly [19]. Munson et al. pointed out that the social properties and attitudes of players may affect the results of gaming simulations. Based on Munson et al.'s consideration, we are considering using the Shin-Life Career Game to analyze the relationship between the lifestyles of people who have proven to be successful in real life and the results of their gaming (input-output efficiency). We think that lifestyle of people is a reflection of their policies in their resource allocation in life. This new approach may provide a new answer to the long-standing criticism of gaming simulations.

REFERENCES

[1] A. Sakata, T. Kikuchi, R. Okumura, M. Kunigami, A. Yoshikawa, M. Yamamura, and T. Terano, "The Shin-Life Career Game: Pursuing Your New Life Style through Gaming Simulation," in *eKNOW 2021, The Thirteenth International Conference on Information, Process, and Knowledge Management*, Nice, France, July 2021, pp. 14–20. [Online]. Available: https://www.thinkmind.org/index.php?view=instance&instance=eKNOW+2021[Accessed:2021-12-01]

[2] A. Charnes, W. W. Cooper, and E. Rhodes, "Measuring the efficiency of decision making units," *European journal of operational research*, vol. 2, no. 6, pp. 429–444, 1978.

[3] C. S. Greenblat, *Designing games and simulations: An illustrated handbook*. SAGE Publications Inc., 1988.

[4] A. Boussofiane, R. G. Dyson, and E. Thanassoulis, "Applied data envelopment analysis," *European journal of operational research*, vol. 52, no. 1, pp. 1–15, 1991.

[5] T. Terano, "Learning business decisions through cases and games," *J. Soc. Instrum. Control Eng*, vol. 46, no. 1, pp. 44–50, 2007, (in Japanese).

[6] M. Nakamura, "ゲーミングシミュレーションにおけるファシリテーション (Facilitation in Gaming Simulations)," in *Gaming Simulation*. Union of Japanese Scientists and Engineers (JUSE), 1998, pp. 169–206, (in Japanese).

[7] D. C. Thatcher and M. J. Robinson, "Me—The Slow Learner: Reflections Eight Years on from its Original Design," *http://dx.doi.org/10.1177/1046878190213008*, vol. 21, no. 3, pp. 291–302, aug 2016. [Online]. Available: https://journals.sagepub.com/doi/abs/10.1177/1046878190213008[Accessed:2021-12-01]

[8] T. Kikuchi, Y. Tanaka, M. Kunigami, T. Yamada, H. Takahashi, and T. Terano, "Debriefing framework for business games using simulation analysis," in *General Conference on Emerging Arts of Research on Management and Administration*. Springer, 2018, pp. 64–76.

[9] O. Koshiyama, M. Kunigami, A. Yoshikawa, and T. Terano, "Analyzing Behaviors of Business Game Learners Using a Modified Performance Sheet," *Simulation & Gaming*, vol. 21, no. 2, pp. 86–95, 2011, (in Japanese).

[10] A. Sakata, T. Kikuchi, R. Okumura, M. Kunigami, A. Yoshikawa, M. YAMAMURA, and T. Terano, "Uncovering Users' Decisions through Serious Game Playing with A Formal Description Method," in *Proceedings of the Thirteenth China-Japan International Workshop on Information Technology and Control Applications (ITCA2020)*. Enshi, Hubei, China: 2020 The 13th China-Japan International Workshop on Information Technology and Control Applications, 2020, pp. 221–228. [Online]. Available: https://ci.nii.ac.jp/naid/120006998456[Accessed:2021-12-01]

[11] T. Terano, H. Suzuki, Y. Kuno, H. Fujimori, H. Shirai, C. Nishio, N. Ogura, and M. Takahuashi, "Understanding Your Business through Home-Made Simulator Development," *Developments in Business Simulation and Experiential Learning: Proceedings of the Annual ABSEL conference*, vol. 26, mar 1999. [Online]. Available: https://absel-ojs-ttu.tdl.org/absel/index.php/absel/article/view/930[Accessed:2021-12-01]

[12] A. Sakata, T. Kikuchi, R. Okumura, M. Kunigami, A. Yoshikawa, M. Yamamura, and T. Terano, "A Basic Research to Develop a Method to Classify Game Logs and Analyze them by Clusters," in *the 7th International Workshop on Advanced Computational Intelligence and Intelligent Informatics (IWACIII 2021)*, Beijing, China, 2021.

[13] S. S. Boocock, "The Life Career Game," *The Personnel and Guidance Journal*, vol. 46, no. 4, pp. 328–334, dec 1967.

[14] S. S. Boocock and J. S. Coleman, "Games with Simulated Environments in Learning," *Sociology of Education*, vol. 39, no. 3, p. 215, 1966.

[15] J. Ferrara, "Games for Persuasion: Argumentation, Procedurality, and the Lie of Gamification," *Games and Culture*, vol. 8, no. 4, pp. 289–304, jul 2013. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1555412013496891[Accessed:2021-12-01]

[16] I. Dunwell, P. Lameras, S. De Freitas, P. Petridis, M. Hendrix, S. Arnab, and K. Star, "Metycoon: A game-based approach to career guidance," in *2013 5th International Conference on Games and Virtual Worlds for Serious Applications, VS-GAMES 2013*, 2013.

[17] Tertiary Education Commission, "The Real Game." [Online]. Available: https://www.careers.govt.nz/resources/tools-and-activities/the-real-game/[Accessed:2021-12-01]

[18] I. Tallman, L. R. Wilson, and M. A. Straus, "SIMCAR: A game simulation method for cross-national family research:," *https://doi.org/10.1177/053901847401300207*, vol. 13, no. 2, pp. 121–144, sep 2016. [Online]. Available: https://journals.sagepub.com/doi/abs/10.1177/053901847401300207?journalCode=ssic[Accessed:2021-12-01]

[19] W. Munsen, J. Horan, L. Miano, and C. Stone, "Another look at the life career game," *Pennsylvania Personnel and Guidance Association Journal*, vol. 4, pp. 36–38, 1976.

[20] R. D. Duke, *Gaming: the future's language*, 1st ed. SAGE Publications Inc., 1974.

[21] M. Kunigami, R. Okumura, T. Kikuchi, A. Sakata, and T. Terano, "Agents' Resources Allocation and Growth Modeling for Social and Organizational Agent Simulations," in *JSAI Special Interest Group on Business Informatics (SIG-BI #17)*, Online, 2021, pp. 55–58, (in Japanese). [Online]. Available: http://sig-bi.jp/meeting_17.html[Accessed:2021-12-01]

[22] Statistics Bureau of Japan, "Monthly results," https://www.stat.go.jp/english/data/roudou/index.html[Accessed: 2021-12-01], 2021, (in Japanese).

[23] Statistics Bureau of Japan, "労働力調査（詳細集計）2020 年 平均結果 (Labour Force Survey (Detailed Tabulation) 2020 Average Results)," https://www.stat.go.jp/data/roudou/sokuhou/nen/dt/[Accessed: 2021-12-01], 2020, (in Japanese).

[24] H. Kakizawa and O. Umezaki, "Effect of performance appraisal, compensation, and job characteristics on employees' motivation: an empirical analysis of firm microdata and a questionnaire survey," *JILPT Research Report*, vol. 52, no. 5, pp. 67–82, 2010, (in Japanese).

[25] Y. Zhu, "Responding to the challenges of globalization: human resource development in japan," *Journal of World Business*, vol. 39, no. 4, pp. 337–348, 2004.

[26] C. Brown and M. Reich, "Developing skills and pay through career ladders: Lessons from japanese and us companies," *California Management Review*, vol. 39, no. 2, pp. 124–144, 1997.

[27] S. Horowitz, S. Buchanan, M. Alexandris, M. Anteby, N. Rothman, S. Syman, and L. Vural, "The rise of the freelance class: a new constituency of workers building a social safety net," 2005.

[28] A. G. Kazi, R. M. Yusoff, A. Khan, and S. Kazi, "The freelancer: A conceptual review," *Sains Humanika*, vol. 2, no. 3, 2014.

[29] J. Kirchner and K. Wilhelm, "Employee or freelance worker," in *Key Aspects of German Employment and Labour Law*. Springer, 2018, pp. 45–52.

[30] C. Inman and C. Enz, "Shattering the myths of the part-time worker," *Cornell Hotel and Restaurant Administration Quarterly*, vol. 36, no. 5, pp. 70–73, 1995.

[31] Cabinet Office, "日本経済 2010 (Japan Economy 2010)," https://www5.cao.go.jp/keizai3/2010/1210nk/nk10.html[Accessed: 2021-12-01], 2021, (in Japanese).

[32] J. Rubery, A. Keizer, and D. Grimshaw, "Flexibility bites back: the multiple and hidden costs of flexible employment policies," *Human Resource Management Journal*, vol. 26, no. 3, pp. 235–251, 2016.

[33] R. PRESTON and C. DYER, "Human capital, social capital and lifelong learning: an editorial introduction," *Compare: A Journal of Comparative and International Education*, vol. 33, no. 4, pp. 429–436, 2003.

[34] A. Azadeh, T. Nazari, and H. Charkhand, "Optimisation of facility layout design problem with safety and environmental factors by stochastic dea and simulation approach," *International Journal of Production Research*, vol. 53, no. 11, pp. 3370–3389, 2015.

[35] D. Berre, J. Vayssières, J.-P. Boussemart, H. Leleu, E. Tillard, and P. Lecomte, "A methodology to explore the determinants of eco-efficiency by combining an agronomic whole-farm simulation model and efficient frontier," *Environmental Modelling & Software*, vol. 71, pp. 46–59, 2015.

[36] H. Sameie and M. Arvan, "A simulation-based data envelopment analysis (dea) model to evaluate wind plants locations," *Decision Science Letters*, vol. 4, no. 2, pp. 165–180, 2015.

[37] E. Shadkam and M. Bijari, "Multi-objective simulation optimization for selection and determination of order quantity in supplier selection problem under uncertainty and quality criteria," *The International Journal of Advanced Manufacturing Technology*, vol. 93, no. 1, pp. 161–173, 2017.

[38] M. Kunigami, T. Kikuchi, and T. Terano, "A knowledge extraction from epidemic control simulation," *eKNOW 2021, The Thirteenth International Conference on Information, Process, and Knowledge Management*, pp. 21–26. [Online]. Available: https://www.iaria.org/conferences2021/fileseKNOW21/68005_eknow2021.pdf[accessed:2021-12-01]

[39] A. Raith, P. Rouse, and L. Seiford, "Benchmarking using data envelopment analysis: Application to stores of a postand banking business," in *Multiple Criteria DecisionMaking and Aiding*, 2019, pp. 1–39.

[40] A. Raith, O. Perederieieva, F. Fauzi, K. Harton, A. Lee, K. Lin, H. Priddey, and M. Rouse. pyDEA. [Online]. Available: https://github.com/araith/pyDEA[Accessed:2021-12-01]

Educational Applications of Augmented Reality (AR) and Virtual Reality (VR) to Enforce Teaching the "National Academy of Engineering Grand Challenges for Engineering in the 21st Century"

Bibhav Bhattarai
Computer Science & Software Engineering
Auburn University
Auburn, USA
bzb0079@auburn.edu

Daniela Marghitu
Computer Science & Software Engineering
Auburn University
Auburn, USA
marghda@auburn.edu

Abstract — **The National Academy of Engineering's "Fourteen Grand Challenges for Engineering in the Twenty-First Century" identifies challenges in science and technology that are both feasible and sustainable to help people and the planet prosper. Four of these challenges are: advance personalized learning, enhance virtual reality, make solar energy affordable and provide access to clean water. In this work, the authors discuss developing of applications using immersive technologies, such as Virtual Reality (VR) and Augmented Reality (AR) and their significance in addressing four of the challenges. The Drinking Water AR mobile application helps users easily locate drinking water sources inside Auburn University (AU) campus, thus providing easy access to clean water. The Sun Path mobile application helps users visualize Sun's path at any given time and location. Students study Sun path in various fields but often have a hard time visualizing and conceptualizing it, therefore the application can help. Similarly, the application could possibly assist the users in efficient solar panel placement. Architects often study Sun path to evaluate solar panel placement at a particular location. An effective solar panel placement helps optimize degree of efficiency of using the solar energy. The Solar System Oculus Quest VR application enables users in viewing all eight planets and the Sun in the solar system. Planets are simulated to mimic their position, scale, and rotation relative to the Sun. Using the Oculus Quest controllers, disguised as human hands in the scene, users can teleport within the world view, and can get closer to each planet and the Sun to have a better view of the objects and the text associated with the objects. As a result, tailored learning is aided, and Virtual Reality is enhanced. In a camp held virtually, due to Covid-19, K12 students were introduced to the concept and usability of the applications. Likert scales metric was used to assess the efficacy of application usage. The data shows that participants of this camp benefited from an immersive learning experience that allowed for simulation with inclusion of VR and AR.**

Keywords-Augmented Reality; Engineering Challenges; Immersive Technology; Virtual Reality.

I. INTRODUCTION

The "Fourteen Grand Challenges for Engineering in the Twenty-First Century" published by the National Academy of Engineering, lists science and technological challenges that are both attainable and sustainable to help people and the planet prosper [1]. The grand challenges of engineering were announced in 2008 by a committee of leading technological thinkers, considering the fact that the earth's resources are finite and that the world's population is depleting them at an unsustainable rate. These challenges were broadly classified into fourteen game-changing goals. Working towards these goals, as per the committee, is a way for improving life on the planet [2]. This research makes use of immersive technologies to addresses four of such challenges: 1. Advance personalized learning, 2. Enhance virtual reality, 3. Make solar energy affordable, and 4. Provide access to clean water.

AR and VR are two emerging, immersive technologies in recent times. AR creates a composite view by adding digital content to a real-world view, often by using the camera of a smartphone while VR creates an immersive view where the user's view is often cut off from the real world. In AR, users' world views remain intact and virtual objects simply augment the reality, whereas, in VR, users' world views are totally altered, and they can no longer see their actual surroundings.

In this research, a VR application aims to address the first two challenges while two AR applications aim to address the last two challenges. The VR application assists users in visualizing and understanding our solar system by using a VR headset. Users can take an immersive, virtual tour of the solar system. This virtual simulation closely parallels the movements of the planets, as well as their form, scale, and location in relation to the Sun. Thus, this application enables users to view our solar system in an immersive environment, which could be helpful in visualizing and comprehending a system that is not easily observable. The Drinking Water AR application displays information on drinking water accessibility and the environmentally sustainable use of water bottles rather than plastic cups. The application can be used to locate drinking water related information by simply pointing the device camera towards a Point of Interest (POI). Also, it can be used to file and view water-related complaints. Thus, the application helps users to conveniently identify drinking water related information inside Auburn University (AU), thus providing easy access to clean drinking water. The Sun Path AR application helps users visualize Sun's path at a selected date and location. Students study the Sun's path in several areas, but they often fail to visualize and comprehend it. Architects often analyze the sun's path to evaluate the positioning of solar panels at a particular location. An effective solar panel placement helps optimize solar energy cost. Thus, the application could possibly assist the users in efficient solar panel placement.

Empirical studies on the effectiveness of adding mobile game-based augmented reality into basic education suggests that AR techniques can boost student learning [3]. Similarly, there is an AR application to hydrate dementia-affected older adults [4]. The application reminds, inspires, directs, and monitors hydration among those adults. Likewise, students were readily engulfed in AR and their ability to interact with the interface and control virtual objects helped them to understand more advanced concepts of Earth-Sun relationships [5]. All the above-mentioned works in the literature back up this study's argument that AR can help with customized learning, resource access, and visualizing abstract concepts.

In conclusion, the applications serve as a proof of concept for use of immersive technology in addressing engineering concerns. In addition, K-12 students were introduced to the concept and usability of applications at a camp held virtually due to Covid-19. Likert scales metric was used to assess the efficacy of application usage.

In Section II, the paper discusses previous work by other authors related to this research. The project architecture used in the research is then presented in Section III. In Section IV, the paper depicts the usability study of the research. In Section V, the result of the study is reported. Finally, in Section VI, the authors provide conclusions and future work briefings.

II. RELATED WORK

A. AR/VR Modes and Characteristics

Immersive technology blurs the line between the real and virtual worlds, allowing users to feel fully immersed in the experience [6]. Immersive technology leverages a 360 space to either create a new reality or to extend the reality into some degree of virtuality. Virtual environment often shuts down a user from his/her reality and absorbs them into a new, digital environment [7]. As a result, immersive technology immerses users in a simulated world, altering their mental state. Immersive technology can be broadly categorized into four categories: 360, VR, AR, and MR. Virtual reality and augmented reality are two main types of immersive technology. Virtual reality (VR) fully takes over one's vision, giving users the feeling of being transported from the physical world to a virtual one. On the other hand, augmented reality (AR) simply overlays virtual objects onto the user's view of the real world. Milgram's reality-virtuality continuum is a pictorial representation of the spectrum of immersive technologies. Figure 1 depicts the transition and chasm from a real environment to a virtual environment.

Based on the underlying implementation scheme, AR is classified into three different categories: Marker-less AR, Mark-based AR, and Location-based AR. In the same way, VR is classified in 3 Degrees of Freedom (DoF) and 6DoF based on user's degree of freedom.

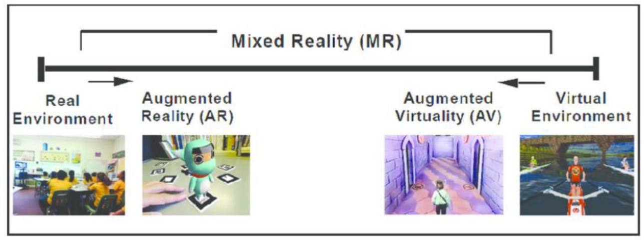


Figure 1. Milgram's reality-virtuality continuum.

Augmented Reality creates a composite view by adding virtual components to users' real view. AR is quite popular these days in various fields such as social media, learning, shopping and so forth. With the advent of Snapchat filters, AR became quite popular in social media. Soon after Facebook too integrated filter-based AR functionalities in many of its applications. Similarly, Ikea has AR features in its shopping application with the help of which customers can pick a product and place it at different points in their world view to see how the virtual product fits in their world view. Likewise, various apps such as Quiver, Blippar and Aurasma use AR to help student with learning [8]. There are basically 3 types of AR: Marker-based AR, Marker-less AR, and Location-based AR.

Marker-based AR uses pre-defined markers set by the developer of the application. When the markers are detected in the real world, virtual objects are augmented to the scene. Markers may be any form of 2D image, including black- and-white and color images. Figure 2 depicts AR content overlay over a pre-defined marker.

Figure 2. Marker based AR [9].

Marker-less AR is not bounded to a particular marker, but rather allows users to position objects anywhere they want within their real-world view. After placing an object, even if the device camera is removed from the line of sight, the application still remembers the position of the object using a method called Simultaneous Localization and Mapping (SLAM), and so when the device is brought back into line of sight the object is once again visible [9]. Figure 3 is an AR enabled retail application by Ikea. It is a marker-less AR app that allows users to place virtual products at desired position before buying them, thus assisting users with product selection and decision-making.

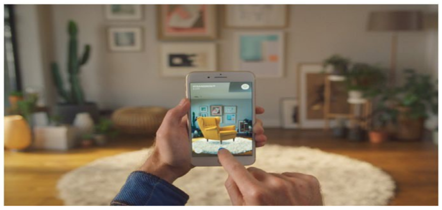


Figure 3. IKEA AR app example [10].

Location-based AR enables the ability to place virtual objects at various GPS coordinates. Location-based AR, in its simplest form, collects data from device components such as GPS, accelerometer and digital compass to identify the device location and position. The application then compares device data to POI information and adds virtual objects to the real environment accordingly [11]. An example of one of the most popular location-based AR apps is Pokémon Go. Figure 4 depicts another location-based AR application where different points of interest objects are overlaid as per their corresponding GPS coordinates.

Figure 4. Location based AR app [11].

VR is an immersive technology that allows users to interact with a virtual environment as if it were the reality. In virtual reality, Head-Mounted Displays (HMDs) are important for bringing the technology to life. An HMD is worn over the head, with the user's world view entirely obscured and only the screen displays visible in front of their eyes. The display supports a stream of data, images, and other such material. Currently, there are several powerful 3DoF and 6DoF HMDs available on the market. Google Cardboard is an example of a 3DoF headset and supports 3DoF (rotational movement around the x, y, and z axes). Similarly, Oculus Quest by Facebook, illustrated in Figure 5, is an example of a 6DoF headset and supports 6DoF (rotational movement around the x, y, and z axes, up, down forward, and backward).

6DoF tracking ensures a higher level of immersion than 3DoF as the user presence is more authentic. Figure 6 illustrates 3DoF and 6DoF tracking.

Figure 5. Oculus Quest headset [12].

B. Applications in Academic Settings

A survey study regarding use of augmented reality provided a scenario in which enabled mobile devices were used for learning and the associated pros and cons of the device usage was evaluated [3]. The questionnaire type survey is based on one single application – EduPARK – which analyzes mobile learning via students' opinion regarding the use of mobile devices for learning. The survey considers a total of 244 students at primary Portuguese Education System. The study participants consisted of students aged 10-16 years old among which 51.6 percent were girls and 48.4 percent were boys. The EduPARK application is designed for a specific urban park in Portugal.

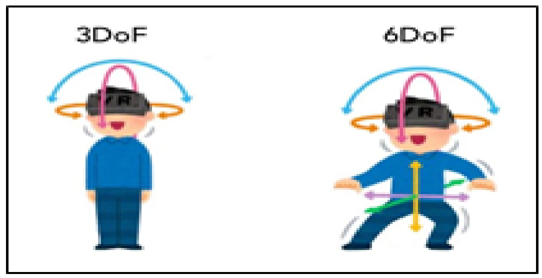


Figure 6. 3DoF and 6DoF [13].

The application uses Augmented Reality (AR) to provide various biological and historical references of the local park. The app was developed in Unity 5 using Vuforia framework and makes use of Vuforia's 2D marker-based technology. The marker-based technology allows the app to detect images/markers, pre-defined by the app creator, and overlay AR contents when the markers are detected by the device camera. As per the paper, the markers were manually installed in either tiles already existing in the park, or on plaques positioned for the purpose of sticking the markers onto them. The authors of the paper weigh in on students' perspective with the application usage. The findings of the paper suggest that the overall perspective remained positive with application usage amongst the students. The study also suggests that students believe that mobile devices, in general, are beneficial when they want to quickly find up-to- date information. However, students had their concerns with some of the external aspects of the application usage such as unstable, slow access to internet connectivity, restrictions forbidding them from carrying mobile devices to the classroom and ease of distraction by other applications inthe mobile device.

All in all, this paper suggests that use of AR mobile applications in learning can be beneficial. A study was done that proposed an AR app that helps cognitively impaired elderly people with hydration [4]. Even though a significant number of older adults are capable of drinking water/fluid by themselves, several cognitive deficits such as poor initiation, decreased motivation, amnesia, and premature decay of intention may hinder their capability [14]. Poor Initiation in older adults is observed when they fail to recall, and this deficit is common in elderly people with dementia [14]. Due to poor initiation, old adults fail to recall where and how to fetch water. Often, older adults have a degraded sense of taste and smell due to which drinking water might not be as quenching. Thickening of orbitofrontal cortex, a part of the brain that pleases and is activated after drinking water [15], when medically observed in older adults, results in lack of fulfillment and delight that follows water intake [15]. Premature decay of intention occurs when a certain activity takes longer than anticipated time to fulfil, or when an activity is thought of, but execution is hindered by some other distraction. Decay in intention is significantly higher in elderly people with cognitive defects [16]. The paper claims that the AR app proposed has advantages over existing water drinking reminder apps when it comes to helping cognitively impaired old adults to stay hydrated [4]. The app makes use of Vuforia marker- based technology and a game like activity to motivate users to meet/increase water intake. Furthermore, it also mentions carrying out a feasibility study of two versions of the app- basic and advanced - with elderly people (in assistance with their caregivers) to find out which of the two could be more suitable. It is, therefore, clear from the paper that the proposed AR game is beneficial for hydration amongst elderly people since it assists them to cope with their cognitive disabilities.

In the application-based paper, the authors use AR involving exercises designed to teach spatial concepts of rotation/revolution, solstice/equinox, and seasonal variation of light and temperature [5]. It utilizes ARToolkit to teach about Earth-Sun relationships to thirty undergraduate geography students. Users utilized a lightweight Cy-Visornf DH-440 head mounted display (HMD) with a Logitech QuickCam Pro 3000 video camera attached. The HMD and camera were connected to a laptop running Windows XP and ARToolkit version 2.52 software. The paper claims that students find it challenging to understand spatial concepts and phenomena that are complex, and the use of AR based application resulted in a significant improvement in student understandings along with reduction in misunderstandings. Often, teachers use 3D objects or props available in the classroom to explain complex concepts but both teacher and students struggle since the available objects often fail to mimic the actual concept. AR based applications usually come in handy at such scenario and eradicate the need for props. The research made use of pre- and post-assessment worksheets, and the analysis of the assessment resulted in some definitive statistics as follows:

- In general, conceptual, and factual understanding ofthe concepts improved in all cases.
- The most significant improvement was seen in those with lower pre-assessment scores.
- Most of the students resorted to pictorial descriptions to help illustrate their understanding on both pre and post assessment which further fortified the stance on use of pictures being more intuitive when it comes to understanding and explaining complex spatial concepts

The research also made some qualitative analyses and drew some definitive conclusions as stated below:

- Ability to interact with the interface and control over virtual objects helped students to understand more advanced concepts.
- In some cases, the students could no longer distinguish the difference between real and superimposed virtual objects. In no time, they felt like all virtual objects were assimilated in the real world.

The paper, thus, explores AR's potential to help student visualize complex spatial concepts, and puts forth a definitive conclusion that AR rightly assists students with their learning and understanding.

III. PROJECT ARCHITECTURE

A. *Approaches to Deployment*

The approach to deployment depends on the software development model and the wireframes of the applications. Waterfall development model was utilized to implement each stage of the application development and wireframes were utilized to easily identify the application requirements and deployment.

Waterfall development model: Waterfall model was used to develop all three applications discussed in this paper. The entire software development process is split into different phases, and each phase is carried out sequentially in this approach. For each of the applications the same fixed set of phases were defined as illustrated in Figure 7.

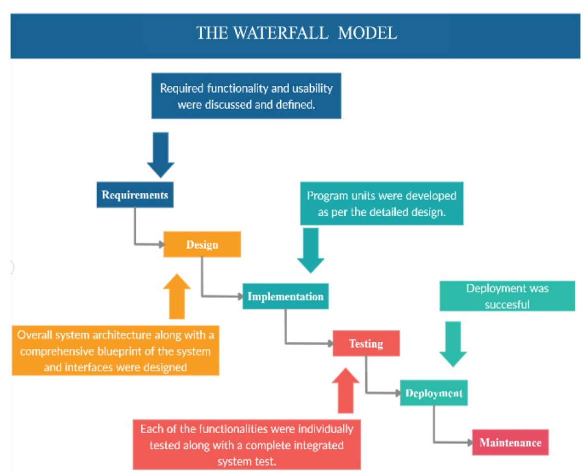


Figure 7. Waterfall Model.

Wireframes for Drinking Water AR Application: The welcome activity is a splash screen that shows the logo of the application and lets users know that the application is starting up. After the splash screen is successfully rendered, if it is the first time that the user is using the app then the app will ask the user for device camera and GPS permissions. When and if the user allows all required permissions, then the loading device location dialog is shown while device GPS is asynchronously being fetched by a background thread. After the location is fetched the app makes use of ARCoreLocation to fetch and position the water marker overlay on the device camera view. The main activity also has a view/file complaint button which can be used by general users to file complaint and admin users to view and resolve the complaints. The user authentication and data storage functionalities are achieved using Google's Firebase (a cloud database). The application wireframes are illustrated in Figure 8.

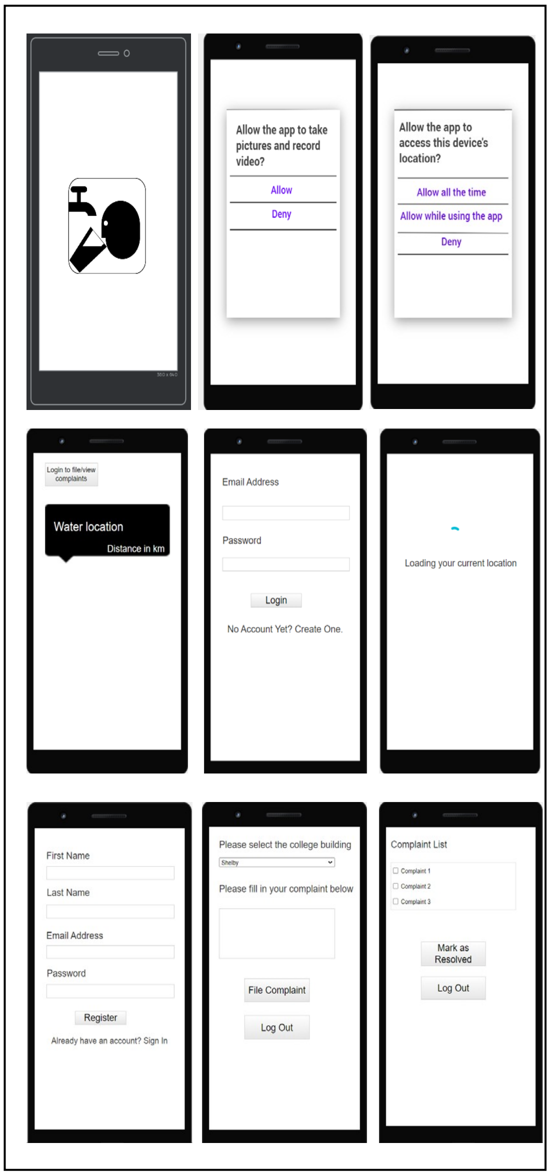


Figure 8. Drinking Water AR application wireframe.

Wireframes for Sun Path AR Application: The welcome activity is a splash screen that shows the logo of the application and lets the user know that the application is starting up. After the splash screen is successfully rendered, if it is the first time that the user is using the app then the app will ask the user for device camera permission. When and if the user allows camera access, the user is taken to the main activity of the application. All other functionalities of the application is found in the main screen of the app. The application wireframes are illustrated in Figure 9.

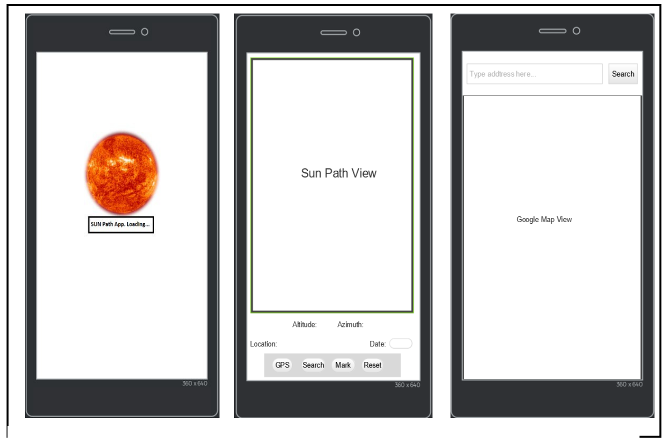


Figure 9. Sun Path AR application wireframe.

Wireframes for Solar System VR Application: The main screen is the world space view for the user. The view includes all eight planets and the Sun in the solar system. All planets are simulated to mimic their position, scale, and rotation relative to the Sun. Users can use the Oculus Quest controllers to teleport within the world view. To give users a more realistic feel, the controllers are disguised as human hands in the scene. Users can get closer to each planet and the Sun to have a better view of the objects and the text associated with the objects. The application wireframe is illustrated in Figure 10.

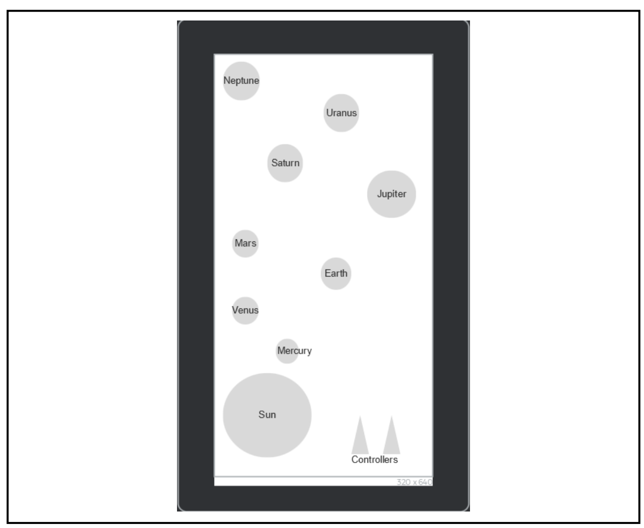


Figure 10. Solar System VR application wireframe.

B. Equipment Selection

Software & hardware requirements stay the same for AR applications and differ with the VR application.

AR applications software requirements:

- Minimum Android version: 7 (API level 24).
- Target Android version: 9 (API level 28)

AR applications hardware requirements:

- ARCore supported Android mobile devices.
- Target Android version: 9 (API 28)

VR application software requirements:

- Quest builds 20.0 release

VR application hardware requirements:

- Oculus Quest.

C. Drinking Water AR Application

The application began with the identification of the following functional requirements:

- Use Augmented Reality (AR) to show drinking water availability, consumption, and statistics on eco-friendly endeavor to reduce plastic cups in some of the buildings within Auburn University.
- Users of the application can see information overlay via their device camera when pointed to the respective buildings.
- Provision of two level of user authentication: General and Admin.
- General users can sign up and login to file complaints related to drinking water problems, if any.
- Admin users can sign up and login to view complaints and resolve them accordingly

The pictorial representation in Figure 11 is the flowchart that depicts the runtime flow of the water application. When the application is started it first checks to see whether the device is supported. If the device is supported, then the application seeks user permission to use device camera and GPS coordinates since both components are required for the application to run. Once the permissions are granted then the application initializes ARCore and ARCorelocation functionalities asynchronously. After the asynchronous methods return Future objects, the application renders the location markers.

The water AR application is developed in Android Studio using Java programming language, and libraries such as Google's ARCore, Google's FireBase and ARCoreLocation by APPoly. ARCore is Google's platform for building AR experiences. It assists a device to understand its real environment so that it can augment it. Two fundamental features of ArCore are as follows:

- Motion tracking: allows tracking position of the mobile device relative to the world.
- Understanding of the real world: Allows devices to understand vertical and horizontal surfaces and planes [17].

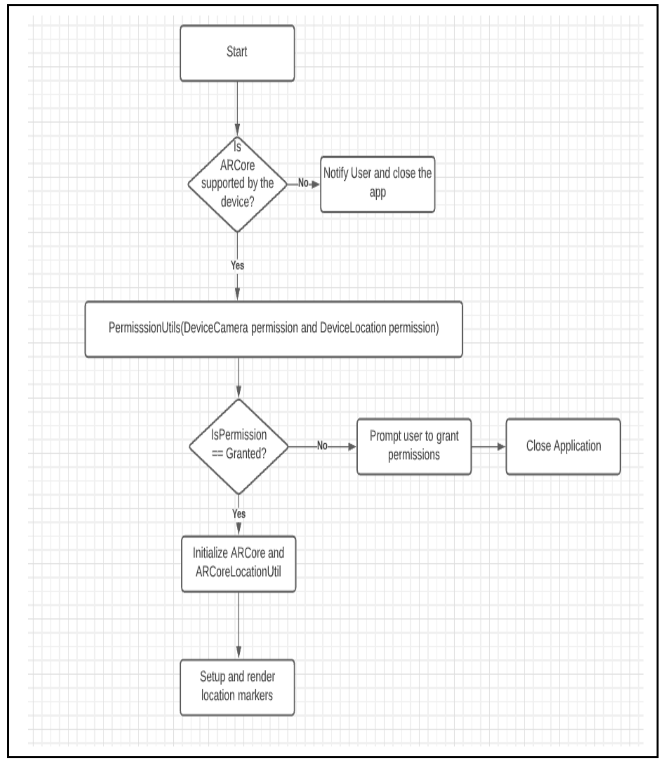


Figure 11. Flowchart of Drinking Water AR Application.

ARCore API which handles session lifecycle, access to device camera and pose is instantiated using ARCore session class. While this session is running ARCore holds exclusive access to device camera. Since this class consumes a significant amount of heap memory of the device, it is essential to call close method to release memory while not using the session. Failure to close may result in app crashing [18]. Similarly, ArSceneView is a SurfaceView which integrates with ARCore to render a scene [19].

Two of the methods from the ArSceneView class that have significant implementation in the application are getArFrame method which returns the most recent ARCore Frame, if available, and getSession method which returns the ARCore Session used by the view. Likewise, Frame class in ARCore captures the state and changes to the AR system by making a call to session object. It makes use of the getCamera method of the class to get the camera object [20]. Once the libraries are imported, to place a virtual object in a scene, anchor must be defined. Anchor class describes a fixed location and orientation in the real world [21].

Anchor in the application is obtained from the ArCoreLocation library by APPoly. APPoly is a software company based in the United Kingdom and contributes to the open-source community with various software packages. One such software package is ARCoreLocation. Since ARCore does not support use of real-world coordinates in its AR space [22], this application makes use of the ARCore Location library to realize the location-based functionality in the app.

The location library used to realize location- based AR is ARCore-Location: 1.2 [23]. ARCoreLocation allows the water app to position AR objects at real-world GPS coordinates. The real-world GPS coordinates (longitude and latitude) are provided to the application by making use of a JSON file.

The application data related to users and complaints is handled using Google's Firebase – a cloud service that is used to authenticate users and store data in Cloud Firestore. Cloud Firestore is a NoSQL, document-oriented database in which data is stored in documents and can be used to easily store, sync and query data for applications. These documents are organized into collections. There are no table or rows, unlike SQL databases. A series of key-value pairs is stored in each document. Subcollections and nested objects can be found in documents, all of which can include primitive fields such as strings or abstract objects such as lists. The relation between data, document and collection is summarized in Figure 12.

Figure 12. Collection in Cloud Firestore.

The application has two collections: complaint collection and user collection. Complaint collection holds the complaint documents that the users filed, and user collection holds the users registered in the system.

D. *Sun Path AR Application*

The application began with the identification of the following functional requirements:

- Use Augmented Reality (AR) to show Sun's path at a given date, time and location.
- Users of the application can see sun path related information overlay in their device camera while using the application.
- Users can search for any location coordinates provided by Google Maps.

The pictorial representation in Figure 10 is the flowchart that depicts the runtime flow of the sun path application. When the application is started, it seeks user permission to use device camera since the component is required for the application to run. Once the permission is granted then it initializes the default scene and overlays the sun path on top of the device camera view. The user then can select custom location, date and time and the application will update the scene accordingly. Figure 13 depicts the flowchart for the application.

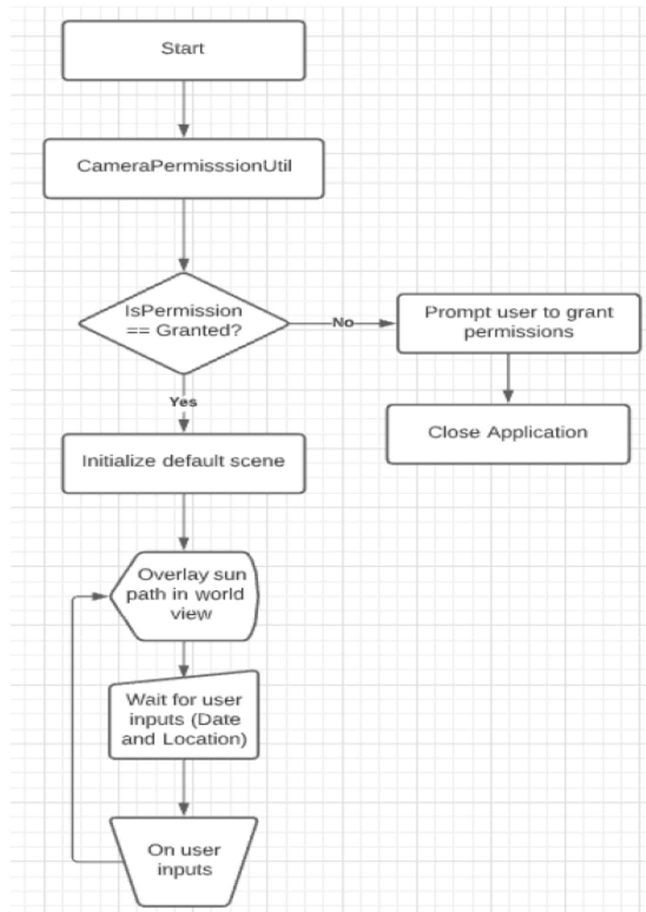


Figure 13. Flowchart of Sun Path AR Application.

The Sun path AR application is developed in visual studio using JavaScript programming language and React Native framework. React Native provides developers with a community of open-source modules that can be readily incorporated in app development. An overview of components is as illustrated in Figure 14.

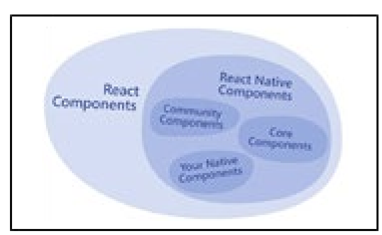


Figure 14. React Native overview [24].

This application is realized into three main custom components: 1. Location Component 2. Display Component, and 3. Main Component. Each of these components make use of core components and community components and interact with one another. Location Component: This is the component where location-based logic and code is written. This component makes use of following community components:

- @react-native-community/geolocation
- react-native-google-places-autocomplete
- react-native-maps

Display Component: This is the main user interface component where UI logic and code is written. This component mainly comprises of core components such as View, Text and ScrollView. The community components used are:

- @react-native-community/datetimepicker
- react-native-vector-icons/MaterialCommunityIcons

Main Component: Is the engine of the application. All custom components are called here along with the following community components:

- react-native-WebView
- react-native-camera

Amongst various sun position calculation algorithms (such as Spencer, Pitmann and Vant-Hull, Walraven, PSA, and Michalsky), PSA has superior accuracy and performance [25]. Figure 15 illustrates PSA algorithm’s performance in terms of accuracy in calculating zenith distance, azimuth, and sun vector deviation.

	Average	Standard deviation	Mean deviation	Range
Error in zenith distance	−0.001	0.114	0.091	[−0.398, 0.370]
Error in azimuth	0.002	0.190	0.138	[−1.568, 1.461]
Sun vector deviation	0.147	0.080	0.065	[0.000, 0.409]

Figure 15. PSA algorithm’s performance.

User provided GPS coordinates, date and time is fed into the PSA algorithm function. The function returns Sun spherical coordinates and that is used in a projection matrix to visualize the sun path and overlay it on top of the world view. The integration of the algorithm in the application is shown in Figure 16.

Three.js library is used by the app to draw 3D objects on the device's camera view. Three.js is a 3D library that draws 3D using WebGL. Essentially, the 3D library is used to build objects and then link them. It simplifies the production and manipulation of scenes, lighting, shadows, materials, textures, and 3D math, among other items. The renderer is the most critical aspect of a Three.js app. A Renderer receives a Scene and a Camera and renders the scene. A Scene object, which resides in the renderer, includes other objects such as mesh objects, light objects, and Object3D.

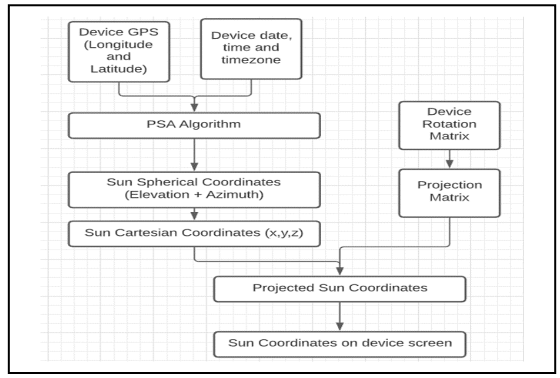


Figure 16. PSA algorithm integration in the application.

In three.js, a Mesh is the product of combining three things:

- A Geometry: It refers to the shape of the object.
- A Material: This refers to aspects like the geometry's color and texture.
- The object's location, direction, and scale in the scene in relation to its parent.

The application uses Amazon Simple Storage Service (Amazon S3) in Amazon Web Services (AWS) to load all of the front-end JavaScript files (such as Three.js), HTML and CSS. Amazon S3 utilizes REST API to efficiently store and retrieve data during runtime.

E. Solar System VR Application

The application began with the identification of the following functional requirements:

- Use Virtual Reality (VR) to create a 3D environment that simulates the solar system.
- Simulation will include planetary revolution and rotation.
- Users can teleport within the solar system to have a closer look at the Sun, planets and their moons.

Figure 17 shows the flowchart of the runtime flow of the Solar System VR application, developed in Unity using C# programming language. When the application is started, it first checks to see if the headset in which the application is being run is compatible. The application currently only supports Oculus Quest, and so trying to run it on other headset will cause the application to crash. After the initial validation is successful, the app will then initialize the camera component and the world space/scene of the application. Immediately after, the application will render all GameObjects of the scene and start the planetary rotation script (which is used to simulate planet revolution around the Sun). While in the world space of the application, the user can use controllers to teleport to different areas in the solar system and have a closer look at each of the planets. As illustrated in Figure 18, a scene in Unity can have objects that are called GameObjects. GameObjects serve as containers for components. Depending on the type of object desired, various combinations of components can be added to a GameObject. Developers can either use in-built components or create a custom component using Unity Scripting API [26].

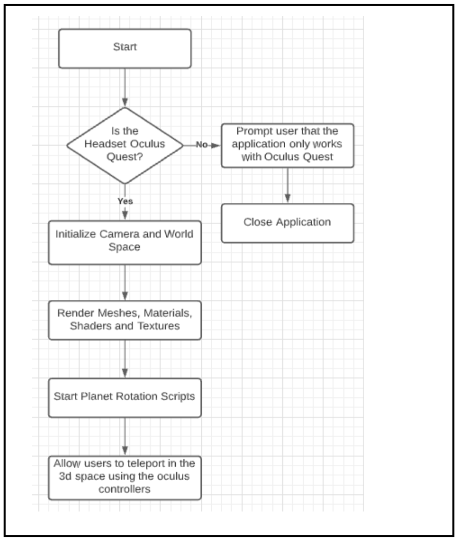


Figure 17. Flowchart of Solar System VR Application.

Transform component, which defines position and orientation of the object it is attached to, is the only indispensable component in a GameObject. All other components either default or custom can be attached or detached from a GameObject.

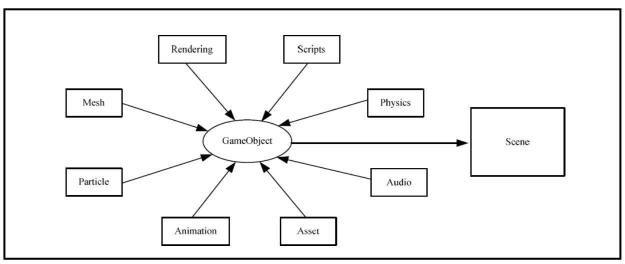


Figure 18. Unity3D GameObject Component Model.

IV. USABILITY STUDY

A. Virtual Educational K12 Camp

Research in the Formation of Engineers (RFE) computing virtual camp was conducted for K-12 students, in which, students from grade 9 to 11 participated. These students were instructed on important topics of AR/VR and were also asked to use the applications. The following observations were gathered from students' responses:

- All the students were unsure if they had used AR/VR applications before, as shown in Figure 18.
- Many students indicated that the use of AR/VR functionalities helped them in better understanding of subject topics.
- Students equivocally agreed that the apps were easier to use and that they were able to effortlessly determine drinking water sources and sun location.

Pre-survey and post-survey detail the discrepancies in subjects' comprehension before and after using the AR/VR applications developed for this research. Students developed a greater understanding of the technology by using the applications. According to the post-survey findings, as depicted in Figure 19, 50 percent of the students strongly agreed, and the other 50 percent agreed that using such technologies was interesting. The Drinking Water AR app made it easy to find drinking water places, according to 50 percent who agreed somewhat, 25 percent who agreed, and the remaining 25 percent who strongly agreed. Similarly, 50 percent strongly agreed, 25 percent agreed, and 25 percent slightly agreed that determining the sun's location using the Sun Path AR app was simpler.

B. Pre-Survey

According to the pre-survey findings, as shown in Figure 19, 100 percent of the students were unaware of the AR/VR technology prior to being introduced in this study.

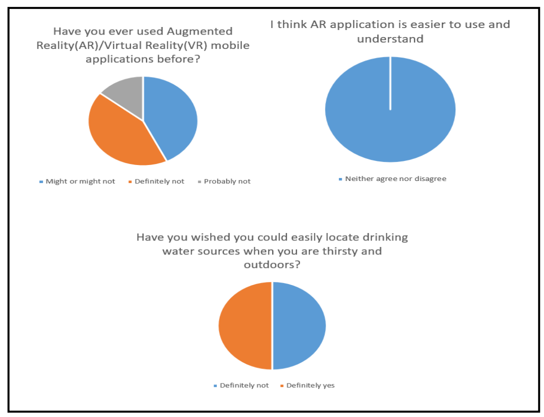


Figure 19. Pre-Survey results.

C. Post Survey

According to the Post Survey findings, as shown in Figure 20, 100 percent of the students agreed that using AR/VR technologies is interesting.

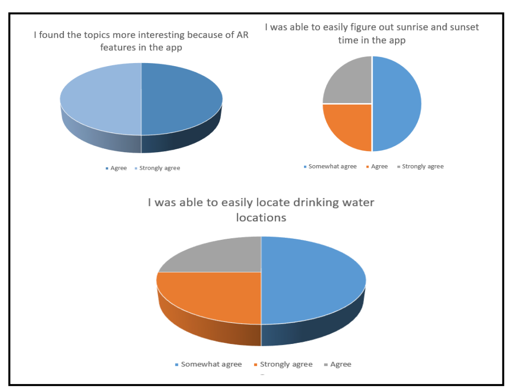


Figure 20. Post-Survey results.

V. RESULTS

A. *Drinking Water AR Application*

The application makes use of location-based AR to overlay virtual objects when the device is pointed towards the line of sight of POIs. Currently, Haley building, and Shelby Engineering buildings are the POIs for the application. Figure 21 shows the overlay when the app is brought to the line of sight of one of the coordinates. By pointing their phones towards the POIs, users can quickly identify drinking water sources and related information. In addition, the app also promotes civic engagement. Users can register/sign into the program and then file complaints about water supplies.

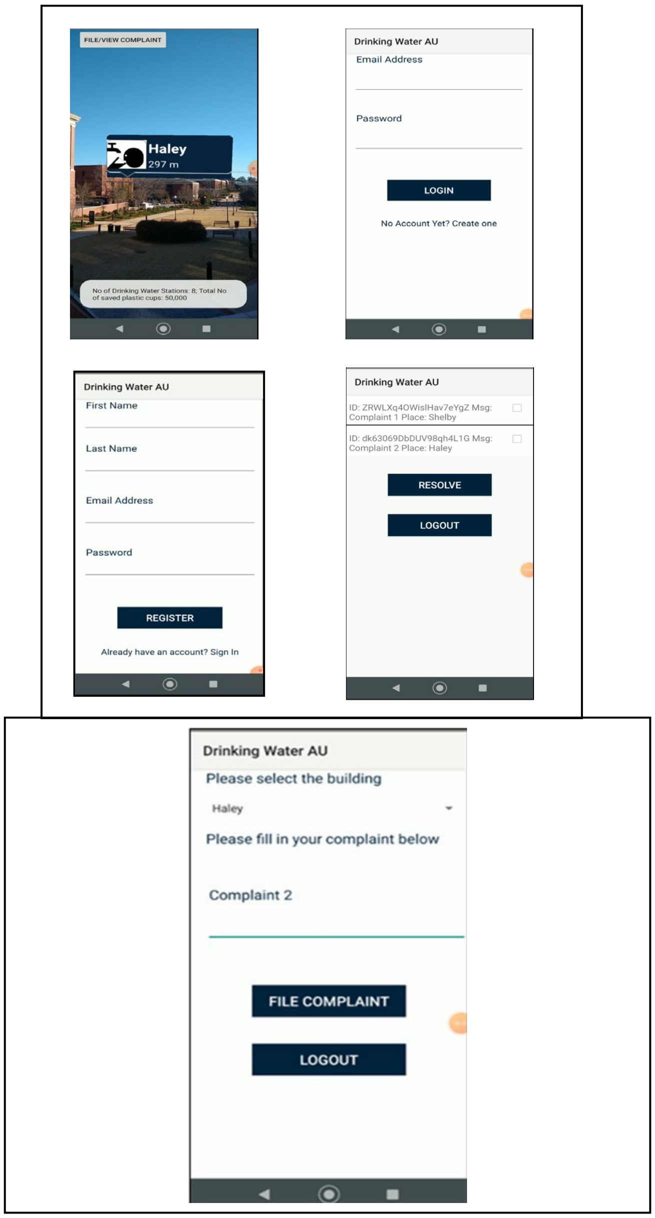


Figure 21. Drinking Water AR Application prototype.

Also, the application has provisions to add admin users. Admin users can look at the complaints and mark them resolved when accomplished. Currently, people on campus can use Google Maps to locate buildings but they do not have access to building related information. In the future, the application could be expanded to provide users with not only water information about the buildings, but also information about the buildings' internal mappings.

B. *Sun Path AR Application*

The application makes use of marker-less AR to overlay virtual objects in device's camera view. The main scene of the app displays Sun's path at a given date, time, and location, as illustrated in Figure 22. Sun is the major source of energy to our planet, and examining its path is essential for better harvesting its energy. Sun path diagrams provide a wealth of information on how the sun can affect a site and structure over the year. The solar azimuth and altitude for a given position can be determined using the diagram. A conventional way of examining its path is by manually plotting points/lines in the diagram to get solar azimuth and altitude. Accurate and timely analysis of Sun's path plays a significant role in multitude of sectors. This app eliminates the need to manually measure the position of the sun at aspecific date, time, and place.

Users can easily access sun related information such as sun position, sunrise time and sunset time. Users can use the search functionality in the app to visualize Sun's path in any coordinates searchable in Google Maps API, as demonstrated in Figure 22. Practical uses, such as estimating solar power and solar water capacity, as well as agricultural applications, are possible with this app.

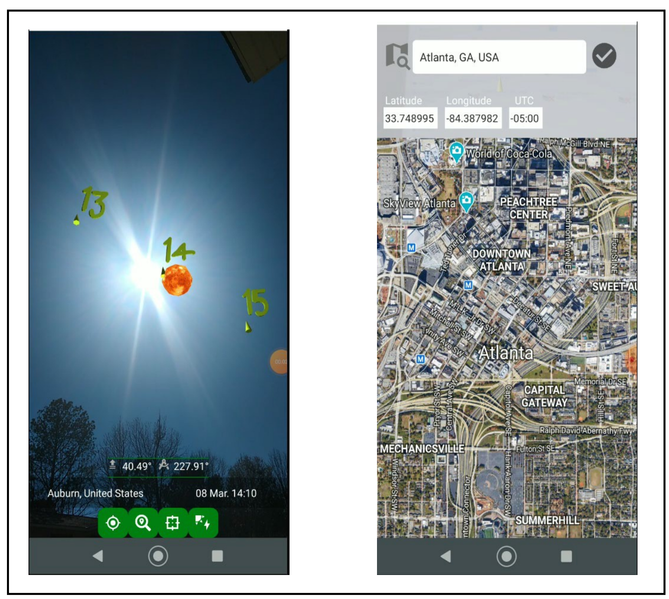


Figure 22. Sun Path AR Application prototype.

C. Solar System VR Application

The application makes use of VR to simulate our solar system. In the main scene, users can see the movements of the planets, as wells as their form, scale, and location in relation to the Sun. Users can use controllers as their hands to teleport within the app and have a better visual of the Sun and planets, as illustrated in Figure 23. This can be useful to K-12 education as it provides an immersive, interactive way to visualize and comprehend the solar system. This way of teaching using VR could be extended to other subjects.

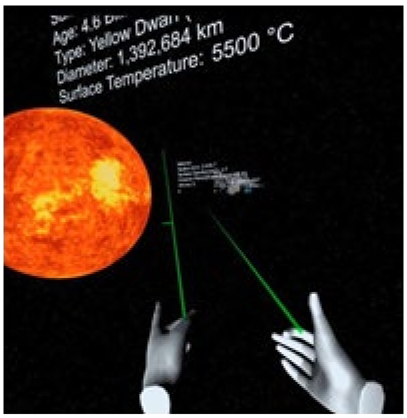


Figure 23. Solar System VR Application prototype.

VI. CONCLUSIONS AND FUTURE WORK

The Drinking Water AR app served as a prototype to resolve the issue of access to safe drinking water while also encouraging public participation by enabling users to file water-related complaints. By assisting users in visualizing sun path at a given time, date, and place, the Sun Path AR application served as a prototype to help users learn about sun path and its role in making solar energy affordable. It provides solar azimuth and altitude information to the user, eliminating the need to manually calculate the values usinga sun path diagram. The Solar System VR app acted as a model for enhancing virtual reality by creating an immersive and interactive solar system application. The app aided the user in visualizing a concept which is not readily apparent.

The effectiveness of apps was also evaluated among K-12 students using a Likert scale-based pre- and post-survey metric. The study included twelve students from various schools across the United States. Based on the user reviews, it is fair to say that the applications were effective in terms of interaction, functionality, usability, and user experience.

However, the implementations and evaluations had some limitations that could be addressed in the future. A virtual camp, conducted online due to Covid-19, was not quite effective to quantitatively evaluate the effectiveness of the work. In the future, the authors propose evaluating the application in an in-person camp with greater number of participants. Besides, following changes to the applications is proposed:

- Currently, the water application only supports two of the buildings inside Auburn University, and so scaling it to add more buildings is proposed.
- The mock data used for drinking water application could be replaced with actual data from the university.
- Both the sun path application and the water application are developed for android phones only. So, equivalent versions of the applications compatible to iPhone could be developed.
- Similarly, solar system VR application is only runnable in Oculus Quest Headset and could be built to support a greater number of headsets.

ACKNOWLEDGMENT

This work is made possible due to an NSF Grant # 1826181.

REFERENCES

[1] B. Bhattarai and D. Marghitu, "Use of Augmented Reality (AR) and Virtual Reality (VR) to Address Four of the “National Academy of Engineering Grand Challenges for Engineering in the 21st Century,” in Proc. ICWMC, 2021, pp. 44-53..

[2] “Grand Challenges - 14 Grand Challenges for Engineering.” http://www.engineeringchallenges.org/challenges.aspx (retrieved: Jan. 24, 2021).

[3] L. Pombo and M. M. Marques, “Improving students’ learning with a mobile augmented reality approach – the EduPARK game,” ITSE, vol. 16, no. 4, pp. 392–406, Nov. 2019, doi: 10.1108/ITSE- 06-2019-0032.

[4] S. Lehman, J. Graves, C. Mcaleer, T. Giovannetti, and C. C. Tan, “A Mobile Augmented Reality Game to Encourage Hydration in the Elderly,” in Human Interface and the Management of Information. Information in Applications and Services, vol. 10905, pp. 98–107, 2018.

[5] B. E. Shelton and N. R. Hedley, “Using augmented reality for teaching Earth-Sun relationships to undergraduate geography students,” in The First IEEE International Workshop Augmented Reality Toolkit, Sep. 2002, p. 8 pp.-, doi: 10.1109/ART.2002.1106948.

[6] H.-G. Lee, S. Chung, and W.-H. Lee, “Presence in virtual golf simulators: The effects of presence on perceived enjoyment, perceived value, and behavioral intention,” New Media & Society, vol. 15, no. 6, Art. no. 6, Sep. 2013, doi: 10.1177/1461444812464033.

[7] C. Covarrubias, “Immersive Technology: What is it and How can we Use it Today?,” Advrtas, Mar. 10, 2021. https://advrtas.com/immersive-technology/ (accessed Mar. 10, 2021).

[8] “6 Exciting AR Apps for Student Learning,” Edutopia. https://www.edutopia.org/blog/ar-apps-for-student-learning-monica-burns (retrieved: Mar. 10, 2021).

[9] Y. El Filali and K. Salah-ddine, “Augmented Reality Types And Popular Use Cases,” vol. 8, pp. 91–97, Apr. 2019.

[10] A. Ayoubi, “IKEA Launches Augmented Reality Application,” Architect, Sep. 21, 2017. https://www.architectmagazine.com/technology/ikea-launches-augmented-reality-application_o (retrieved: Mar. 10, 2021).

[11] Locatify, “Location Based Augmented Reality Apps (AR & RTLS),” Locatify. https://locatify.com/blog/location-based-augmented-reality-apps-2017-rtls-ar/ (retrieved: Mar. 10, 2021).

[12] “Oculus Headsets: The Original Quest is Back | Oculus.” https://www.oculus.com/quest/ (retrieved: Mar. 10, 2021).

[13] D. Heaney, “How virtual reality positional tracking works,” VentureBeat, May 05, 2019. https://uploadvr.com/how-vr-tracking-works/ (retrieved: Mar. 10, 2021).

[14] J. C. Mentes, “A typology of oral hydration problems exhibited by frail nursing home residents,” J Gerontol Nurs, vol. 32, no. 1, pp. 13–19; quiz 20–21, Jan. 2006, doi: 10.3928/0098- 9134-20060101-09.

[15] I. E. T. de Araujo, M. L. Kringelbach, E. T. Rolls, and F. McGlone, “Human cortical responses to water in the mouth, and the effects of thirst,” J Neurophysiol, vol. 90, no. 3, pp. 1865–1876, Sep. 2003, doi: 10.1152/jn.00297.2003.

[16] J. Pacheco, J. O. Goh, M. A. Kraut, L. Ferrucci, and S. M. Resnick, “Greater cortical thinning in normal older adults predicts later cognitive impairment,” Neurobiol Aging, vol. 36, no. 2, pp. 903–908, Feb. 2015, doi: 10.1016/j.neurobiolaging.2014.08.031.

[17] “ARCore,” Google AR & VR. https://arvr.google.com/arcore/ (retrieved: Mar. 10, 2021).

[18] “Session | ARCore,” Google Developers. https://developers.google.com/ar/reference/java/com/google/ar/c or e/Session (retrieved: Mar. 10, 2021).

[19] “ArSceneView | Sceneform (1.15.0),” Google Developers. https://developers.google.com/sceneform/reference/com/google/a r/ sceneform/ArSceneView (retrieved: Mar. 10, 2021).

[20] “Frame | ARCore,” Google Developers. https://developers.google.com/ar/reference/java/com/google/ar/co r e/Frame (retrieved: Mar. 10, 2021).

[21] “Anchor | ARCore,” Google Developers. https://developers.google.com/ar/reference/java/com/google/ar/co re/Anchor (retrieved: Mar. 10, 2021).

[22] Appoly, “ARCore Location - Android Studio - Appoly News & Opinions,” Appoly, Mar. 28, 2018. https://www.appoly.co.uk/2018/03/28/arcore-location/ (retrieved: Mar. 10, 2021).

[23] Appoly, appoly/ARCore-Location. Appoly, 2021.

[24] “Core Components and Native Components · React Native.” https://reactnative.dev/docs/intro-react-native-components (retrieved: Mar. 10, 2021).

[25] M. Blanco-Muriel, D. C. Alarcón-Padilla, T. López-Moratalla, and M. Lara-Coira, “Computing the solar vector,” Solar Energy, vol. 70, no. 5, pp. 431–441, Jan. 2001, doi: 10.1016/S0038-092X(00)00156-0.

[26] “Unity – Manual: GameObjects.” https://docs.uity3d.com/Manual/GameObjects.html (retrieved: Mar. 10, 2021).

Co-development of a Virtual Reality Tool Designed by and for Students for Training in Electrical Hazards: the VirtuElec Project

Jean-Pierre Toumazet, Jean-Yves Laronde
Institut Universitaire de Technologie Clermont Auvergne – dept. GEII (electrical engineering) de Montluçon
Université Clermont Auvergne
Clermont-Ferrand, France
Email : j-pierre.toumazet@uca.fr

Jean-François Vérité
2J Process SAS Company
Virtual reality development department
Aix-en-Provence, France

Abstract - **The teaching of electrical hazards for future professionals is an important issue. This problem is complex, because to train students in risk, it is necessary to confront them with dangerous situations, but without making them take risks. The VirtuElec project was born in this context: co-producing, with a company specialized in virtual reality, an environment simulating electrical hazards and allowing to train according to different scenarios at different levels of competence. The originality of the project is to involve the students themselves in the construction of this environment. By integrating a project team, they worked in a design office to co-develop this tool and enrich it with video and virtual supports: a training support carried out by students and for students. Students who participated in this project gained knowledge in the areas of electrical hazards, virtual reality, and teamwork, but they felt they gained the most proficiency in the last two skills. Finally, this tool was implemented within the framework of a semi-autonomous training module combining real and virtual experimentation, in order to offer students the most complete experience possible of the professional situations they will face in their future jobs.**

Keywords - Co-design; Electrical hazards Simulation; Immersive Virtual Reality; Interactive Devices; Multi-disciplinary; Problem-Based Learning; Computer Supported Collaborative Learning; Professional gestures

.

I. INTRODUCTION

It is essential for students in the field of competence of electrical engineering to have a good knowledge of electrical hazards, in order to be able to identify them and prevent accidents, which are usually very serious. This is indispensable first in the context of their training and for the actions they will have to carry out in the professional world. This more than true since, with the development of apprenticeship training and more generally the professionalization of training, learners are increasingly confronted in their training course with situations of potential danger in an industrial environment.

At present, training exists for the students concerned, but it takes place under conditions quite different from those they will face in their professional activity. Even if some interventions are indeed carried out in conditions close to the industrial context, on a real electrical cabinet, they require the presence of a trainer for a student, due to the presence of very real risks. Moreover, this situation is stressful for the student faced with the risk (even if all precautions are taken) and for the trainer, who assumes a heavy responsibility.

The objective of the project, co-developed with a company specialized in virtual reality, was the design of a virtual environment presenting different scenarios of electrical risks. Depending on the level of expertise of the learner, a mission is given to him or her and he or she must take the right decisions in terms of choice of intervention and protection equipment, of behavior in the face of a risk, and of control of professional gestures.

The objective is to have a unique, innovative, modern and efficient tool developed specifically for our training needs. This allows learners to be able to train at their own pace, on a realistic and secure tool, but also to be able to address all aspects of electrical risk, in autonomy and to benefit from an individualized experience, with an individualized teacher feedback, focusing on the key elements related to each situation. The originality of the project is that it is a real partnership with a company, really involving students who have worked in a design office context, co-developing the tool in terms of ergonomics and functionality. Because of the health context related to the Covid19 pandemic, this collaborative work was necessarily more difficult, and it required the use of collaborative work tools implementing an Agile method in order to make the exchanges more efficient: an educational tool developed by students for students [1]. Obviously, the purely virtual experience can appear to be a bit reductive compared to a real situation. This is why the created tool has been implemented in the framework of a new teaching module based on experiential learning and aiming at developing students' autonomy. The virtual experimentation was thus completed by an experimentation on real systems, but involving lower electricity power for safety reasons.

After a presentation of the context of the use of virtual reality for the prevention of occupational risks and the modalities of development of the project in Section 2, Section 3 describes the methodology followed by the project team, the functionalities of the developed tool, and the methodology and tools used. Finally, Section 4 presents first results of the evaluation of the perception of the project by the students who participated in its conception, as well as the choice of implementation of the virtual tool in relation to real-life experiences, in order to provide the students with the most complete experience possible.

II. Context

The use of virtual reality for education has been the subject of many articles in recent years, addressing both the interest of this new tool, but also its limitations [2]-[5]. If we refer more specifically to the field of occupational safety and health, realistic virtual environments have many advantages [6][7]. The first and most obvious is the possibility of exploring an environment of potential danger, without taking risks, but also without putting colleagues at risk or causing damage to equipment. The learner also has the opportunity to perform actions and make mistakes without risk to him or her, and without feeling the pressure of other colleagues or the assessor, as would be the case in a real-life scenario [8]. He or she also has the possibility of being wrong, without consequence, and as many times as necessary. All these conditions lead the learner to be a real actor of his or her formation: the solution cannot come from elsewhere; the learner is obliged to act, to interact with what surrounds him or her, to progress in the mission.

The specific case of electrical hazards, which currently remains one of the main causes of fatal accidents in industrial environments, has been the subject of numerous studies and developments in the field of virtual reality. In particular, we can mention the application of virtual reality for the training of electricians working on substations [9][10], in the field of construction [11] or power distribution networks [12], and more generally in all fields where electrical risks are present [13]. These generally concern tools for experienced professionals, the aim of which is to enable them to train on devices on which it is usually difficult, if not impossible, to train, either because of their difficult access or because it is impossible to manoeuvre on these elements without causing a customer blackout. Other types of educational tools exist in the context of electrical risk training, the most common being based on videos illustrating risk situations and including quizzes that allow students to position themselves in terms of what to do when an electrical risk appears. Even if these tools illustrate realistic situations, the students are always in the position of an outside observer, and are never really confronted with the potential danger. These learning conditions are therefore far from reality.

The VirtuElec project's approach is complementary to these developments: it aims to enable the training of students with no professional experience related to interventions on electrical installations. In this context, the virtual environment is of course intended to reproduce as closely as possible the real environment, but not in order to remind the learners of their daily life, but on the contrary to make them discover what their real future environment will be. In these circumstances, it is important to simplify the environment by not including too many non-essential elements, so that the learner can quickly focus on his or her mission.

In addition, the target audience is wide, and skill levels in electrical hazard situations are very different. It was therefore necessary to provide a virtual environment compatible with several intervention scenarios.

This project also fits into the restructuring process of university and technological formation in France, now based on a competency-based approach. Within this framework, teaching methods are expected to evolve significantly, and the use of video, virtual reality and augmented reality media to help students gain autonomy is logically expected to grow and generalize.

III. Materials and Methods

The first step in this work to create a virtual environment from realistic situations and equipment present at the university was to constitute a 'project team'. The work carried out is highly multidisciplinary and has been the subject of a close partnership between (1) the "Electrical engeneering team" including two teachers in the field of electrical hazards, two technicians specialized in this field, and 9 undergraduate students; (2) the "partner company", a specialist in virtual reality and recipient of several awards in this field, (3) the "audio-visual team", composed of two technicians, for the production of audio and video media and (4) the "pedagogical engineering team", made up of two pedagogical engineers whose mission is to ensure the accompaniment of all the actors in this innovative approach (Figure 1).

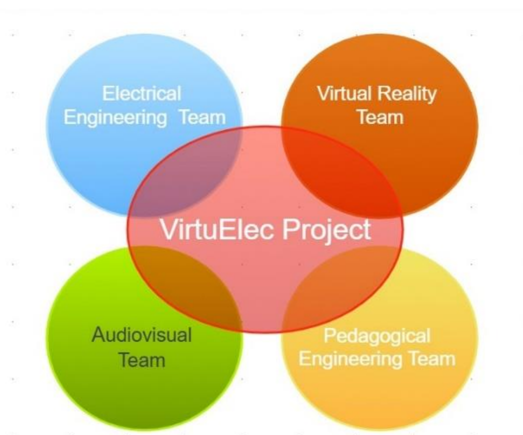


Figure 1. Synoptic of the VirtuElec project

The highly multidisciplinary nature of the project required the implementation of collaborative tools. This is all the more true that the specific context linked to the Covid19 pandemic has strongly complicated this process of joint work. In addition to the difficulties linked to the highly multidisciplinary aspect of the project, there were also difficulties in exchanging ideas during meetings, as well as the complexity of carrying out tests and trials, demonstrations of equipment and simulations of the modelled equipment as well as the envisaged scenarios. In order to be able to work remotely during the containment phases, different tools were therefore set up to share information, documents and experience.

The first one consisted in the implementation of the Microsoft Teams tool, at the initiative of the project manager and the teaching team. Microsoft Teams is the tool chosen by our university to ensure pedagogical continuity during the containment phases [14]. Within the framework of this project, it allowed a collaboration between the various teams involved and located on different geographical sites in France. The functionalities available to them were exploited at several levels: first of all, a collaborative space was created, in the form of a "VirtuElec" team, in order to allow information exchanges. The framework documents of the project (specifications, planning, framing document) were thus shared. This space was also intended to be a place of exchange, and it was the support of our bimonthly progress meetings, involving all the participants. Beyond these imposed deadlines, the exchange space was also used freely by the students, to exchange on the progress of the project, without intervention of the technical and pedagogical team. A document sharing system was also set up in order to allow the co-construction of the different stages of the project, with a progress follow-up managed in the form of different versions of documents, thus allowing the students to go back if necessary. Finally, during this very delicate period, the schedules of all parties were regularly disrupted, and the large number of people involved made it difficult for everyone to participate in the scheduled meetings. It was therefore decided to record videos of the most important meetings, with the prior agreement of all participants, in order to allow everyone to follow the project. It should therefore be noted that, while the Covid19 crisis obviously complicated the realization of this work, it also accelerated the deployment of tools that were not previously in use and that proved to be of real added value for the monitoring of the project.

Another modality has been put in place in order to gain efficiency in the monitoring of the project: the Agile method [15][16]. This can be defined as a project management method that consists of breaking down a project into a succession of small, quickly attainable objectives. Each step corresponded to a very short period of time, from one to two weeks, and was complementary to the overall project planning, which spanned three years and included design, deployment and feedback. In terms of concept, the Agile method emphasizes people and their interactions over processes and tools, close collaboration between different stakeholders, and adaptation to change rather than following a plan. This approach has many advantages. It allowed us to adapt to unforeseen circumstances that arose, such as the underestimation of the time needed to model the virtual environment or the complexity of describing exhaustively the different tasks to be performed in each scenario. But the most important was the notion of feedback, which is at the heart of the Agile methodology. Thus, at each step, a quick assessment was made, allowing to orientate the next steps of the project.

From a practical point of view, the virtual environment was developed to be implemented on Oculus Quest 2 virtual reality headsets, which are autonomous headsets allowing greater freedom of movement and better portability of the device [17]. The possibility of connecting the helmet to a large screen for demonstrations was also provided. Finally, the tool made also allows recording the journey of each learner in the virtual environment in video format, to be able to debrief a posteriori.

The actions carried out by the students in relation to the project teams can be divided into 3 main steps:
The first consisted of the design of the 3D environment, including the virtual electrical cabinet on which the learners will have to intervene, but also the operative part, consisting of a robotic arm, as well as a intervention preparation room where the necessary equipment is available. For more realism, a real device present on the training center inspired this virtual set. In this context, the students had to imagine the virtual electrical installation, draw up the electrical diagram, define and then model all the components, and finally to design the implementation in the complete virtual environment (Figure 2).

The second step was to design the intervention scenarios for electrician apprentices, simulating 3 levels of competence in a professional environment. The simplest is the beginner level, (including two options: step-by-step guidance or free learning) during which the operator must replace a defective element that was previously indicated to him. The most complex is the autonomous level (autonomy in actions, alerts are displayed in case of error and aids are available) during which the operator must look for the cause of a failure and perform troubleshooting. Finally, the expert level (anticipation of risks before the intervention) during which the operator must manage the safety of a teamworking on an installation. For each of these levels, the learner is expected to be able to choose the protective equipment and tools required for their mission, follow the correct procedure, and perform the correct technical actions.

In order to have a common language between the “electrical engineering team” and the “virtual reality team”, the scenarios description was developed in Grafcet (graphic programming, easily interpretable by all) [18] (Figure 3).

Figure 2. Actual electrical cabinet (a) and associated VR modelling (b)

For the virtual experience to be realistic, it was essential to give the learner the opportunity to make mistakes up to virtual electrification, simulated by a vibration. On the initiative of the students in charge of the project, these possible errors were translated into different colours in the graphic. The green colour logically corresponds to the case where the operations are performed correctly. The red one corresponds to errors that require to stop the current action and to start it again (the troubleshooter does not intervene on the right element for example). Finally, the orange one corresponds to errors that do not have an immediate consequence, but will cause an accident in the long term (the operator has incorrectly assessed the risk, or poorly chooses its protective equipment). In case of false manipulation, a vibration of the joysticks and a flash light simulates the electric shock (Figure 4).

Figure 3. Programming of intervention scenarios

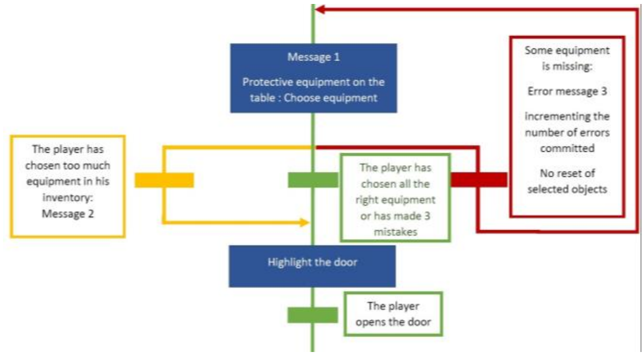


Figure 4. Example of programming scenarios in grafcet language

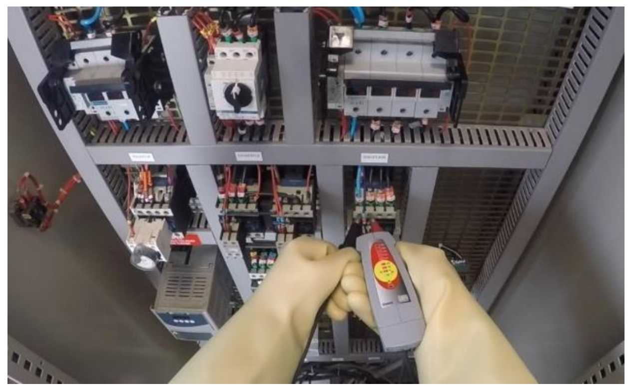


Figure 5. Image extracted from a video explaining the voltage measurement process

Finally, to illustrate the complex technical gestures, difficult to represent in the virtual environment, because demanding in terms of calculation capabilities, videos are accessible in the form of help in the virtual environment. These videos were shot with immersive cameras to illustrate as accurately as possible the technical actions to be performed, as shown in Figure 5 in a troubleshooting procedure

The final step, which is still under optimisation process, is the encoding of the scenarios algorithm by the partner company and their testing in the virtual environment for validation.

During the entire development phase of the project, a double follow-up was carried out: first, a technical follow-up, with regular meetings (each week) including the different teams involved in the progress made, and a pedagogical follow-up in the form of questionnaires offered to students in order to collect their feelings on different aspects of the project. The areas questioned were the evaluation of the accompaniment in the project, the links between the project and the teaching, the material and technical environment and the contributions for them of the project, in particular its multidisciplinary aspect.

IV. Preliminary Results

The pedagogical interest of this project lies at two different levels; the creation of a custom VR tool for the training of students on the one hand, and the pedagogical approach of co-design of this tool, including future users in the design of the training tool on the other hand. Now, the tool is still under development, and it is therefore too early to assess its impact on the electrical risk training of students. This analysis will of course be carried out as soon as the tool is in place, scheduled for december 2021. Following this initial implementation and feedback from the student evaluations, we plan to continue to evolve the scenarios and to a lesser degree the virtual environment, in order to make it even more efficient. This evolution will be based on the results of surveys submitted to students in order to analyze their appropriation of the virtual reality tool. These tests will be carried out at two levels: firstly with beginners, for whom the virtual environment will be the only experience of electrical risks, but also with final year students, who will have had the opportunity to work on real electrical installations. The idea behind this double evaluation is to assess the tool's ability to address both the problem of discovering electrical risks for non-electricians and the realism of simulated situations for more experienced users.

The analysis that we have already been able to do relates to the approach of co-designing the teaching support by involving the students themselves. The results of the evaluations, relating to the items described in the description of the method, made it possible to highlight several interesting elements.

First, the students expressed an overall satisfaction level of 9.5/10 for this project. They particularly appreciated the autonomy they were given (100% satisfaction) as well as the support and follow-up (89% very satisfied, 11% satisfied).

In terms of support, the students did not have any difficulty managing the high number of interlocutors around them. On the contrary, they appreciated the cohesion and the diversity of this team and the associated skills were really experienced as an asset. Collaborative work with specialists in other technical fields was evaluated as an asset by 78% of students, both an asset and a constraint by 11% and neither an asset nor a constraint by 11% of students

In terms of the links between the project and teaching, the students' perception is clearly that they have developed competence in fields not directly related to their core training, electricity: 56% strongly agree and 44% agree with this perception. They are also aware that they have made progress on aspects related to electrical safety, but less noticeably: 0% strongly agree, 78% somewhat agree, and 22% somewhat disagree with this perception. It is clear that in this respect, students are underestimating their rise in competence in their field of specialization. To be able to write the intervention scenarios in the virtual environment, students had to reach a high level of mastery of the intervention rules in a context of electrical risk, but this essential aspect appeared secondary to the students compared to the new skills in virtual reality and video production. The reason may be that these two areas are completely absent from their initial competency panel, so the discovery was total. However, from the teacher point of view, it is in the field of electrical risks that the students in charge of the project have made the most technical progress.
In terms of the contribution of the project, the multidisciplinary and collaborative dimensions of the project were the most appreciated by the students, more than the technical contribution on the heart of the subject, that is to say the management of electrical risks. The collaborative work demonstrated the need to communicate with people with other skills, and it was these values of openness and communication that were considered the most beneficial elements of the project. Thus, to the question of the main skill developed in the framework of this project, 56% of the students answered the ability to communicate with interlocutors from various fields of competence, 33% the ability to formalize expectations and only 11% the ability to master the technical elements.

Even if this tool is still in the optimization phase, the first version has already been tested as part of the implementation of a new pedagogical approach based on competency and the increase in student autonomy. This approach was initiated this year for students entering the university. The aim is to confront students with problems similar to those they are likely to encounter in a professional environment, and to guide them in their progress while leaving them free to develop their own experimental methodology, to learn from their mistakes, and thus to increase their competence.

This approach is implemented in particular in the field of electrical energy production, and it is in this context that the VirtuElec tool makes sense. An experimental learning situation has been set up, within the framework of the understanding of the functioning of the photovoltaic energy production and the implementation of the complete chain of production and transformation of the electric energy. The idea here is to couple classical experimentation with virtual reality: the experimentation is carried out on low-power structures, which do not present any electrical risks and allow autonomous testing for the students. If it allows to understand the operating principles of the various devices, it remains however rather far from the conditions of implementation of the real systems. The virtual reality tool offers here this possibility of change of scale, by authorizing the intervention on realistic systems simulating real installations. More than the virtual tool itself, it is therefore the combination of reality and virtuality that allows us to address a wide range of situations and to increase the student's competence (Figure 6).

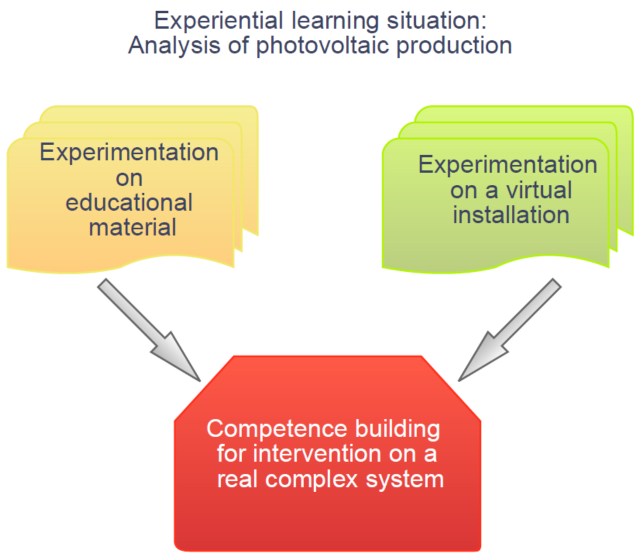


Figure 6. Description of the experiential learning approach mixing virtuality and reality

The first feedback from the students on this approach clearly shows the interest of this coupling of real experimentation and virtual experimentation. The virtual representation only makes sense when it is associated with real problems, and identified experimentally. On the other hand, experimentation on pedagogical models only makes sense when virtual reality allows us to evaluate how to transpose these results to a more realistic scale.

This phase of deployment of the VirtuElec tool has thus opened up new perspectives, by federating a larger teaching team around these aspects mixing real and virtual experimentation. It will also be possible to enrich it with another approach, which is still in its infancy and which consists of creating digital twins of two experimental devices present at the university. The students will thus have access to a complete panel of solutions coupling virtuality and reality.

V. CONCLUSION

The developed system allows the learner to be truly immersed in the virtual context. It allows students to project into their future professional world, and to progress at their own pace, and with a level of autonomy that they can manage, by requesting or not to contextual aids present in the virtual environment.

The realization of this project by the students aroused a great deal of enthusiasm and the organisation of the project team as an engineering office allowed a lot of interaction within the micro-enterprise thus formed. While the initial idea of involving students in the design of teaching tools was primarily intended to enable students to develop technical and pedagogical skills, rather, it was found that they placed greater emphasis on the opening of their field of expertise to virtual reality, video production, and team-based collaborative work experience. This awareness of the perceived gap in the interest of the project between teachers and students was only possible thanks to the support of the project by the pedagogical team and the associated desire to better understand how this type of pedagogical initiative is perceived by the main stakeholders. This better understanding of the expectations and motivations of the various people involved will make it possible to sustain and strengthen these actions and better support future initiatives. Beyond the development phase of the project, the challenge is now to optimise its integration into the existing educational context. The choice that has been made is to mix virtual and real experience, the virtual experience having the aim of allowing the experimentation of dangerous situations impossible to manage in a university context. Finding the right balance between real and virtual experience will require further efforts, so the evaluation of the users' perception of the project will continue

ACKNOWLEDGMENT

This research was financed by the French government IDEX-ISITE initiative 16-IDEX-0001 (CAP 20-25). In this context, the “Learn'in Auvergne”program was launched, with the aim of supporting educational teams in the design, implementation and evaluation of educational and digital innovations. The project presented in this article is the result of collective work. Thanks to J-J Diaz and B Beghein from the Electical Engineering team, Cauchy and D. Saint-Gérand from the Audiovisual Team, and the students in charge of the project: T. Clair, S. Coletto, C. Duplessy, T. Sauvagnat, F. Harduin De Grosville, B. Collet, M. Rumeo, M. Poillot, A. Da Costa

REFERENCES

[1] J-P; Toumazet, L. Hamon and C. Robles, “VirtuElec: A Tool Designed by and for Students for Training in Electrical Hazards”, International Conference on Mobile, Hybrid, and On-line Learning eLmL 2021, July 18, 2021 to July 22, 2021,Nice, France, pp. 21-24, ISSN: 2308-4367, ISBN: 978-1-61208-873-0.

[2] J. Radianti, T. A. Majchrzak, J. Fromm and I. Wohlgenannt, “A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda”, Computers & Education, vol. 147, 2020. ISSN 0360-1315,
https://doi.org/10.1016/j.compedu.2019.103778.

[3] L. Jensen and F. Konradsen, “A review of the use of virtual reality head-mounted displays in education and training”, Education and Information Technologies, vol. 23, no 4, pp. 1515–1529, 2018. http://dx.doi.org/10.1007/s10639-017-9676-0.

[4] J. Martín-Gutiérrez, C. E. Mora, B. Añorbe Díaz, and A. González-Marrero, “Virtual technologies trends in education”, EURASIA Journal of Mathematics, Science and Technology Education, vol. 13, no. 2, pp. 469–486, 2017.

[5] M. Němec, R. Fasuga, J. Trubač and J. Kratochvíl, "Using virtual reality in education", 15th international conference on emerging elearning technologies and applications- IEEE, pp. 1–6, 2017.

[6] M. Kassem, L. Benomran and J. Teizer, "Virtual environments for safety learning in construction and engineering: seeking evidence and identifying gaps for future research", Visualization in Engineering, vol. 5, no. 16, 2017. https://doi.org/10.1186/s40327-017-0054-1

[7] P. Wang, P. Wu, J. Wang, H.L. Chi and X. Wang, "A critical review of the use of virtual reality in construction engineering education and training", International Journal of Environmental Research and Public Health, vol. 15, no. 6, p. 1204, 2018.

[8] K. Patel, J.N. Bailenson, S. Hack-Jung, R. Diankov and R. Bajcsy, "The effects of fully immersive virtual reality on the learning of physical tasks", Proceedings of PRESENCE 2006: The 9th Annual International Workshop on Presence, August 24 – 26, Cleveland, Ohio, USA, 2006.

[9] M. Da Silva and M. H. Cascone, "Immersive Virtual Training for Substation Electricians", IEEE Virtual Reality (VR), pp. 451–52, 2017

[10] E.H. Tanaka, J.A. Paludo, C.S. Cordeiro, L.R. Domingues, E.V. Gadbem and A. Euflausino, "Using Immersive Virtual Reality for Electrical Substation Training", International Association for Development of the Information Society, 2015.

[11] D. Zhao, J. Lucas and W. Thabet, "Using Virtual Environments to Support Electrical Safety Awareness in Construction", Proceedings of the 2009 Winter Simulation Conference (WSC) pp. 2679–90, 2009. http://dx.doi.org/10.1109/WSC.2009.5429258

[12] T.M. Buriol, M. Rozendo, K. De Geus, S. Scheer and C. Felsky, "A Virtual Reality Training Platform for Live Line Maintenance of Power Distribution Networks", ICBL2009- International Conference on Interactive Computer Aided Blended Learning, pp. 1–13, 2009.

[13] M. Halpin, R. Halpin and P. Curtis, "Simulation-Based Electrical Safety Training," IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC), pp. 1137–1142, 2015.

[14] E. Hewson and K. W. Chung, "Beyond the VLE: transforming online discussion and collaboration through Microsoft Teams", Proceeding of the Online Open and Flexible Higher Education Conference of the European Association of Distance Teaching Universities, pp. 36-47, Madrid, October 2019.

https://pure.coventry.ac.uk/ws/portalfiles/portal/27637722/Proceedings_OOFHEC2019.pdf#page=4_

[15] D. Adamson et al., "Towards an Agile Approach to Adapting Dynamic Collaboration Support to Student Needs". Int J Artif Intell Educ, vol. 24, pp. 92–124, 2014. https://doi.org/10.1007/s40593-013-0012-6

[16] J. Baschin, D. Inkermann, T. Vietor, "Agile Process Engineering to support Collaborative Design", Procedia CIRP, Vol. 84, pp. 1035-1040, ISSN 2212-8271, 2019, https://doi.org/10.1016/j.procir.2019.05.010.

[17] C. Hillmann, "Comparing the Gear VR, Oculus Go, and Oculus Quest" Unreal for Mobile and Standalone VR". Apress, Berkeley, CA. C., pp. 141-167, 2019. https://doi.org/10.1007/978-1-4842-4360-2_5

[18] G. Barbieri and D. Andres Gutierrez, "A GEMMA-GRAFCET Methodology to enable Digital Twin based on Real-Time Coupling",Procedia Computer Science, vol. 180, ISSN 1877-0509, 2021, https://doi.org/10.1016/j.procs.2021.01.122.

Dynamic VLC Navigation System in Crowded Buildings

Manuela Vieira, Manuel Augusto Vieira, Paula Louro, Alessandro Fantoni
ADETC/ISEL/IPL,
R. Conselheiro Emídio Navarro, 1959-007
Lisboa, Portugal
CTS-UNINOVA
Quinta da Torre, Monte da Caparica, 2829-516,
Caparica, Portugal

e-mail: mv@isel.ipl.pt, mv@isel.pt, plouro@deetc.isel.pt, afantoni@deetc.isel.ipl.pt

Pedro Vieira
ADETC/ISEL/IPL,
R. Conselheiro Emídio Navarro, 1959-007
Lisboa, Portugal
Instituto das Telecomunicações
Instituto Superior Técnico, 1049-001,
Lisboa, Portugal
e-mail: pvieira@deetc.isel.pt

***Abstract*— This paper investigates the applicability of an intuitive risk of transmission wayfinding system in public spaces, virtual races, indoor large environments and complex buildings using Visible Light Communication (VLC). Typical scenarios include: finding places, like a particular shop or office, guiding users across different floors, and through elevators and stairs. The system is able to inform the users, in real time, not only of the best route to the desired destination, through a route without clusters of users, but also of crowded places. Data from the sender is encoded, modulated and converted into light signals emitted by the transmitters. Tetra-chromatic white sources are used providing a different data channel for each chip. At the receiver side, the modulated light signal, containing the ID and the 3D geographical position of the transmitter and wayfinding information, is received by a SiC optical sensor with light filtering and demultiplexing properties. Since lighting and wireless data communication is combined, each luminaire for downlink transmission becomes a single cell, in which the optical Access Point (AP) is located in the ceiling and the mobile users are scattered across the overlap discs of each cell, underneath. The light signals emitted by the LEDs are interpreted directly by the receivers of the positioned users. Bidirectional communication is tested. The effect of the location of the Aps is evaluated and a 3D model for the cellular network is analyzed. In order to convert the floorplan to a 3D geometry, a tandem of layers in a orthogonal topology is used, and a 3D localization design, demonstrated by a prototype implementation, is presented. Uplink transmission is implemented, and the 3D best route to navigate through venue is calculated. Buddy wayfinding services are also considered. The results showed that the dynamic VLC navigation system enables to determine the position of a mobile target inside the network, to infer the travel direction along the time, to interact with received information and to optimize the route towards a static or dynamic destination.**

Keywords- Visible Light Communication; Indoor navigation; Bidirectional Communication; Wayfinding; Optical sensors; Multiplexing/demultiplexing techniques.

I. INTRODUCTION

This paper is an extended version from the one presented in ALLSENSORS 2021 [1].

Optical wireless communication has been widely studied during the last years in short-range applications. Therefore, communications within personal working/living spaces are highly demanded. The availability of portable communication devices, such as smartphones and tablets increases the demand on mobile wireless connectivity. Several technologies have been investigated to provide wireless connections to the users in indoor and outdoor environments. Nowadays, indoor positioning methods are mainly based on Wi-Fi, Bluetooth, Radio-Frequency Identification (RFID) and Visible Light Communications (VLC) [2][3][4][5]. VLC is a data transmission technology [6] that can easily be employed in indoor environments since it can use the existing LED lighting infrastructure with simple modifications [7][8]. VLC can be regarded as a light based Wi-Fi, i.e., instead of radio waves uses visible light to transmit the data. It presents advantages when compared with the Wi-Fi, namely the invulnerability to the hackers since it does not penetrate through the wall, its high capacity and efficiency. Once lights are essential part of operating rooms, the VLC technology finds applications in a wide variety of fields like: in medical and healthcare, in airlines and aviation, in supermarkets and railway stations, in retail stores, in hidden communication or in Line-of -Sight (LoS) applications as in traffic control, vehicle to vehicle communication or smart street lighting.

The VLC systems use the wavelength range between 380 nm and 780 nm and the LEDs are used as light sources and transmitters. Therefore, the LEDs are twofold by providing illumination, as well as communication. LEDs are incoherent light sources and transmitting information can only be realized by the optical intensity change. Here, the On-OFF keying (OOK) modulation scheme is used. In the sequence, we propose to use modulated visible light, carried out by white tetra-chromatic low cost LEDs. The use of those LEDs provides different data channel for each chip offering the possibility of Wavelength Division Multiplexing (WDM), which enhances the transmission data rate. At the receiver side, the modulated light signal is received by a SiC photodetector, based on a tandem a-

SiC:H/a-Si:H pin/pin structure, which presents light filtering and demultiplexing properties decoding the received information [9] [10]. Here, when different visible signals are encoded in the same optical transmission path, the device multiplexes the different optical channels, performs different filtering processes (amplification, switching, and wavelength conversion) and finally decodes the encoded signals recovering the transmitted information.

Research is still necessary to design LED arrangements that can optimize communication performance while meeting the illumination constraints for a variety of large indoor layouts. Visible light can be used as an ID system in different places such as buildings and subways and can be employed for identifying the room number and its building. The main idea is to divide the space into spatial beams originating from the different ID light sources, and identify each beam with a unique timed sequence of light signals. Fine-grained indoor localization can enable several applications; in airports, supermarkets and shopping malls. Exact location of products can greatly improve the customer's shopping experience and enable customer analytics and marketing [11]. The signboards, based on arrays of LEDs, positioned in strategic directions to broadcast the information [12], are modulated acting as down- and up-link channels in the bidirectional communication.

In this paper, a LED-supported positioning and navigation VLC system is proposed. After the Introduction, in Section II, a VLC scenario for large environments is established, the emitters and receivers are characterized and the communication protocol presented. In Section III, the main experimental results are presented, the effect of the location of the optical Access Points (Aps) is evaluated and a model for the different cellular networks is analysed. Square and hexagon mesh are tested, and a 2D localization design, demonstrated by a prototype implementation, is presented. Uplink transmission is implemented and the 2D best route to navigate through venue calculated. In Section IV, the conclusions are drawn showing that the system makes possible to determine the position of a mobile target inside the network, to infer the travel direction along the time and to interact with information received.

II. VLC NETWORK

In the follow, the VLC network is described.

A. Self-localization

Self-localization is a fundamental issue since the person must be able to estimate its position and orientation (pose) within a map of the environment it is navigating. We consider the path to be a geometric representation of a plan to move from a start pose to a goal pose. Let us consider a person navigating in a 2D environment (Figure 1). Its non-omnidirectional configuration is defined by position (x, y) and orientation angle δ, with respect to the coordinate axes. $q(t) = [x(t), y(t), \delta(t)]$ denotes its pose at time t, in a global reference frame. In cooperative positioning systems, persons are divided into two groups, the stationary persons and the moving persons. Let us consider that $q_i(t,t')$ represents the pose of person *i* at time *t´* relative to the pose of the same person at time *t* and $q_{ij}(t)$ denotes the pose of person *j* relative to the pose of person *i* at time *t*. $q_i(t,t')$ is null for people standing still and non-zero if they move. These three types of information $q_i(t)$, $q_i(t,t')$ and $q_{ij}(t)$ compose the basic elements of a pose graph for multi-person cooperative localization.

Person-to-person spread to be the most common form of transmission of COVID-19, occurring mainly among people who are within 2m of each other for a prolonged period of time. This means people should stay the recommended distance apart from others. It also means people should avoid gathering in groups, crowded places and mass gatherings. So, in crowded building the routes to a specific place should avoid those regions. We consider that the risk of transmission exists if $q_{ij}(t)$ is less than 2 m. The system has to alert the users to stay away from those regions and to plan the better route to the desired wayfinding services. To estimate each person track the pure pursuit approach [13], [14] is used. The principle took into account the curvature required for the mobile receiver to steer from its current position (t_1) to its intended position (t_2). By specifying a look-ahead distance, it defines the radius of an imaginary circle. Finally, a control algorithm chooses a steering angle in relation to this circle. Then, this allows to iteratively construct the intermediate arcs between itself and its goal position as it moved, thus, obtaining the required trajectory for it to reach its objective position. To avoid the risk of transmission, in the same frame of time and in known crowded regions, $q_{ij}(t)$ is estimated and the steering angle readjusted.

B. Virtual scenario and Architecture

When we are looking for the shortest route to a place, we want to be guided on a direct, shortest path to our destination. Advances in wayfinding responds to our need to establish markers and identify patterns by creating spatial relationships between users and areas. Directional aids remain in force, but there is now a digital framework that goes beyond providing basic information. This expansion has encouraged the rise of multi-device connectivity that can tell users where they are (current point), where they need to be (goal point) and what they need to do to get there (path).

Interaction between planning, control, and localization is important. The localization (Where am I?) senses the environment and computes the user position, the planning (Where am I going?) computes the route to follow from the position and the control (How do I get there?) moves the user in order to follow the route. A destination can be targeted by user request to the Central Manager (CM).

The simulated scenario is an airport. Here, the traveler, equipped with a receiver, navigates from outdoor to indoor. It sends a request message to find the right track and, in the available time, he adds customized points of interest and halls to boarding. During his path, the passenger is advised how to reach its destination and the possibility to use location based advertising services. The requested information is sent by the emitters at the ceiling to its receiver. In Figure 1, the proposed architecture and scenario are illustrated.

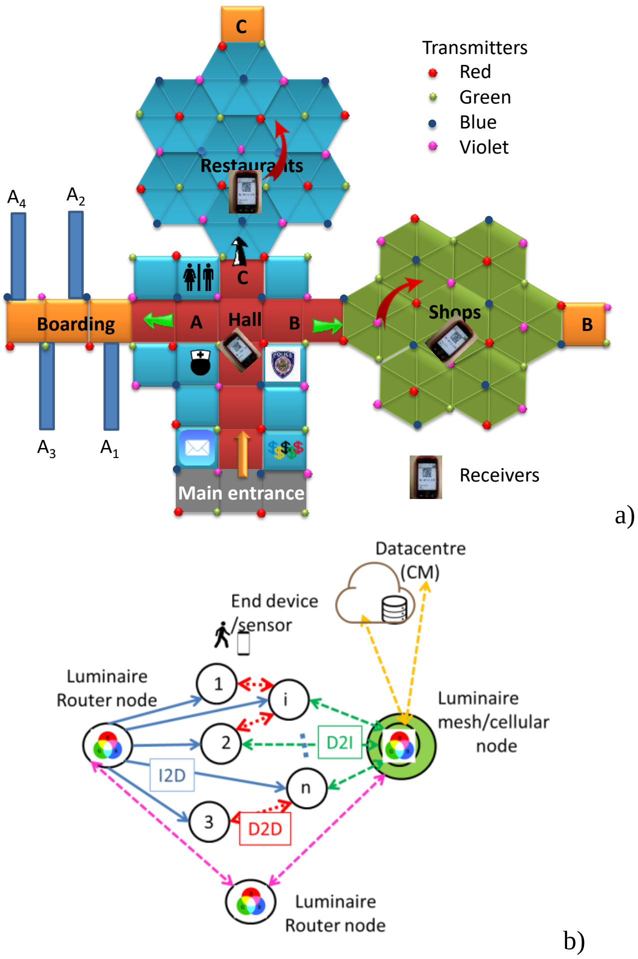


Figure 1 a) Optical infrastructure and indoor layout.b) a) Mesh and cellular hybrid architecture

Two topologies were considered: the square for the main hall and the hexagonal for the marketing zones (Figure 1a). Data from the infrastructure is encoded, modulated and converted into light signals emitted by the transmitters. Tetra-chromatic (Red, Green, Blue, Violet) white sources are used, providing a different data channel for each chip. At the receiver side, the modulated light signal, containing the ID and the 2D geographical position of the transmitter and wayfinding information, is received by a SiC photodetector with light filtering and demultiplexing properties. To synchronize the signals from multiple LEDs, the transmitters use different IDs, allowing the signal to be reconstructed at the receiver.

We propose a mesh cellular hybrid structure to create a gateway-less system without any external gateways needed. We propose this network configuration since it is wireless and ad-hoc. It spans all devices, is wire free, demonstrate resiliency to physical obstructions and adapt to changes in the transmission medium. Therefore, a mesh network is a good fit because it dynamically reconfigures itself and grows to the size of any installation. It is also a secure and trustworthy network [15]. As illustrated in Figure 1b, the luminaires, in this architecture, are equipped with one of two types of nodes: A "mesh" controller that connects with other nodes in its vicinity. These controllers can forward messages to other devices (I2D) in the mesh, effectively acting like routers nodes in the network. A "mesh/cellular" hybrid controller equipped with a modem provides IP base connectivity to the central manager services (CM). These nodes act as border-router and can be used for edge computing. Under this architecture, the short-range mesh network purpose is twofold: enable edge computing and device-to-cloud communication, by ensuring a secure communication from a luminaire controller to the edge computer or datacenter (I2CM), through a neighbor luminaire controller with an active cellular connection; and enable peer-to-peer communication ($I2I_{IP}$), to exchange information between smart devices.

Building a geometry model of interiors of buildings is complex since the interior structure has to be seen as an aggregation of several different types of objects (rooms, stairs, etc.) with different shapes. In the proposed architecture the logical model is easier since represents each room/crossing/exit with a node (Figure 1a), and a path as the links between nodes. The user positions can be represented as $P\ (x,\ y)$ by providing the horizontal positions $(x,\ y)$ The indoor route throughout the building is presented to the user by a responding message transmitted by the ceiling luminaires that work also either as router or mesk/celular nodes (Figure 1b). With this request/response concept, the generated landmark-based instructions help the user to unambiguously identify the correct decision point where a change of direction (pose) is needed, as well as offer information for the user to confirm that he/she is on the right way.

III. VLC LINK MODELS

A. *VLC dynamic navigation system (position and route control)*

To support people's wayfinding activities in unfamiliar indoor environments, a method able to generate ceiling landmark route instructions using VLC is proposed. The dynamic navigation system is composed of several transmitters (LEDs ceiling luminaires), which send the map information and path messages required to wayfinding. A mobile optical receiver, using joint transmission, extracts user location to perform positioning and, concomitantly, the transmitted data from each transmitter. To synchronize the signals from multiple LEDs, the transmitters use different ID´s, such that the signal is constructively at the receiver. Bidirectional communication between the emitters and the receivers is available in strategic optical access point (Li-Fi zone).The block diagram of the VLC system is presented in Figure 2.

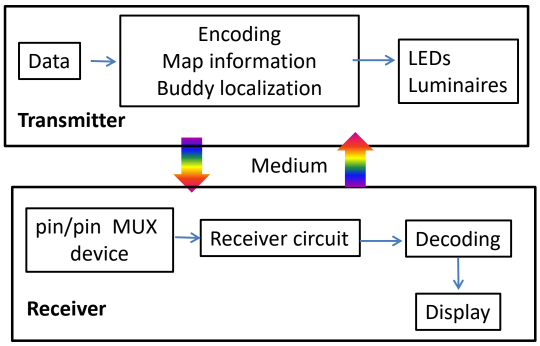


Figure 2. VLC block diagram of the VLC system.

The system is composed by the transmitter and the receiver modules located, respectively, at the infrastructures and at the mobile users. The VLC receiver, at the reception end of the communication link, is responsible to extract the data from the modulated light beam. It transforms the light signal into an electrical signal that is subsequently decoded to extract the transmitted information.

The principal components of the VLC system are the LEDs, which act as the communication sources and the SiC WDM devices that serve as receiving elements. Data from the sender is converted into an intermediate data representation, byte format, and converted into light signals emitted by the transmitter module. The data bit stream is input to a modulator where an ON–OFF KEYING (OOK) modulation is utilized. Here, a bit one is represented by an optical pulse that occupies the entire bit duration, while a bit zero is represented by the absence of an optical pulse.

The light signal is received by the WDM photodetector that detects the on/off states of the LEDs, generates a binary sequence of the received signals and convert data into the original format. The LEDs emit light when the energy levels change in the semiconductor diode. The wavelength depends on the semiconductor material used to form the LED chip. For data transmission, commercially available polychromatic white LEDs were used at the nodes of the network. On each node only one chip is modulated for data transmission and carries useful information while the others are only supplied with DC to maintain white colour illumination. Red (R; 626 nm), Green (G;530 nm), Blue (B; 470 nm) and violet (V; 390 nm) LEDs, are used [16][17]. Since lighting and wireless data communication is combined, each luminaire for downlink transmission becomes a single cell, in which the optical access point (AP) is located in the ceiling and the mobile users are scattered within the overlap discs of each cells underneath.

LEDs are modeled as Lambertian sources where the luminance is distributed uniformly in all directions, whereas the luminous intensity is different in all directions. The luminous intensity for a Lambertian source is given by Equation (1) [18]:

$$\mathrm{I}(\emptyset) = \mathrm{I}_N \cos(\emptyset)^m \quad (1)$$

Where m is the order derived from a Lambertian pattern, I_N is the maximum luminous intensity in the axial direction and ϕ is the angle of irradiance. The Lambertian order m is given by:

$$m = \frac{ln(2)}{ln\left(cos(\emptyset_{1/2})\right)} \quad (2)$$

For the proposed system, the commercial white LEDs were designed for illumination purposes, exhibiting a wide half intensity angle ($\phi_{1/2}$) of 60º. Thus, the Lambertian order m is 1.

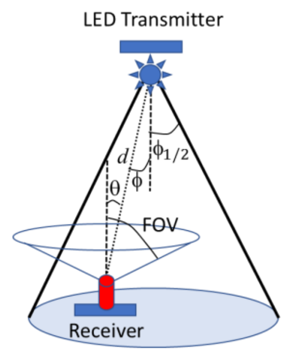


Figure 3. Geometry of the relative position of the transmitter and receiver units.

The light signal is received by the WDM photodetector that detects the on/off states of the LEDs, generates a binary sequence of the received signals and convert data into the original format. For simplicity we will consider a line of sight (LoS) connection for both VLC links, which corresponds to the existence of straight visibility between the transmitter and the receiver. In Figure 3, it is plotted the geometry of the transmitter and receiver relative position, with emphasis to the main parameters used for characterization of the LED source and the photodiode receiver (angles of irradiance and illumination, transmitter's semi-angle at half-power and field of view). The Lambertian model is used for LED light distribution and MatLab simulations are used to infer the signal coverage of the LED in the illuminated indoors space [19] [20].

The VLC photosensitive receiver is a two terminal double PIN photodetector based on a multilayer heterostructure, p-i'(a-SiC:H)-n/p-i(a-Si:H)-n. Two transparent front and back contacts are used [9]. The device presents high sensitivity and linear response in the visible range generating at the terminals a proportional photocurrent. It fast response enables the possibility of high speed communications. Modulated light supplied by the polychromatic LEDs is used for data transmission. The signals are encoded into colours intensities emitted by red, green, blue and violet LEDs. The generated photocurrent is processed using a transimpedance circuit and the proportional voltage is processed, by using signal conditioning techniques (adaptive bandpass filtering and amplification, triggering and demultiplexing), until the data signal is reconstructed at the data processing unit (digital conversion, decoding and decision) [16].

B. *Large Environments emitters and receivers*

Lighting in large environments is designed to illuminate the entire space in a uniform way. Ceiling plans for the LED array layout is shown in Figure 4.

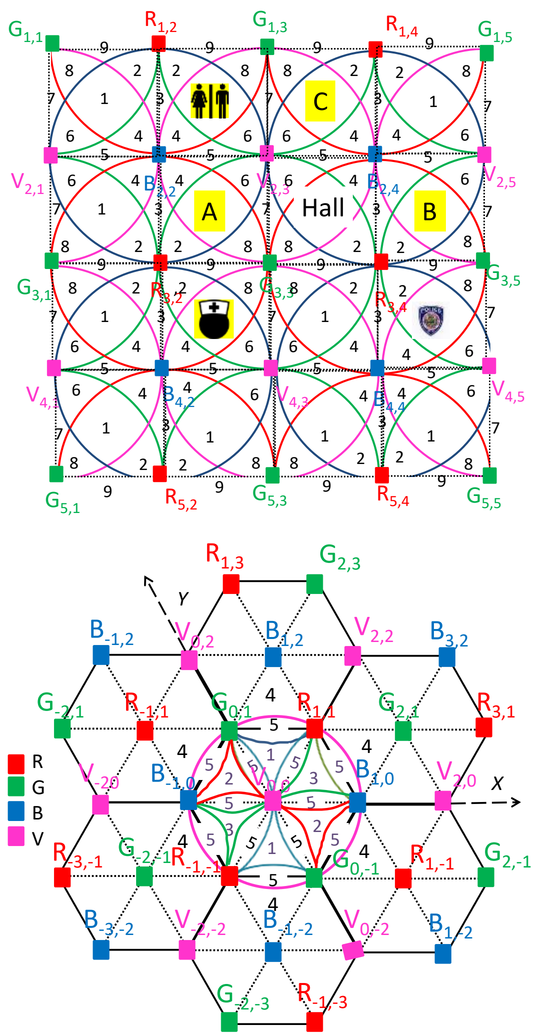


Figure 4. Illustration of the optical scenarios (RGBV >modulated LEDs spots). Cellular topologies: a) Clusters of cells in square topology . b) Clusters of cell in hexagonal topology.

Two topologies considered: the square, (Figure 4a) and the hexagonal (Figure 4b). In the square topology the cells have squares shapes while in the hexagonal one they are distributed to form a hexagonal shaped constellation. To receive the information from several transmitters, the receiver must be positioned where the circles from each transmitter overlap, producing at the receiver, a multiplexed (MUX) signal that, after demultiplexing, acts twofold as a positioning system and a data transmitter. In both, the grid sizes were chosen to avoid overlap in the receiver from adjacent grid points. The coverage map for a square unit cell is displayed in Figure 5. All the values were converted to decibel.

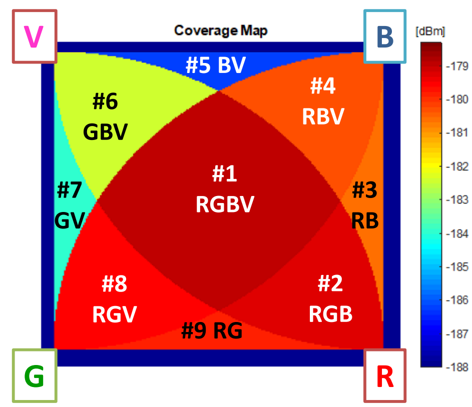


Figure 5. Illustration of the coverege map in a square unit cell.

Friis' transmission equation is frequently used to calculate the maximum range by which a wireless link can operate. The coverage map is obtained by calculating the link budget from the Friis Transmission Equation [21]. The Friis transmission equation relates the received power (P_R) to the transmitted power (P_E), path loss distance (L_R), and gains from the emitter (G_E) and receiver (G_R) in a free-space communication link.

$$P_{R\,[dBm]} = P_{E\,[dBm]} + G_{E\,[dB]} + G_{R\,[dB]} - L_{R\,[dB]} \qquad (3)$$

Taking into account Figure 3, the path loss distance and the emitter gain will be given by:

$$L_{R\,[dB]} = 22 + 20 \ln \frac{d}{\lambda} \qquad (4)$$

$$G_{E\,[dB]} = \frac{(m+1)A}{2\pi d_{E-R}^2} I(\emptyset) \cos(\theta) \qquad (5)$$

With A the area of the photodetector and d_{E_R} the distance between each transmitter and every point on the receiver plane.

The receptors act as active filters [9] [16]. Due to their filtering properties the gains are strongly dependent on the wavelength of the pulsed LEDs. Gains of 5, 4, 1.7 and 0.8 were used, respectively, for the R, G, B and V LEDs.

The coverage map, Figure 5, was obtained by calculating the link budget using the Equation (3). The input parameters are displayed in Table 1. All the values were converted to decibel.

TABLE 1. LINK BUDGET INPUT.

Variable	Value			
	Red LED	**Green LED**	**Blue LED**	**Violet LED**
I_N(mcd)	730	650	800	900
G_E(dB)	Equation (5)			
G_R	5	4	1.7	0.8
L_R(dB)	Eq. (4)			

Users in different locations are served simultaneously by the same transmitter leading to a fine grained implementation. Due to the overlapping coverage area of adjacent Aps joint transmission exists. In Table II, and for both topologies, the overlap regions below each AP (footprints) are displayed. Results show that the received power in each cell depends on the receiver position. Nine separated levels were found, in the square topology, and correspond to the nine possible combinations of the pulsed LEDs framed at corners of the unit cell (Figure 4a). The same occurs in the hexagonal topology (Figure 4b) where five different levels are detected.

TABLE 2. FINE-GRAINED TOPOLOGIES: FOOTPRINT REGIONS.

Footprint regions	Square topology	Hexagonal topology
#1	RGBV	RGV
#2	RGB	GBV
#3	RB	RBV
#4	RBV	RGB
#5	BV	RGBV
#6	GBV	-
#7	GV	-
#8	RGV	-
#9	RG	-

In both topologies, each node, $X_{i,j}$, carries its own color, X, (RGBV) as well as its ID position in the network (i,j). The overlap regions (footprints) are pointed out in Figure 4 and reported in Table 1. The device receives multiple signals, finds the centroid of the received coordinates and stores it as the reference point position (# in Figure 5).

C. *Modulation technique*

An on-off keying (OOK) modulation scheme was used providing a good trade-off between system performance and implementation complexity.

The OOK transmits data by sequentially turning *on* and *off* the LED providing digital dimming support. To create a communication protocol to ensure the required system performance and overcome the technology constraints, a 32 bits data frame was designed. Three control fields, one for synchronism (Sync.) and two for the identification of the cell (ID) begin each frame. This sequence is followed by a fourth block that is for the payload, as it is shown, for both topologies, in Figure 6. A stop bit is used at the end of each frame.

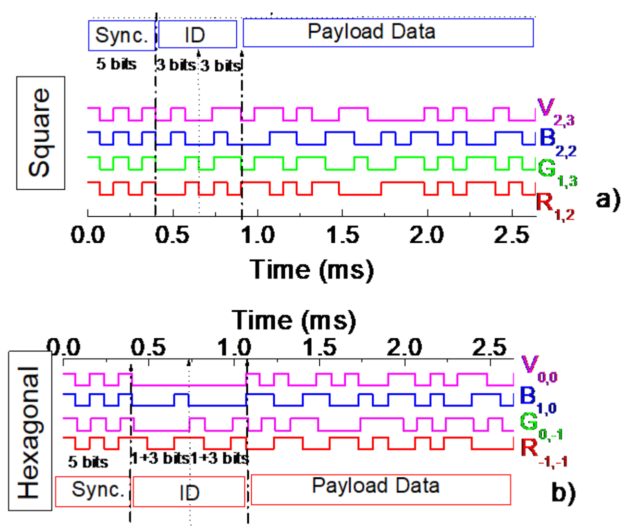


Figure 6. Data frame structure. Representation one encoded message, in a time slot from the array in the network: a) Square topology; R_{12} ; $G_{1,3}$; $B_{2,2}$ and $V_{2,3}$ are the transmitted node packet. B) Hexagonal topology; $R_{-1,-1}$; $G_{0,-1}$; $B_{1,0}$ and $V_{0,0}$ are the transmitted node packet.

The first five bits in the frame are used for time synchronization. The same synchronization header [10101], in an ON-OFF pattern, is imposed simultaneously to all emitters. Each colour signal (RGBV) carries its own ID-BIT, so, the next bits give the coordinates of the emitter inside the array ($X_{i,j}$). Cell's IDs are encoded using a binary representation for the decimal number. In the square topology (Figure 4a) six bits are used: the first three for the binary code of the line and the other three for the column. In the hexagonal topology, 60º Cartesian coordinates were applied (Figure 4b). Here, an extra bit was added at the beginning of the binary code to represent the number's sign: setting that bit to *0* is for a positive number, and setting it to *1* is for a negative number. The remaining bits in the number indicate the absolute value. So, the next eight bits (ID) are assigned, respectively, to the *x* and *y* coordinate (*I,j*) of the emitter in the array (Figure 3b). For both, the last bits, in the frame, are reserved for the message send by the X_{ij} node (payload data). With this information, the method will give an unique answer, *i.e.*, the location of the receiver in the array ($X_{i,j}$) and the broadcast information.

Results show that in the square network $R_{1,2}$, $G_{1,3}$, $B_{2,2}$ and $V_{2,3}$ are the transmitted node packets, in a time slot, from the unit cell where the restrooms are located (Figure 4a). In the hexagonal network, the nodes $R_{-1,-1}$, $G_{0,-1}$, $B_{1,0}$ and $V_{0,0}$, at the first ring of the restaurant zone (Figure 4b) are the transmitters.

IV. Experimental Results and Discussion

In this section, the main results are presented and discussed.

A. Decoding

In a data frame, the MUX signal at the receiver, due to the joint transmission of four R, G, B and V optical signals, is presented in Figure 7. The data acquisition was obtained under environment light. On the top, the bit sequence used to drive the LEDs is displayed. This sequence allows all the *on/off* sixteen possible combinations of the four input channels (2^4).

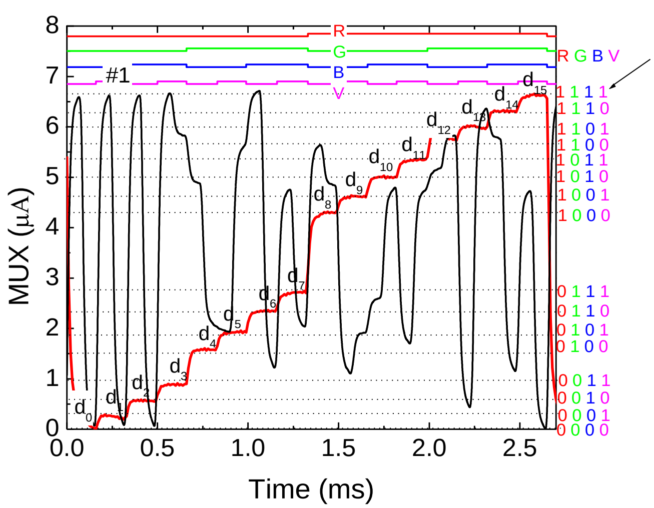


Figure 7. MUX signal of the calibrated cell. On the top the transmitted channels packets [R, G, B, V] are depicted. A received MUX signal is also superimposed to exemplify the decoding process.

Results show that the code signal presents as much separated levels as the *on/off* possible combinations of the input channels, allowing decoding the transmitted information [16]. All the ordered levels (d_0-d_{15}) are pointed out at the correspondent levels, and are displayed as horizontal dotted lines. In the right hand side the match between MUX levels and the 4 bits binary code assigned to each level is shown. For demonstration of the decoding technique, a signal received, in the same frame of time, when the receiver is in the main the hall (Figure 4a) underneath position #1, is also added. Comparing the calibrated levels (d_1-d_{15}) with the different assigned 4-digit binary code, the decoding is straightforward.

After decoding each input channel, and taking into account the frame structure (Figure 6), the position of the receiver and its ID in the network is revealed [22] [23]. The footprint position comes directly from the synchronism block, where all the received channels are simultaneously *on* or *off*. For instance, in any footprint #1 the maximum amplitude detected (see arrow) corresponds to the binary word [1111], meaning that the received information comes from the overlap of the four input channels.

B. Positioning

In Figure 8, for the same frame time and in three successive instants (t_1, t_2, t_3), the received MUX signals, when the receiver is in the main hall and moves from #5 to #9, through #1 (Figure 4a), confirm the decoding process.

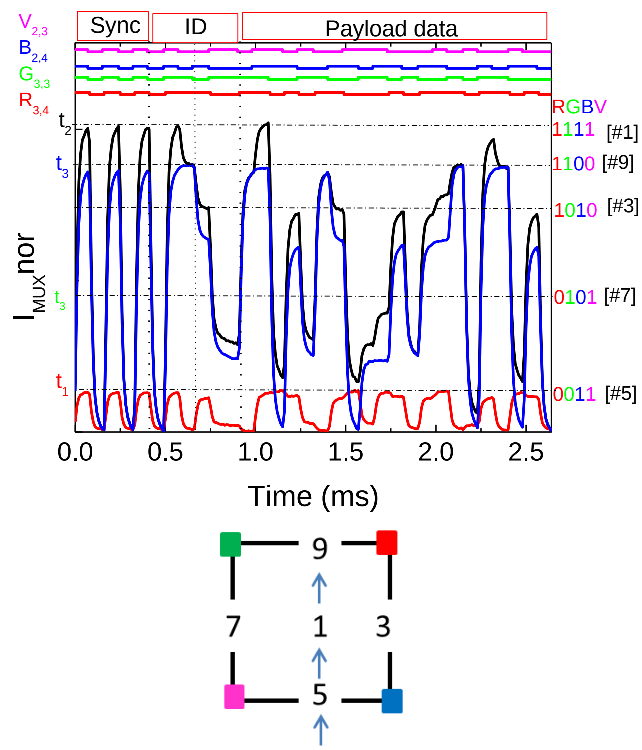


Figure 8. Fine-grained indoor localization and navigation in successive instants (t_1, t_2 and t_3): MUX/DEMUX signals in a point-to-point path across the main hall (#5>#1> #9) at the. On the top the complete transmitted channels packets [R, G, B, V] are decoded.

Decoding, when the four channels overlap (#1), is set on the top of the figure to direct into the packet sent by each node. The footprint position comes directly from the synchronism block. For instance, if the maximum amplitude detected corresponds to the binary word [0011], it means that it has only received the overlap transmission from the blue and the violet channels (footprint #5). In the right side of the figure the levels ascribed to #1, #3, #5 and #9 are point it out.

Each decoded message carries, also, the transmitter's node address. So, the next block of six bits, in the square topology (or eight in the hexagonal one), gives de ID of the received node. In Figure 8, in position #5 the network location of the transmitters are $B_{2,4}$ [010;100] and $V_{2,3}$ [010;011] and while in #1 the assigned transmitters are $R_{3,4}$, $G_{3,3,}$ $B_{2,4}$ and $V_{2,3.}$ The last block is reserved for the transmission of the advertising (Payload data). The stop bit (0) is used always at the end of each frame.

C. *Point-to-point Route*

To compute the point-to-point along a path, we need the data along the path. The input of the aided navigation system is the coded MUX signal, and the output is the system state decoded at each time step (Δt). As a proof of concept, in the lab, a navigation data bit transition was tested by moving the receiver along known pattern path.

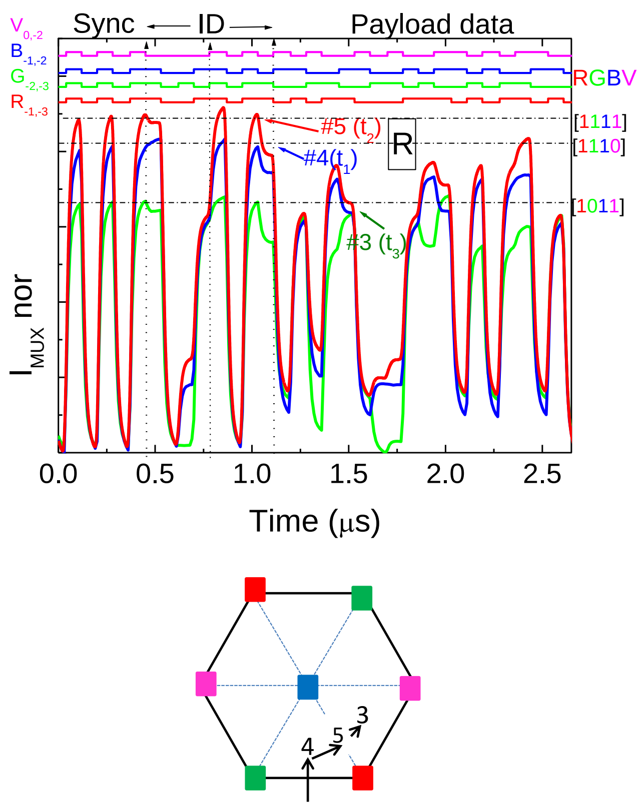


Figure 9 Fine-grained indoor localization and navigation in successive instants instants (t_1, t_2 and t_3). Signal acquisition through the restaurants area (R). On the top the transmitted channels packets are decoded [R, G, B, V].

In this example (Figure 8) at t_1 the user enters the hall by line #5, it goes to position #1 at t_2 and it chooses the boarding terminal C (#5) at t_3. Results show that, as the receiver moves between generated point regions, the received information pattern changes. Between two consecutive data sets, there is a navigation data bit transition (channel is missing or added). We observe that when the receiver moves from #1 to # 9 (Figure 8) two different ID channels are missing ($B_{2,4}$ and $V_{2,3}$). Here, the 4-bynary bit code has changed from [1111] to [1100].

In Figure 9, a path across the hexagonal topology was also tested. Here, the receiver enters the restaurants area (#4> #5> #3).

Main results from both topologies show that fine grained localization is achieved by detecting the wavelengths of the received channels in each region. The location and path of a mobile receiver was obtained based on the LED-based navigation system. In an orthogonal layout (hall), the square topology is the best. It allows crossroads and the client can walk easily in the horizontal, vertical or both directions. In concentric layouts, to fill all the space with hexagon presents advantages (restaurants, and shops areas). Here, the client can move around and walk between the different rings toward the outside region.

D. *Bidirectional Communication*

The VLC is a wireless broadband technology. It provides multiuser with simultaneous access communication. All the nodes communicate with each other through a centralized controller and the signal is used to establish a point-to-point link between the transmitters and the receiver.

Bidirectional communication between VLC emitters and receivers at a handheld device can be established through a control manager linked to a signboard. Each ceiling lamp broadcasts a message with its ID and useful information which is received and processed by the receiver. Using a white polychromatic LED as optical source for uplink, the receptor sends to the local controller a "request" message with its location (ID) and adds its needs for the available time (Payload data). For route coordination, the local controller emitter sends the downlink "response" message. In Figure 10, the MUX signal assigned to a "request" and a "response" message are displayed. In the top, the decoded information is presented. In the right side, the match between the MUX signal and the 4-binary code is pointed out.

Here, in a time slot, the traveler, in position #3 ($R_{3,2}$, $B_{2,2}$), sends to the central controller the message "request" in order to add the points of interest (boarding or the right track). After that it is advised, through a "response" message, that the request was received, how to reach its destination in time and how to use location based services.

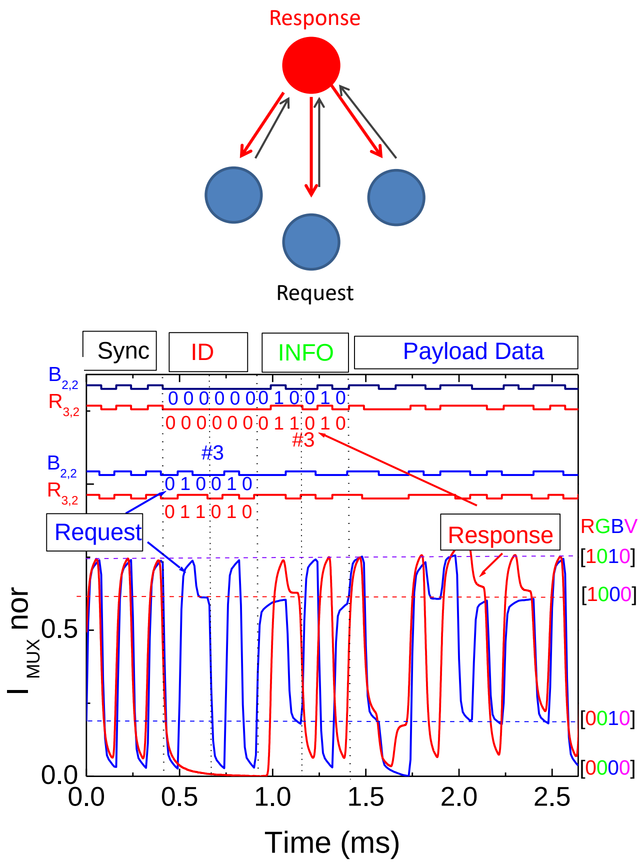


Figure 10. Bidirectional communication: MUX/DEMUX signals assigned to a "request" and a "response" message. On the top the transmitted channels packets [Xi,j] are decoded.

In bidirectional VLC transmission we use different codes in the frame structure for uplink and downlink. Taking into account the frame structure (Figure 6), results show that the codification of both signals is synchronized (Sync). The "request" message includes the complete address of the traveller (Sync+ID) and the help need (Payload Data). In the "response" message the block (ID), in a pattern [000000], means that a response message, from the local manager, is being sent. The next block (6 bits) identifies the address (INFO) for which the message is intended and finally in the last block appears the requested information (Payload Data). Here, the emitter controller [000000] responds to a request of a passenger located in position # 3 ($R_{3,2}$, $B_{2,2}$) and sends back the requested information.

E. Multi-person cooperative localization

An unforeseen pandemic is facing the world caused by a corona virus known as SARS-CoV-2. Transmission of the coronavirus is possible indoors, especially when people spend extended periods in crowded and poorly ventilated rooms [24]. The widely accepted main transmission mechanism is through droplet borne pathways.

Person-to-person spread to be the most common form of transmission of COVID-19, occurring mainly among people who are within 2m of each other for a prolonged period of time. This means people should stay the recommended distance apart from others. It also means people should avoid gathering in groups, crowded places and mass gatherings. So, in crowded building the routes to a specific place should avoid those regions. We consider that the risk of transmission exists if the distance between two users is less than 2 m. The system has to alert the users to stay away from those regions and to plan the better route to the desired wayfinding services.

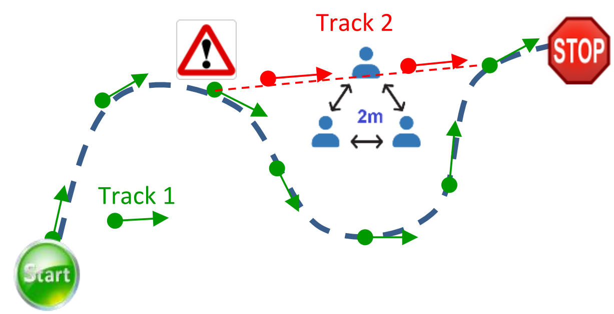


Figure 11 Safety close person virtual track.

The existence of congested zones can be locally detected by the "mesh / cellular" hybrid controller (Figure 1b), which is also equipped with a modem providing IP base connectivity to the central manager. The hybrid controller integrates the number of requests and individual positions, received during the same time interval. Once the individual positions are known, the relative positions are calculated. If the relative position is less than 2 m, a contamination risk locally exists and an alert message is send for the users and the CM is informed. This alert allows the CM to recalculate, in real time, the best route for the users that request wayfinding services avoiding crowded regions as exemplified in Figure 11.

V. CONCLUSIONS

A VLC multi-person cooperative localization dynamic LED-assisted navigation system for large indoor environments was proposed. For lighting, data transmission and positioning, white LEDs were used. A SiC optical MUX/DEMUX mobile receiver decodes the data and based on the synchronism and ID of the joint transmitters infers its location, point-to-point path, timing and user flows.

A VLC scenario was established and the communication protocol presented. Bidirectional communication between the infrastructures and the mobile receivers were analysed. Two cellular networks were tested and compared: square and hexagonal. Main results show that, for both topologies, the location of a mobile receiver, concomitant with data transmission is achieved. The LED-aided VLC navigation system makes possible to determine the position of a mobile target inside the network, to infer the travel direction along the time and to interact with received information.

Minding the benefits of VLC, it is expected that this type of communication will have an important role in positioning applications. Moving towards real implementation, the performances of such systems still need to improve. As a future goal, we plan to finalize the embedded application, for experimenting in several network layouts. Effects as synchronization, shadowing and ambient light noise will be minimized by distributing lighting sources (MIMO techniques) to optimize the coverage.

Acknowledgements

The work was sponsored by FCT – Fundação para a Ciência e a Tecnologia, within the Research Unit CTS – Center of Technology and Systems, reference UIDB/00066/2020. The project IPL/IDI&CA/2020/Geo-Loc/ISEL, is also acknowledge.

References

[1] M. Vieira, M. A. Vieira, P. Louro, P. Vieira, and A. Fantoni "Wayfinding Services in Crowded Buildings Through Visible Light" The Sixth International Conference on Advances in Sensors, Actuators, Metering and Sensing, ALLSENSOR2021, Copyright: Copyright (c) IARIA, 2021, July 18, 2021, ISSN: 2519-836X, ISBN: 978-1-61208-875-4, pp. 1-8.

[2] C. Yang and H. R. Shao,, "WiFi-based indoor positioning," IEEE Commun. Mag., vol. 53, no. 3, pp. 150–157, Mar. 2015.

[3] X. Y. Lin, T. W. Ho, C. C. Fang, Z. S. Yen, B.J. Yang, and F. Lai, "A mobile indoor positioning system based on iBeacon technology," in Proc. Int. Conf. IEEE Eng. Med. Biol. Soc., pp. 4970–4973, 2015.

[4] C. H. Huang, L. H. Lee, C. C. Ho, L. L. Wu, and Z. H. Lai, "Real-time rfid indoor positioning system based on kalman filter drift removal and heron-bilateration location estimation," IEEE Trans. Instrum. Meas., vol. 64, no. 3, pp. 728–739, Mar. 2015.

[5] N. U. Hassan, A. Naeem, M. A. Pasha, T. Jadoon, and C. Yuen, "Indoor positioning using visible led lights: A survey,"ACM Comput. Surv., vol. 48, pp. 1–32 , 2015.

[6] E. Ozgur E. Dinc, O. B. Akan, "Communicate to illuminate: State-of-the-art and research challenges for visible light communications," Physical Communication 17 pp. 72–85, 2015.

[7] D. Tsonev, H. Chun, S. Rajbhandari, J. McKendry, S. Videv, E. Gu, M. Haji, S. Watson, A. Kelly, G. Faulkner, M., Dawson, H. Haas, and D. O'Brien, "A 3-Gb/s single-LED OFDM-based wireless VLC link using a Gallium Nitride μLED," IEEE Photon. Technol. Lett. 26 (7), pp.637–640, 2014.

[8] D. O'Brien, H. L. Minh, L. Zeng, G. Faulkner, K. Lee, D. Jung, Y. Oh, and E. T. Won, "Indoor visible light communications: challenges and prospects," Proc. SPIE 7091, 709106, 2008.

[9] M. Vieira, P. Louro, M. Fernandes, M. A. Vieira, A. Fantoni and J. Costa, "Three Transducers Embedded into One Single SiC Photodetector: LSP Direct Image Sensor, Optical Amplifier and Demux Device," Advances in Photodiodes InTech, Chap. 19, pp. 403-425, 2011.

[10] M. A. Vieira, P. Louro, M. Vieira, A. Fantoni, and A. Steiger-Garção, "Light-activated amplification in Si-C tandem devices: A capacitive active filter model," IEEE sensor journal, 12, no. 6, pp. 1755-1762, 2012.

[11] A. Jovicic, J. Li, and T. Richardson, "Visible light communication: opportunities, challenges and the path to market," Communications Magazine, IEEE, vol. 51, no. 12, pp. 26–32, 2013.

[12] S. B. Park, D. K.Jung, H. S. Shing, D. J. Shing, Y. J. Hyun, K. Li, and Y. J. Oh, "Information broadcasting system based on visible light signboard," presented at Wireless and Optical Communication 2007, Montreal, Canada, 2007.

[13] R. Rajesh, "Dynamic Vehicle and control" Mechanical Engineering Series, ISBN 978-1-4614-1433-9. 2012.

[14] J. Ackermann, J. Guldner, W. Sienel, R. Steinhauser, and V. Utkin, "Linear and Nonlinear Controller Design for Robust Automatic Steering" IEEE Transactions on Control Systems Technology (Volume: 3, Issue: 1, Mar 1995), pp. 132 – 143, DOI: 10.1109/87.3,70719, 1995.

[15] A. Yousefpour, C. Kong, J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey", Journal of Systems Architecture,Volume 98, pp. 289-330, 2019.

[16] M. Vieira, M. A. Vieira, P. Louro, P. Vieira, A. Fantoni, "Fine-grained indoor localization: optical sensing and detection," Proc. SPIE 10680, Optical Sensing and Detection V, 106800H. 9 May 2018.

[17] M. Vieira, M. A. Vieira, P. Louro, P. Vieira, A. Fantoni, "Light-emitting diodes aided indoor localization using visible light communication technology," Opt. Eng. 57(8), 087105, 2018.

[18] Y. Zhu, W. Liang, J. Zhang, and Y. Zhang, "Space-Collaborative Constellation Designs for MIMO Indoor Visible Light Communications," IEEE Photonics Technology Letters, vol. 27, no. 15, pp. 1667–1670, 2015.

[19] Y. Qiu, H-H. Chen and W-X. Meng, "Channel modeling for visible light communications—a survey", Wirel. Commun. Mob. Comput. 2016; 16:2016–2034, 2016.

[20] S. Imran Raza, S. Jabeen, R. Chaudhry, S. A. Hussain, A. Saeed, M. S. Bhatti and M. H. Raza, "Optical Wireless Channel Characterization For Indoor Visible Light Communications", Indian Journal of Science and Technology, Vol 8 (22), DOI: 10.17485/ijst/2015/v8i22/70605, 2015.

[21] H. T. Friis, "A note on a simple transmission formula" Proc. IRE34, pp. 254–256, 1946.

[22] M. A. Vieira, M. Vieira, P. Louro,V. Silva, and P. Vieira, "Optical signal processing for indoor positioning using a-SiCH technology," Opt. Eng. 55 (10), 107105, 2016.

[23] M. Vieira, M. A. Vieira, P. Louro, and Vieira, "Positioning and advertising in large indoor environments using visible light communication," Opt. Eng. 58(6), 066102, 2019.

[24] C. Iddon, A. Hathaway and S. Fitzgerald, "CIBSE COVID-19 Ventilation Guidance", CIBSE, 2020.

Development of a VR Simulator for Speed Sprayer Operation Training

Yu Tanaka, Oky Dicky Ardiansyah Prima, Katsuyoshi Hotta, Kanayo Ogura, and Koichi Matsuda
Graduate School of Soft. and Info. Science, Iwate Prefectural University
152-52 Sugo, Takizawa, Iwate, Japan
email: g231t024@s.iwate-pu.ac.jp, prima@iwate-pu.ac.jp, g236q004@s.iwate-pu.ac.jp, {ogura_k, matsuda}@iwate-pu.ac.jp

Shoichi Yuki
YAMABIKO Corporation
10-2 Sugo, Takizawa, Iwate, Japan
email: s_yukixx@yamabiko-corp.co.jp

Abstract— **Speed (boom) sprayers, the agricultural chemical spraying vehicles, are generally used in orchards of grapes and apples to efficiently control pests. Speed sprayers require skill to operate, and improper operation can result in pesticides drifting off the field. It also requires significant time to train the operator, as adjustments to travel speed, pressure, and nozzle selection need to be considered. Meanwhile, to make the operation of the speed sprayer easier and safer, improvements to the control panel in the cockpit have become necessary. In this study, we developed a simulator of a speed sprayer based on Virtual Reality (VR) and conducted analysis of the driving and operation of the speed sprayer. Experiments using the developed simulator with ten subjects showed that the trend of the subjects' steering wheel rotation angles was the same as that of an actual speed sprayer driver. Furthermore, the head posture when operating the spray button were also the same as that of the driver operating the actual speed sprayers. While there is room for improvement in terms of the real motion cueing and field of view provided to the user, this simulator allowed us to practice the basic operations necessary for pesticide spraying.**

Keywords-vr-simulator; speed sprayer; virtual reality; training simulator; deep feedforward neural network.

I. INTRODUCTION

Orchards have become an important core sector of agriculture in terms of production value, land use, and development of local agriculture in Japan. Many orchards are sprayed with pesticides. Speed sprayers (boom sprayers) have been introduced to improve the efficiency and labor saving of pesticide spraying in orchards. However, many orchardists are aging and depopulating, making succession training and securing labor a major social problem. This study extends our previous work on the analysis of the development of a Virtual Reality (VR) simulator for speed sprayers [1].

The first domestically produced speed sprayer was made in 1957 in Japan. Early speed sprayers were towed by tractors, but in 1965, they were replaced by self-propelled ones. There are several issues with the use of speed sprayers in terms of occupational safety and environmental protection. Speed sprayers used in orchards increase the amount of pesticides sprayed due to the large spray area. Since the Japanese Ministry of Health, Labor and Welfare introduced a positive list system for pesticide residues in food, feed additives, and veterinary drugs [2], measures to prevent pesticide drift into adjacent fields have become urgent.

Many studies have been conducted to minimize the effects of pesticide drift. These studies include the improvement of spray nozzles, the development of remotely operated sprayers, and the development of unmanned sprayers. The low-drift nozzle could not compensate for the increase in spray drift due to the increase in sprayer speed [3]. The sprayer travel speed had a significant influence on the drift values. However, the coarse spraying did not result in that much higher drift [4]. While various unmanned pest control machines have been developed for more accurate spraying [5][6][7], they have not been widely adopted.

To operate a speed sprayer, the relationship between travel speed, pressure, and nozzle selection must be considered. This operation, therefore, requires a skilled operator. Driving skills are expected in areas where the vehicle must pass through a narrow path between branches. On the other hand, the speed sprayer also needs to be improved as well, since it is difficult to see the surrounding from inside the vehicle and the operation panel is complicated. To solve these problems, there is a need to develop a speed sprayer simulator. This simulator will enable improvement in the work capacity of the spraying sequence and the interface for operating the sprayer. A more efficient interface needs to be developed to simplify tasks that require technical skills.

Research and development of driving simulators are now common because they allow drivers to examine any scenario or situation. During the simulation of each driving task, sensory-based operation information can be analyzed. Attempts were made to develop a tractor driving simulator to prevent accidents during work [8][9]. Fujimoto et al. (2016) simulated the spray distribution of a speed sprayer using the coverage of water-sensitive paper placed in the path of travel [10]. By integrating these studies, the training simulator can be constructed to improve the operation skills of speed sprayers, but such an effort is still developing.

In this study, we build a speed sprayer simulator using VR and analyze the driving and operation of the speed sprayer. To provide the user with the experience of operating a speed sprayer, the control buttons, steering wheel, and other control interfaces are made to resemble those of an actual speed sprayer. Information obtained from the VR simulator, such as button operations, head posture while driving, and steering wheel rotation, is recorded. Cameras were installed in the cockpit of the actual speed sprayer to obtain the same information through automatic image recognition. Finally, we analyze the data from the VR simulator and the actual field operation of the speed sprayer to evaluate the usability of the simulator.

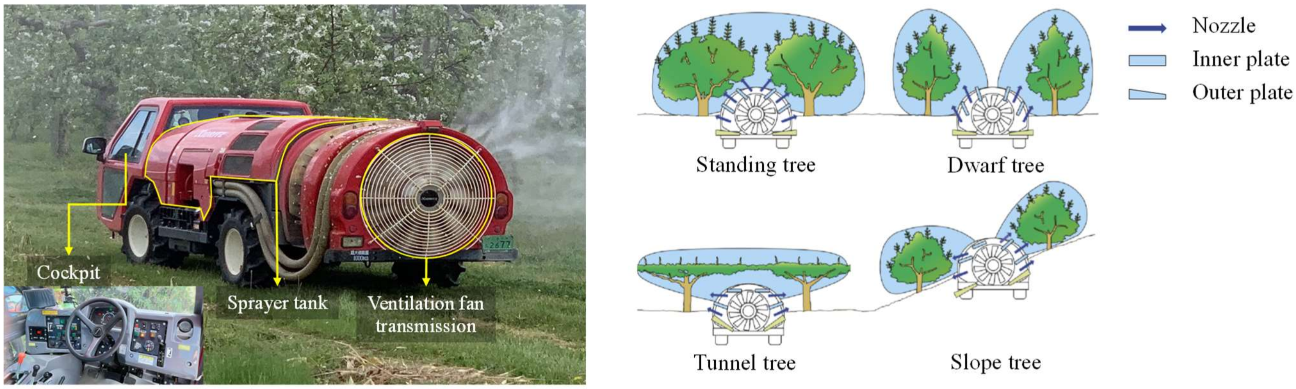


Figure 1. The electric speed sprayer (SSV1091FSC) manufactured by the YAMABIKO Corporation used in this study.

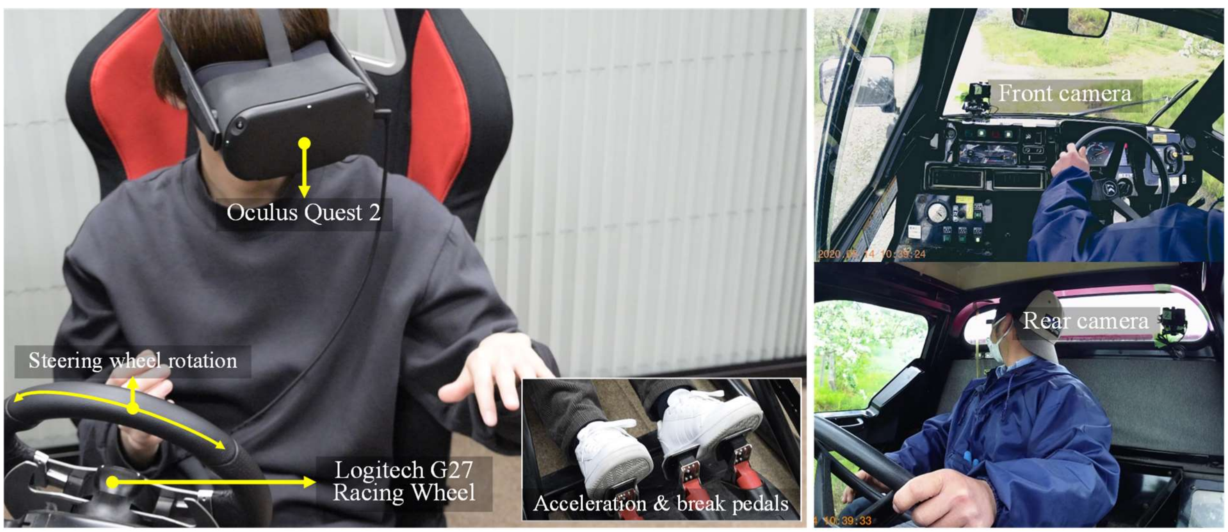


Figure 2. Sensors used in our VR simulator.

The rest of this paper is organized as follows. Section II discusses the speed sprayer in an orchard. Section III describes the proposed speed sprayer simulator. In Section IV, we describe our experiments with the simulator and the results. Finally, Section V summarizes the results of this study and discusses future perspectives.

II. Speed Sprayer in Orchards

Air blast sprayers are used to apply pesticides, plant growth regulators, and foliar nutrients to trees in orchards. Air-blast sprayers can be adapted to different orchard conditions by adjusting the fluid and air delivery systems [11]. A Speed sprayer is an air blast sprayer used for spraying in apple orchards, vineyards, etc. This type of sprayer sprays the pesticide close to the tree, thus reducing the loss of pesticide.

The Speed sprayer is capable of adapting to the spray pattern of the tree. There are four types of tree patterns in orchards: the standing tree, the dwarf tree, the tunnel tree, and the slope types. Figure 1 shows the speed sprayer (SSV1091FSC) manufactured by the YAMABIKO Corporation, which can carry 1000L of pesticide [12]. The speed sprayer can spray pesticides on the entire tree by adjusting the nozzle and the plates that determine the spraying direction as follows.

(1) *Standing trees* : Direct the nozzles and inner plates radially and the outer plate horizontally.
(2) *Dwarf trees* : Direct the nozzles and inner plates to the top but close the top two nozzles.
(3) *Tunnel trees* : Level the topmost plate to block the wind from moving upward and raise the outer plate.
(4) *Slope trees* : Tilt the nozzle and plate in the direction of the slope.

The operation of the speed sprayer is complex. The effective spraying operation depends on the skill of the sprayer operator. Skilled operators continuously check the condition of the trees and adjust the sprayer appropriately.

Several studies have developed simulators to train operators to control agricultural vehicles in orchards [1][13]. Through the simulator, users can experience a variety of training courses, which is expected to improve their driving skills. There have also been attempts to improve the efficiency

TABLE I. SENSORS USED IN THIS STUDY

Sensors	Data	Output
Logitech G27	Steering wheel rotation	-180.0° ~ 180.0°
	Pedal:	
	- Accelerator	0.0 (low) ~ 1.0 (high)
	- Brake	0.0 (low) ~ 1.0 (high)
Oculus Quest 2	Head pose	
	- Position	x, y, z (m)
	- Pitch	-180.0° ~ 180.0°
	- Yaw	-180.0° ~ 180.0°
	- Roll	-180.0° ~ 180.0°
	3D Hand landmark	19 points
GoPro Hero8	Image	3,840x2,160px/60fps (max)

of spraying by introducing machine vision, and its effectiveness has been confirmed in spraying discrete targets [14].

III. SPEED SPRAYER SIMULATOR FOR OPERATION TRAINING

In developing a simulator for speed sprayers, we first consider how to obtain information on operator behavior not only on the simulator, but also in the actual field. Firstly, the sensors must be able to track the driver's head movement and finger movement with Six Degrees of Freedom (6-DoF). Observing the driver's head movement is important because the driver needs to be aware of the trees to be sprayed while operating the nozzle and plates of the speed sprayer. By tracking the finger, we can confirm whether the control buttons for nozzles and plates have been pressed. Secondly, the driver must be able to operate the vehicle using the physical car controls. Finally, the driver must be able to feel the car controls in virtual and physical environments in the same way. The cockpit of the speed spray in the actual field has two camera sensors, one for the front and one for the rear, to detect the driver's behavior while driving and the steering wheel rotations.

A. Sensors

Sensors include wheel and pedal sensors, VR platform sensors, and cameras. All sensors used in this study are as follows.

A.1. Logitech G27 Force Feedback Wheel and Pedal

The Logitech G27 is a gaming racing wheel compatible with PlayStation 2 and 3. These devices were mounted on a cockpit frame manufactured by Rossomodello [15]. Figure 2 shows the steering wheel and pedal of Logitech's G27 attached to the frame and the speed sprayer in the actual field. The steering is equipped with a dual-motor force feedback system that enables the steering to receive inputs from the unevenness of the road surface while driving. The cockpit is equipped with a seat that can be adjusted and reclined. The Logitech G27 operates on Windows and Mac with the Logitech G27 driver.

(a) Exterior

(b) Cockpit

Figure 3. 3D model of the SSV1091FSC in our simulator.

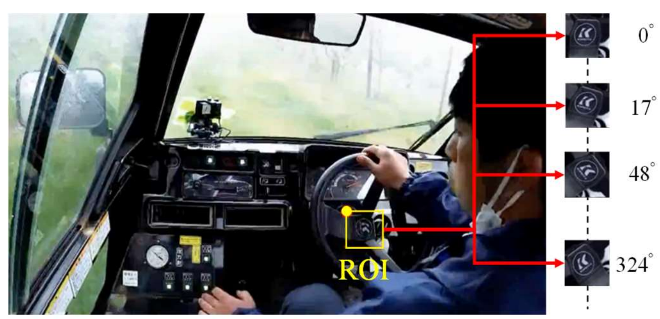


Figure 4. Mark of the steering wheel captured by the rear camera.

A.2. VR Platform: Oculus Quest 2

Oculus Quest 2 is a head-mounted display (HMD)-based VR device developed by Facebook, featuring a 6-DoF angular and linear tracking system that can measure head pose and hand gestures. This system uses Inertial Measurement Units (IMUs) that assess linear acceleration and rotational velocity with low latency and cameras in the HMD that creates a Three-Dimensional (3D) map of the room space and hand landmarks of the user. The initial position of the HMD is pre-calibrated against the position of the car steering wheel since

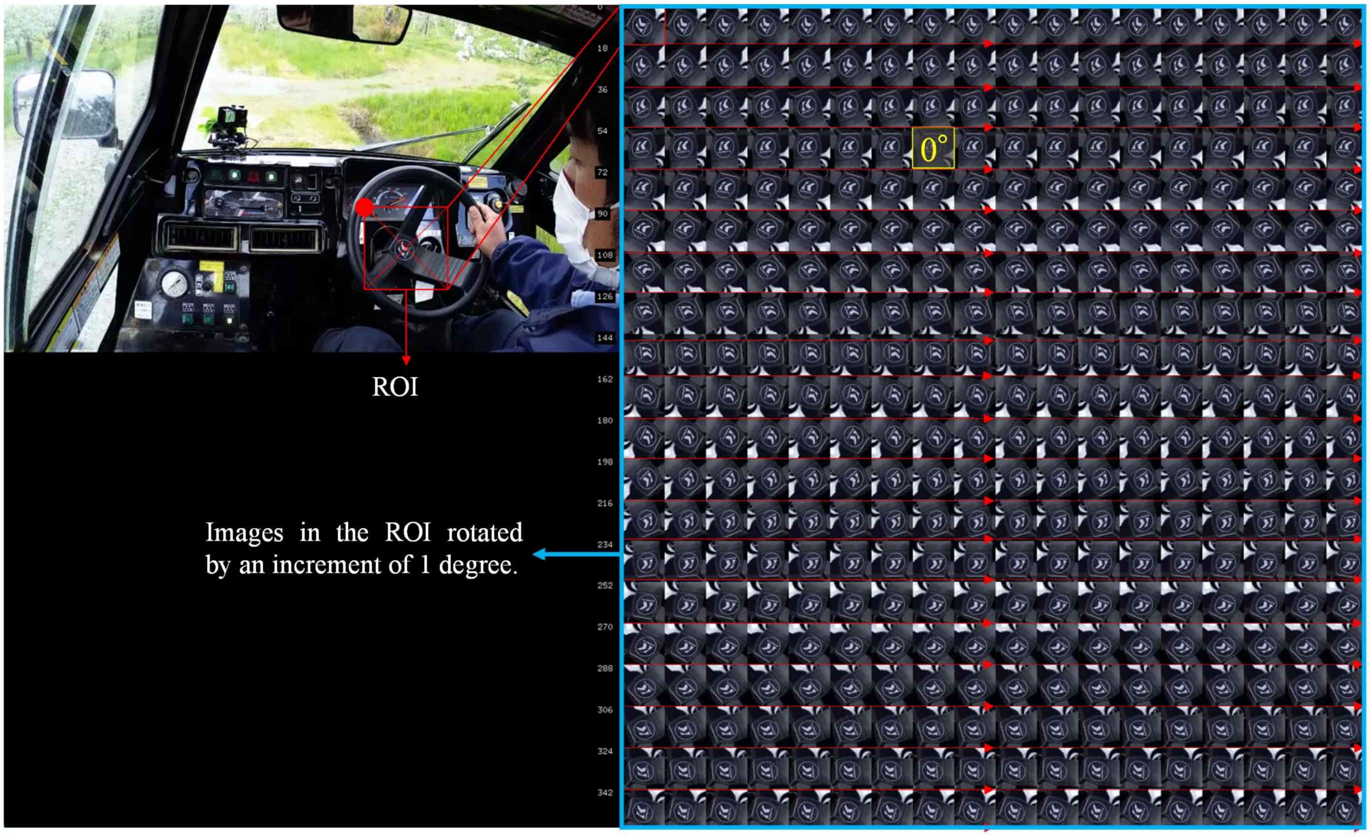


Figure 5. Our tool to semi-automatically specify the angle of the steering wheel marks.

no auxiliary devices, such as base stations, are used to obtain the absolute position of the driver.

A.3. GoPro Hero8 Cameras

Two cameras, as shown in Figure 2, are synchronized to capture videos of the cockpit of the speed sprayer in the actual field from two directions, front and rear. Videos are captured at 60fps with a resolution of 3,840 x 2,160px. The 60fps cameras are considered acceptable because there is no high speed movement in driving or button operation of the speed sprayer. The resulting video from the front camera is analyzed by image processing techniques to obtain the driver's head pose, whereas the video from the rear camera is to obtain the driver's finger movements and steering wheel rotation.

Table I shows data collected from each sensor. The Logitech G27 outputs car driving information such as steering wheel rotation, gas and brake pedals, and sends vibrations from the steering wheel. The amplitude of the vibration can be set in proportion to the speed of the vehicle. The Oculus Quest 2 tracks the driver's head pose and finger movements.

B. Simulator

Our simulator is built for the Unity [16] framework. We chose this framework because of its popularity in game and simulation developments. In addition, we can obtain the 3D data necessary to build the simulation scene from the Unity Asset Store, as well as the Software Development Kit (SDK) for the Logitech G27 used in this study, enabling us to work on the development in a short time.

Figure 3 shows the 3D model of the SSV1091FSC placed in a virtual scene. The scale has been adjusted to give the landscape seen from the model the same appearance as the realistic landscape. Additionally, to simplify the operation of the buttons for spraying the pesticide, these buttons were made larger. The main-button opens or closes all spray nozzles, and the left-, center-, and right-buttons are for spraying the left, center, and right sides of the vehicle's direction of travel, respectively. The operator receives feedback from the sound and the color change of the buttons on pressing them.

The initial position of the Oculus Quest 2 is pre-calibrated against the position of the steering wheel, and no auxiliary device such as a base station can be used to obtain the relative position between these devices. As the simulator is not in mixed-reality, the position of the simulated steering wheel on the VR need to match that of the Logitech G27.

The Oculus Quest 2 runs at a 72fps by default, but to accommodate the reduced frame rate caused by our experimental program, all data was recorded at a sampling rate of 60fps. An Intel Core i7-7700 CPU 3.60GHz, RAM 32GB, NVIDIA GeForce GTX1070 8GB was used as the computing system to control the simulator.

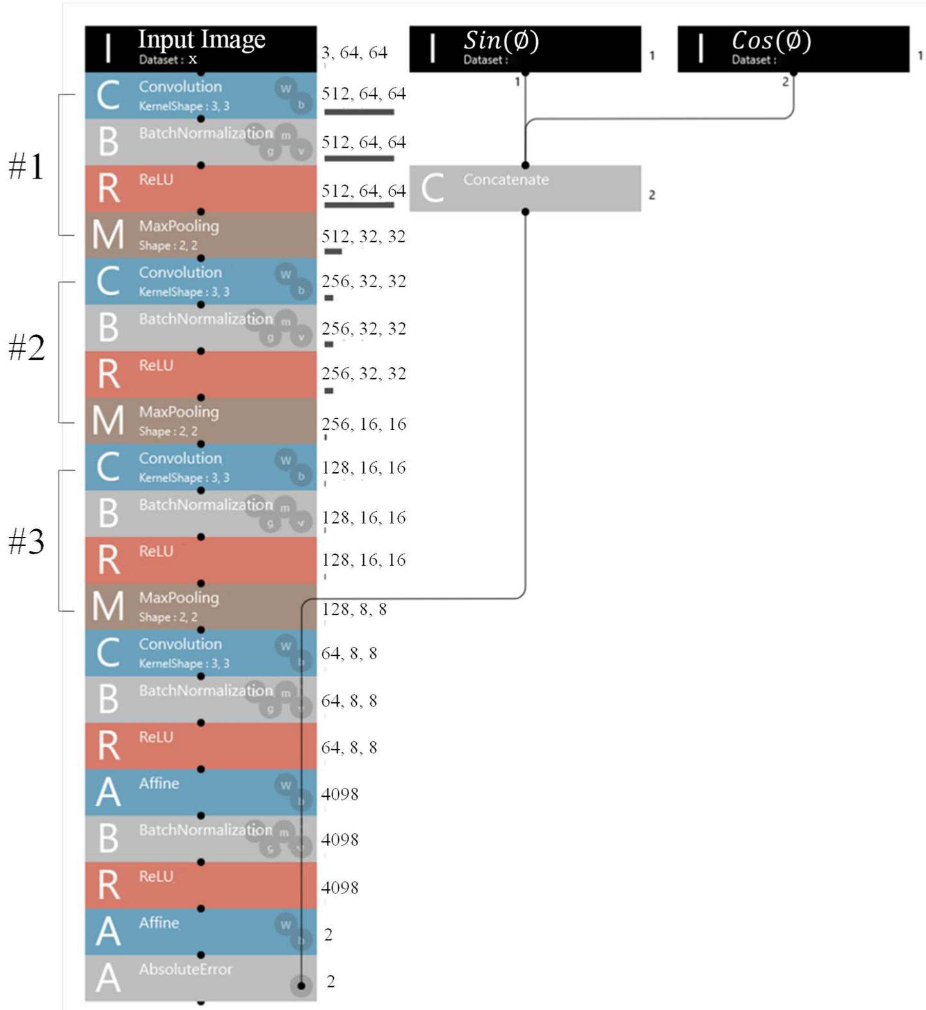


Figure 6. Diagram of the deep feedforward neural network used in this study.

C. Field Measurement

As with the simulator, the operator's behavior and the rotation of the steering wheel during driving a real speed spray are measured using image processing techniques. The methods used to quantify these behaviors are described below.

C.1. Head posture and hand movements

To estimate the head posture from a front-facing camera, facial landmarks need to be extracted from the detected face area. There have been many studies on extracting facial landmarks [17], but we adopted the MediaPipe framework for this study [18]. This framework provides customized machine learning solutions for face detection, facial landmark extraction, and hand tracking.

To extract the facial landmarks, we used the Face Mesh feature of the MediaPipe. From the 468 extracted landmarks, we selected 8 points: the corners of the eyes and mouth, as well as the tips of the nose and chin to determine the head posture. The Perspective-n-Point (PnP) solution was used to estimate the head posture of the 6DoF from these points [19].

The MediaPipe Hands tracks hands and fingers in real time [20]. This feature detects 21 hand landmarks. The tracked fingertip landmarks were used to detect pressed buttons and their frequency.

C.2. Steering wheel rotations

Image recognition was used to estimate the rotation information of the steering wheel from its image. Since most vehicles have a mark on the steering wheel, its rotation can be estimated by measuring the rotation of the mark. For this study, we simply built a Deep Feedforward Neural Network (DFNN) to estimate the rotation of the steering wheel. Figure 4 shows the mark of the steering wheel as it is captured by the rear camera. DFNN extracts the rectangular region of this mark as a Region of Interest (ROI) and estimates the rotation angle of the mark.

To construct a DFNN, pairs of rotation angle data for the image of this ROI are required. Generating these pairs of data one by one manually would be a laborious task. For this study, we developed a tool to generate this data conveniently. This tool works as follows. First, it creates 360 rotated ROIs by

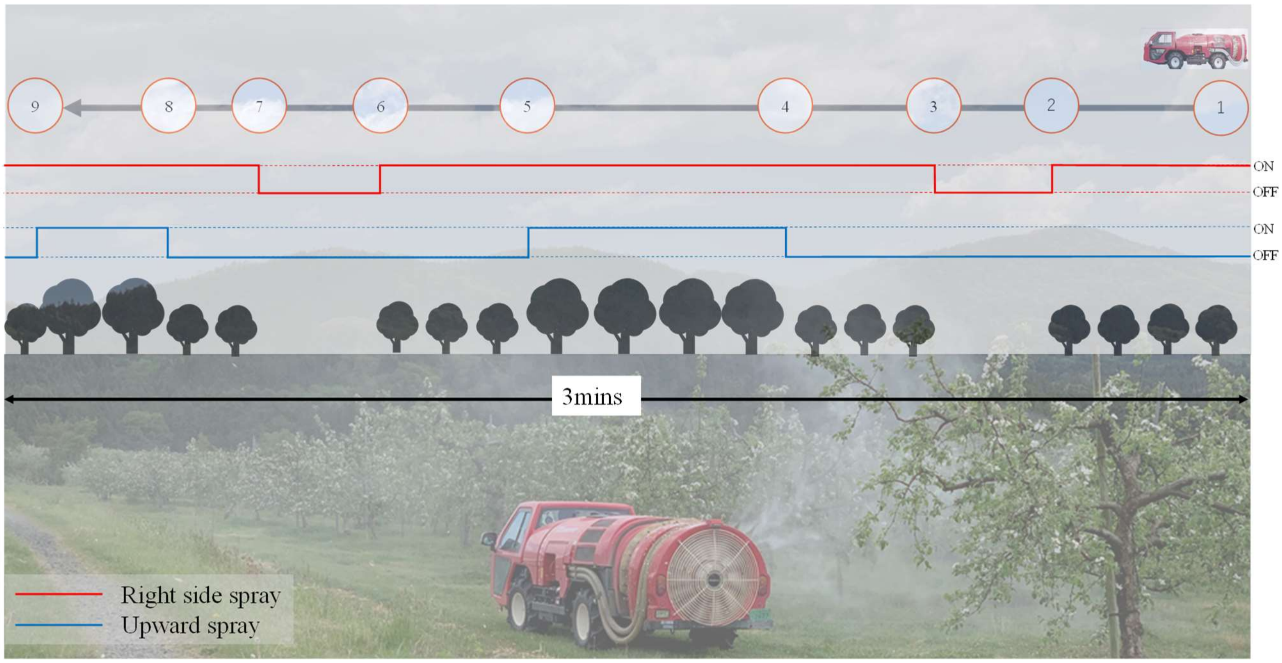


Figure 7. Operator behavior during speed spraying in the experiments of this study.

rotating the ROI image by 1 degree 360 times. By finding a mark with a rotation angle of 0 degree in these images, the rotation angles of the images before and after this mark becomes ±1 degrees. Similarly, the mark with a rotation angle of 1 degree is followed by a mark with a rotation angle of 2 degrees, and the mark with a rotation angle of -1 is preceded by a mark with a rotation angle of -2 degrees. Finally, the rotation angle can be specified for all 360 images in this manner. Figure 5 shows an example of a working interface of our tool. Taking a sufficiently large ROI enables the ROI image to be rotated in its complete shape.

Our DFNN was built using Neural Network Console [21], a deep learning framework developed by Sony Group Corporation designed from the perspective of engineers. Figure 6 shows the diagram of our neural network. The input image is the previously described ROI images in 64x64px. The output is the value of the rotation angle of this image encoded to sine and cosine. This encoding is important when dealing with data with circular topology. We then apply convolution, batch normalization, and Rectified Linear Unit (ReLU) to the input image with the output filter set to 512. The next layer is the maximum pooling layer, defined by a 2x2 pooling window. These procedures are repeated twice, but each time the output filter is reduced by half. Subsequently, the fully connected layer (Affine) is applied after convolution, batch normalization, and ReLU to obtain 4,098 outputs. Again, batch normalization, ReLU, and Affine are applied, and the absolute error between the inferred and true values are evaluated. The network was trained on 44 sets of ROI images corresponding to 1 to 360 degrees for a total of 15,840 images and validated with 18 sets for a total of 6,480 images. This network can infer the rotation angle of the steering wheel with an error of less than 2 degrees, which is comparable to the measurement accuracy of the steering wheel rotation angle of the simulator.

IV. Experiment and Result

This experiment compares the operator's behavior when operating the pesticide spraying in the orchard and when operating it on the simulator to verify whether unique operations are observable in both cases. A three-minute video of the spraying process was selected, and the nine actions taken during the process are summarized below.

- *Action 1*: The operator moves the vehicle to the beginning trees to be sprayed, then presses the main-button to open all spray nozzles, followed by the right-button to start spraying to the right side of the vehicle.
- *Action 2*: The operator presses the right-button again to stop the spraying when the vehicle passes the end point of the group of trees. An action is taken to confirm that the injection has been stopped via the right side mirror.
- *Action 3*: When the next group of trees is reached, the operator presses the right-button again to spray towards the right side of the vehicle. When the next group of trees is reached, the operator presses the right button again to spray towards the right side of the vehicle. At that time, the operator confirms whether the spraying works correctly through looking at the side mirror.
- *Action 4*: The operator presses the center-button to start spraying upward to the tall trees and confirms through the rear window glass.
- *Action 5*: After passing over the tall trees, the operator presses the center-button to stop the upward spraying and confirms through the rear window glass.

(a) Action 1

(b) Action 2

(c) Action 3

(d) Action 4

(e) Action 5

Figure 8. The relationship between the vehicle and the tree for actions 1 through 5.

- *Action 6~9*: The operator performs actions 2~5 to spray a group of trees having similar characteristics.

The actions that occurred during the three-minute spraying task used in this experiment and the direction of the spraying are shown in Figure 7. The vehicle traveled straight at 3.6 km/h and there were few bumps in the path.

To reproduce the above nine actions in the simulator, the operator is given voice guidance. In addition, the buttons in the simulator are virtual. Hence the operator is required to practice pressing the buttons properly while driving the vehicle.

Ten subjects aged between 22 to 26 participated in the spraying experiment on a simulator. These subjects were familiar with operating content in VR or Mixed-Reality (MR) environments. Before starting the experiment, each subject was asked to practice a set of operations twice in about 10

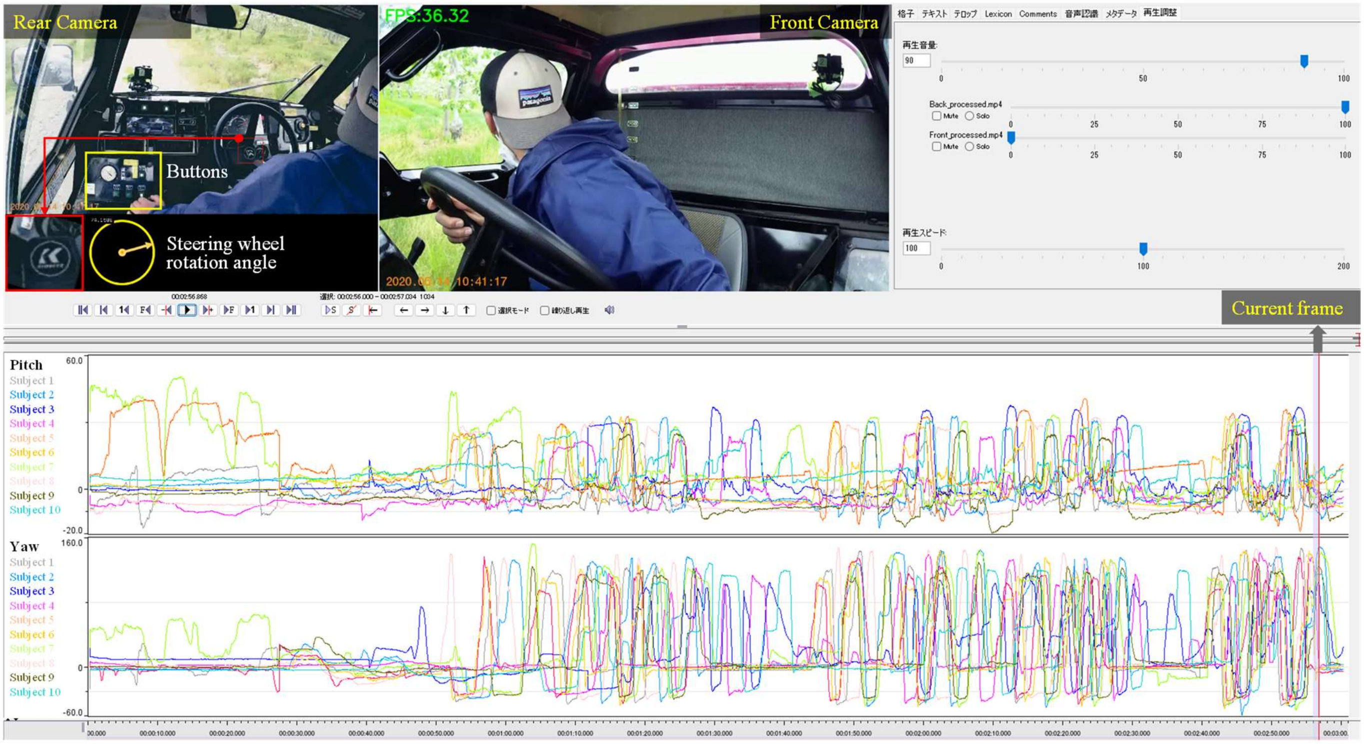


Figure 9. Three-minute of operator behavior data integrated into ELAN.

minutes. However, since this experiment was intended to check how the operator would behave when performing all the operations, we did not require the operator to perform all the operations up to the exact timings. Figure 8 shows images of the relationship between the vehicle and the tree for actions 1 through 5. The first action is to approach a parked speed sprayer to a tree and start spraying. The approach is left up to each subject. The images of actions 2 and 3 show that the trees are lined up about twice as far apart as the length of the vehicle. Finally, the images from actions 4 and 5 show the vehicles before and after passing through a tall tree.

Data as shown in Table I was recorded from the simulator. Since the simulator can be played back using this data, the behavior can be confirmed in detail in the post-experimental analysis. This data was integrated into the EUDICO Linguistic Annotator (ELAN) [22] as time-series data, and the attribute information of the actions were also recorded as annotation labels. Likewise, in the actual speed sprayer, the operator's head posture, hand landmarks, and steering wheel rotation angles were measured from the video taken from the front and rear cameras in the cockpit and integrated into the ELAN. Figure 9 shows the pitch and yaw angles of the heads of ten subjects measured from the simulator during the same three-minute simulation apart from the actual speed sprayer operation. In this way, the operator's behavior data and the videos can be synchronized and displayed.

A. Steering wheel rotation angle

The rotation angle of the steering wheel in action 1 is shown in Figure 10. Overall, the trend of the subject's steering wheel rotation angle is the same as that of the actual speed spray operator, which show large variations in their steering wheel rotation angles. Since the approach steps were not specified in this experiment, some subjects went straight and then turned, while others turned and then went straight. After action 2, there was no significant variation in the steering wheel angles because the vehicle's travel path was linear.

The speed sprayer used in this study is Four Wheel Steering (4WS), while the simulator vehicle used in this study is Two Wheel Steering (2WS), which resulted in differences in driving operation. We intend to analyze the effect of this difference as a simulator for training speed sprayer operation.

B. Characteristics of button operations

As described earlier, there are three operating buttons for spraying pesticides. We attempted to analyze the differences in gestures when operating the buttons by detecting both the operator's hand landmarks in the simulator and in the actual speed spray. However, we found that the operators of the simulator behaved in a fundamentally different way, as they operated the virtual buttons in the air while the actual speed sprayer operators placed their hands on the button platforms.

To observe the differences in button operations, we focus on the head movements. In both the simulator and the actual vehicle, the operator faces in the direction of the button to be operated, hence we analyzed their head movements. Figure 11 shows the operators' head movements in action 9. From this figure, we found that the trend of yaw and pitch angles of the head during button operation are similar even when in different environments.

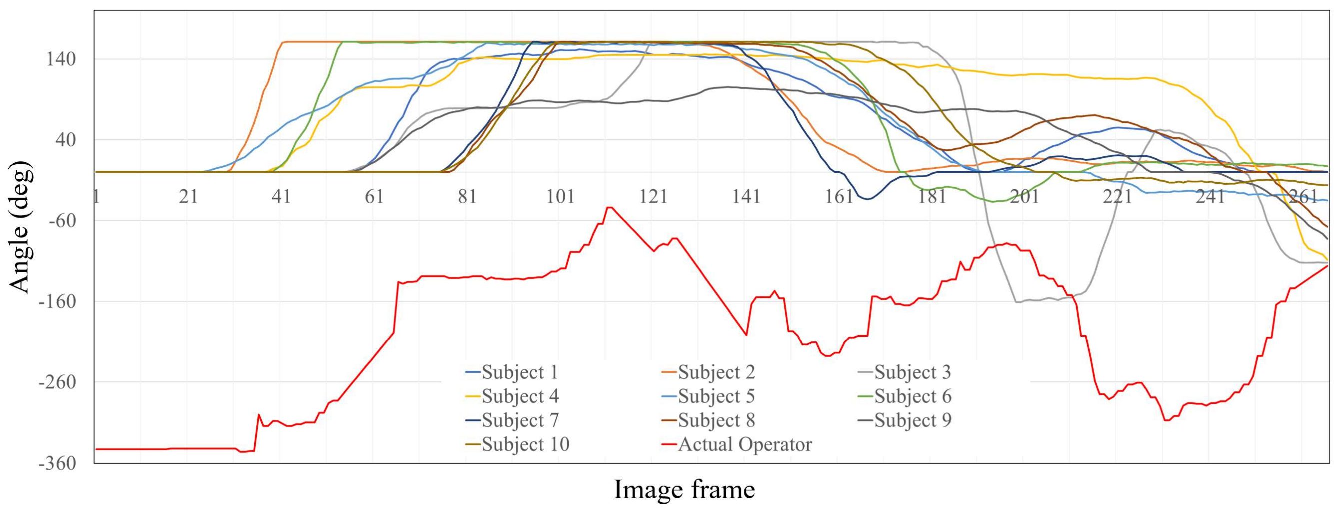
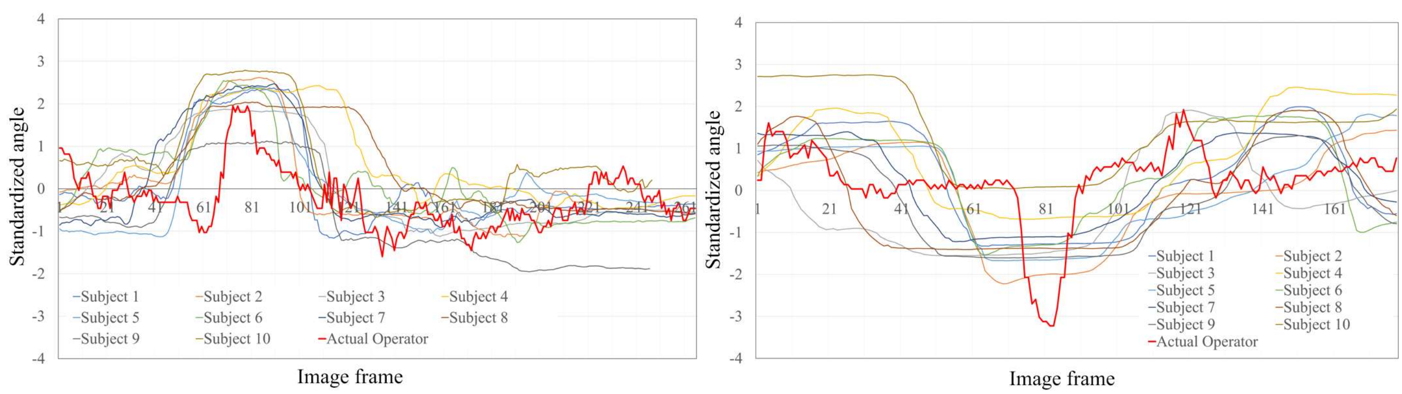


Figure 10. The rotation angle of the steering wheel in action 1.



(a) Pitch angle

(b) Yaw angle

Figure 11. Operators' head movements in action 9.

There was no precise instruction for the button operation timing in this experiment, resulting in a difference in the timing of the button operation among the subjects. For example, the button to stop spraying must be pressed when the vehicle passes the end point of a group of trees, but the subject's judgment of the relative position of the rear-mounted sprayer to the trees causes the button to be pressed at different timings. These situations appear as differences in the variation of the yaw angle of the head, as shown in Figure 12. A positive yaw angle indicates that the subject is checking the relative position of the trees and the sprayer through the window, while a negative yaw angle indicates that the subject is checking the position of the button. The solid and dotted lines indicate subjects who pressed the button early (Group I) and late (Group II), respectively.

V. Conclusion

In this study, we have developed a speed sprayer simulator and analyzed the differences in driving between the actual speed sprayer and the simulator vehicle. From the video footage of the pesticide spraying in the apple orchard, we were able to analyze the operator's speed spray operation characteristics and reproduce each of them on the simulator.

Button operations in the VR simulator were significantly different from those in the actual vehicle. However, we were able to see the characteristics of the button operations from the head movements because the face was always turned in the direction of the button before the button operation in our experiment.

In developing the simulator, we also measured the steering wheel rotation and automatically detected the operator's behavior by image processing. In the future, we plan to improve the simulator by adding a function to measure the amount and percentage of pesticides wasted by the simulator operator and by integrating the control buttons into the steering wheel to reduce head movements.

Acknowledgment

This work was done in collaboration with YAMABIKO Corporation and the Faculty of Software and Information Science, Iwate Prefectural University.

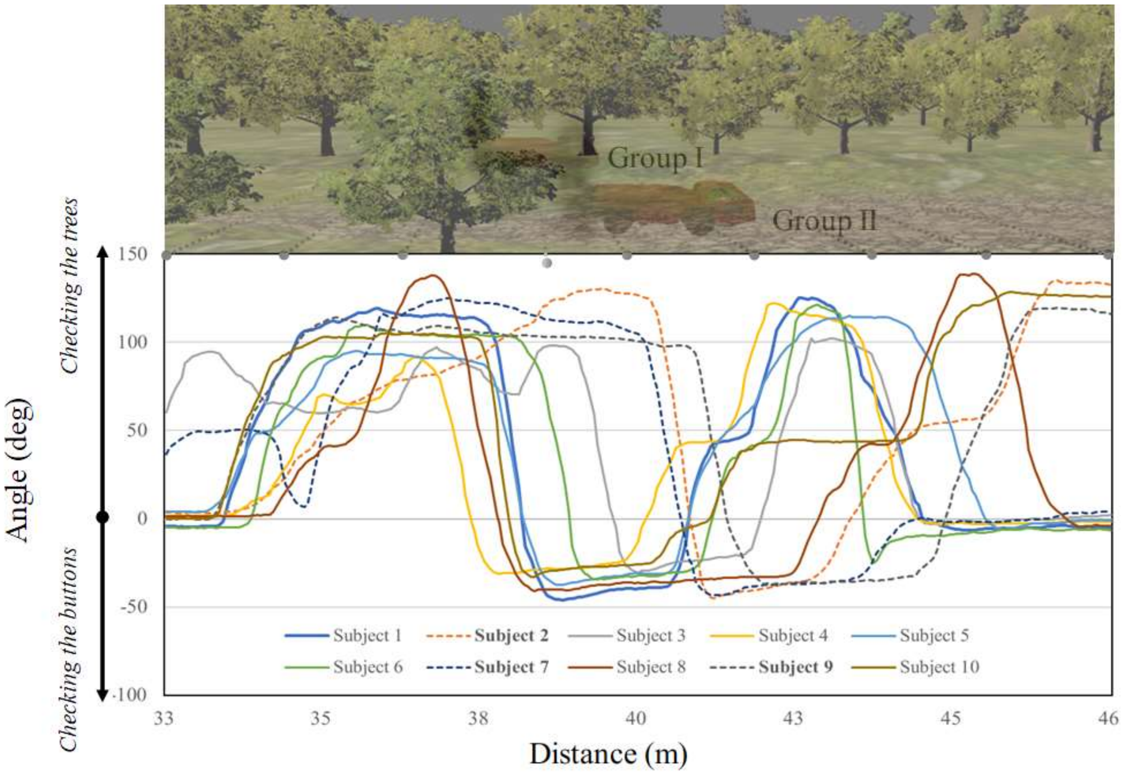


Figure 12. Variation in the yaw angle of the head, which resembles a difference in the timing of button operation. The solid and dotted lines indicate subjects who pressed the button early (Group I) and late (Group II), respectively.

REFERENCES

[1] Y. Tanaka, O. D. A. Prima, K. Ogura, K. Matsuda, and S. Yuki, "Toward the Development of a VR Simulator for Speed Sprayers," ACHI 2021, The Fourteenth International Conference on Advances in Computer-Human Interactions, pp. 1-5, 2021.

[2] The Japanese Ministry of Health, Labor and Welfare (MHLW), "Introduction of the Positive List System for Agricultural Chemical Residues in Foods," https://www.mhlw.go.jp/english/topics/foodsafety/positivelist060228/introduction.html [retrieved: September, 2021

[3] J. C. Van de Zande, H. Stallinga, J. M. G. P. Michielsen, and P. van Velde, "Effect of Sprayer Speed on Spray Drift," Annual Review of Agricultural Engineering, 4(1), pp. 129-142, 2005.

[4] M. Lešnik, D. Stajnko, and S. Vajs, "Interactions between Spray Drift and Sprayer Travel Speed in Two Different Apple Orchard Training Systems," International Journal of Environmental Science and Technology, 12(9), pp. 3017–3028, 2015.

[5] M. Gonzalez-de-Soto, L. Emmi, M. Perez-Ruiz, J. Aguera, and P. Gonzalez-de-Santos, "Autonomous Systems for Precise spraying–Evaluation of a robotized patch sprayer," Biosystems Engineering, 146, pp. 165–182, 2016.

[6] J. H. Han, C. H. Park, Y. J. Park, and J. H. Kwon, "Preliminary Results of the Development of a Single-Frequency GNSS RTK-Based Autonomous Driving System for a Speed Sprayer," Journal of Sensors, pp. 1-9, 2019.

[7] É. de C. C. Penido, M. M. Teixeira, H. C. Fernandes, P. B. Monteiro, and P. R. Cecon, "Development and Evaluation of A Remotely Controlled and Monitored Self-Propelled Sprayer in Tomato Crops," Revista Ciencia Agronomica, 50(1), pp. 8–17, 2019.

[8] D. O. Gonzalez et al., "Development and Assessment of a Tractor Driving Simulator with Immersive Virtual Reality for Training to Avoid Occupational Hazards," Computer and Electronics in Agriculture, 143, pp. 111–118, 2017.

[9] M. Watanabe and K. Sakai, "Development of a Nonlinear Tractor Model Using in Constructing a Tractor Driving Simulator," 2017 ASABE Annual International Meeting, pp. 1–6, 2017.

[10] A. Fujimoto, T. Satow, and T. Kishimoto, "Simulation of Spray Distribution with Boom Sprayer Considering Effect of Wind for Agricultural Cloud Computing Analysis," Engineering in Agriculture, Environment and Food, 9(4), pp. 305–310, 2016.

[11] P. E. Sumner, "Air Delivery Sprayer," The University of Georgia Colleges of Agricultural and Environmental Sciences & Family and Customer Science Bulletin, 979, pp. 1-8, 2009.

[12] Kioritz Speed Sprayer, https://www.yamabiko-corp.co.jp/kioritz/products/category/contents_type=59 [retrieved: September, 2021]

[13] D. O. Gonzalez et al., "Development and assessment of a tractor driving simulator with immersive virtual reality for training to avoid occupational hazards," Computers and Electronics in Agriculture, 143, pp. 111–118, 2017.

[14] H. Asaei, A. Jafari, and M. Loghavi, "Site-specific orchard sprayer equipped with machine vision for chemical usage management" Computers and Electronics in Agriculture, 162, pp. 431–439, 2019.

[15] GTD Simulator, http://www.rossomodello.com/gtd/gtd-top.html [retrieved: September, 2021]

[16] Unity Technologies, https://unity.com/ [retrieved: September, 2021]

[17] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1867-1874, 2014.

[18] Lugaresi et al., “Mediapipe: A framework for building perception pipelines,” arXiv preprint arXiv:1906.08172, 2019.

[19] F.Rocca, M. Mancas, and B. Gosselin, “Head Pose Estimation by Perspective-n-Point Solution Based on 2D Markerless Face Tracking,” Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, 136 LNICST, pp. 67–76. 2014.

[20] Zhang et al., “Mediapipe hands: On-device real-time hand tracking,” arXiv preprint arXiv:2006.10214, 2020.

[21] T. Narihira et al., “Neural Network Libraries: A Deep Learning Framework Designed from Engineers' Perspectives,” arXiv preprint arXiv:2102.06725, 2021.

[22] ELAN (Version 6.0), “The Language Archive,” https://archive.mpi.nl/tla/elan [retrieved: September, 2021]

A Connectivity Improvement Method for Behavior Driven Acceptance Tests

Tugkan Tuglular
Department of Computer Engineering
Izmir Institute of Technology
Izmir, Turkey
mail: tugkantuglular@iyte.edu.tr

Nazım Umut Ekici
Department of Computer Engineering
Izmir Institute of Technology
Izmir, Turkey
mail: nazimekici@iyte.edu.tr

Abstract—**Behavior Driven Acceptance Tests (BDATs) are not necessarily written in a semantic flow. Sometimes they are written in an ad-hoc manner, and some other times they are grouped by features or requirements. Connecting BDATs for faster test execution may prevent reset or set operations in test environments. Moreover, if BDATs cannot be connected, that may mean missing BDATs. Therefore, better-connected BDATs result in better implementation and testing. This work proposes a method for improving the connectivity of BDATs utilizing natural language processing techniques and a graph model-based test generation technique called Event Sequence Graphs (ESGs). For the connection of BDATs, we utilize the technique called elimination of tags by combination in ESGs introduced in our previous work. The proposed method here improves the connectivity of existing behavior-driven acceptance test suites. It is validated through two non-trivial examples. The results demonstrate the feasibility of the proposed method.**

Keywords-model-based testing; event sequence graphs; behavior driven acceptance tests; Gherkin.

I. Introduction

The proposed method in this paper improves the connectivity of behavior-driven acceptance test suites developed using the method presented in [1]. Behavior Driven Development (BDD) is focused on defining fine-grained specifications of the behavior of the targeted system [2]. In BDD, tests are clearly written using a specific ubiquitous language, such as Gherkin [3]. For developing Behavior Driven Acceptance Tests (BDATs), there are environments like Cucumber [3], which forces testers to use a test template using Gherkin language and environments like Gauge [4], which does not impose any language. The scope of this study is BDATs developed in Gherkin.

Although Gherkin and its scenario template helps test designers in writing test scenarios, there is no guidance on the connectivity of test scenarios. All public Github repositories are searched for files with the extension ".feature". Github does not report the number of unique repositories that match a given query. Instead, it reports that there are over 2M unique files that match the query at the time of reporting. Github also limits the query results to the top 1000 files, most of which are hosted in the same repositories (i.e., 1000 file results are not from 1000 unique repositories). By executing the same query at different times, we collected 1314 unique repositories with Gherkin scenarios. The largest repository has 1041 feature files. Our search results can be found at https://github.com/esg4aspl/Gherkin-Scenario-Collection-and-Analysis/blob/main/README.md. We analyzed 5% of these 1314 repositories manually and did not find any work (i.e., explanation, code) related to connectivity of Gherkin scenarios. This work addresses this problem and provides a method for improvement of connectivity of BDATs.

The method proposed here utilizes natural language processing (NLP) techniques and a graph model-based test generation technique. Therefore, we borrow the connectivity definition from the theory of directed graphs. As a solution, we utilize semantic similarity measure to tag BDATs, then transform tagged BDATs into formal graph test models, and finally connect them through elimination by composition method introduced in [1]. So, if there are unconnected BDATs, or Gherkin test scenarios, the proposed method warns test designer to improve existing BDATs by adding new BDATs.

The proposed method assumes that clauses written in Gherkin can be represented by events. In that case, an event-based formal model would fit better to BDATs. Therefore, we propose the use of Event Sequence Graphs (ESGs) for modeling BDATs. To model a BDAT as an ESG, ESGs are extended with tags. This is one of the novelties presented in this paper. Another novelty presented here is the process of finding missing BDATs. To find missing BDATs, the proposed method follows elimination of tags by combination. After the missing BDATs are completed, an ESG without any tags is obtained. The proposed method is explained with a running example in Section III. For evaluation, a BDAT test suite is selected from Github™ and the proposed method is applied to this test suite. The results are shared in Section IV.

This paper makes the following main contributions:

(i) Method: The proposed method creates a corpus from exiting Gherkin statements and tries to match end of a test scenario with the beginning of another test scenario through semantic similarity. A unique tag is automatically generated for the matched statements. For the unmatched statements unique tags are also generated with a table entry of close statements. Then tagged BDATs are transformed to tagged ESGs and combined by utilizing the elimination by composition method introduced in [1]. Analysis of the resulting ESG or ESGs reveals improvement in the connectivity of BDATs.

(ii) Tool: We developed a tool that implements the method explained in (i) and shared in a public repository.

The manuscript is organized as follows: In the next section, the fundamentals of the concepts used in this research are given along with examples and figures. The proposed method is explained in Section III using a running example and Section VI presents the software tools developed and used in this research. Section V gives an evaluation of the proposed method along with a discussion in Section VI. Section VII sets out the threats to validity. Section VIII outlines related work, and the last section concludes the paper.

II. Fundamentals

A. Gherkin

Gherkin uses a set of special keywords to give structure and meaning to executable specifications [3]. It provides the behavior definitions of the intended software not only to product owners and business analysts, but also to developers and testers [5]. Gherkin is a line-oriented language in terms of structure and each line must be divided by the Gherkin keyword except feature and scenario descriptions [3]. In this paper, some of the Gherkin keywords; namely *Feature*, *Scenario*, *Given*, *When*, *And*, *Then*, are utilized. Throughout the paper, the terms Gherkin scenario, scenario, and BDAT are used interchangeably.

Tests should be independent of each other so that they can be run in any order or even in parallel. This principle is also applied in developing BDATs. So, each BDAT should be run manually or automatically independent of other BDATs. However, they should also be composable so that it will be possible to execute a BDAT after a related one.

B. Natural Language Processing

Cosine similarity is a method for measuring similarity between two vectors [6]. By converting text documents to vectors, cosine similarity is widely used to assess the similarity between documents. Term Frequency/Inverse Document Frequency (TF-IDF) is used to convert text documents to vectors [6]. Given a corpus and a document from the corpus, for each word in the document, TF-IDF uses the frequency of the word in the document (its significance for the document) and the corpus (its informativeness for the corpus). By normalizing TF with DF, TF-IDF outputs a relative significance of each word in the document with respect to the corpus. TF-IDF and cosine similarity are commonly used together to assess the similarity of arbitrary documents in the context of the corpus [7], [8].

Text pre-processing is an essential part of NLP, which aims to improve any further processing [9], [10]. Two common types of pre-processing are stop word removal and stemming. Stop words are frequent words in the language, which have little informativeness (e.g., the, a, an for the English language) but can affect the output. Their removal also helps reduce the size of the corpus. Stemming is applied to words to strip them from any modifiers and transform them to their root form. For instance, withdrawal, withdrawing, and withdraws can all be stemmed from the word withdraw. Stemming aids in identifying semantic similarities out of the syntactic context.

C. Event Sequence Graphs

A model of the system, which requires the understanding of its abstraction, helps in testing its behavior. A formal specification approach that distinguishes between legal and illegal situations is necessary for acceptance testing. These requirements are satisfied by event sequence graphs [11].

Differing from the notion of finite-state automata, inputs and states are merged in ESG, hence they are turned into "events" to facilitate the understanding and checking the external behavior of the system. Thus, vertices of the ESG represent events as externally observable phenomena, e.g., a user action or a system response. Directed edges connecting two events define allowed sequences among these events [11]. Definitions from 1 to 3 and related examples and explanations along with Figure 1 are taken exactly as presented in [12]-[15].

Definition 1. An *event sequence graph* $ESG = (V, E, \Xi, \Gamma)$ is a directed graph where $V \neq \emptyset$ is a finite set of vertices (nodes), $E \subseteq V \times V$ is a finite set of arcs (edges), $\Xi, \Gamma \subseteq V$ are finite sets of distinguished vertices with $\xi \in \Xi$, and $\gamma \in \Gamma$, called entry nodes and exit nodes, respectively, wherein $\forall v \in V$ there is at least one sequence of vertices $\langle \xi, v_0, \ldots, v_k \rangle$ from each $\xi \in \Xi$ to $v_k = v$ and one sequence of vertices $\langle v_0, \ldots, v_k, \gamma \rangle$ from $v_0 = v$ to each $\gamma \in \Gamma$ with $(v_i, v_{i+1}) \in E$, for $i = 0, \ldots, k\text{-}1$ and $v \neq \xi, \gamma$.

To mark the entry and exit of an ESG, all $\xi \in \Xi$ are preceded by a pseudo vertex '[' $\notin V$ and all $\gamma \in \Gamma$ are followed by another pseudo vertex ']' $\notin V$. The semantics of an ESG are as follows. Any $v \in V$ represents an event. For two events $v, v' \in V$, the event v' must be enabled after the execution of v iff $(v, v') \in E$. The operations on identifiable components of the GUI are controlled and/or perceived by input/output devices, i.e., elements of windows, buttons, lists, checkboxes, etc. Thus, an event can be a user input or a system response; both are elements of V and lead interactively to a succession of user inputs and expected desirable system outputs.

Example 1. For the ESG given in Figure 1: $V=\{a,b,c\}$, $\Xi=\{a\}$, $\Gamma=\{b\}$, and $E = \{(a,b), (a,c),(b,c),(c,b))\}$. Note that arcs from pseudo vertex [and to pseudo vertex] are not included in E.

Furthermore, α(initial) and ω(end) are functions to determine the initial vertex and end vertex of an ES, e.g., for ES= $(v_0, \ldots, v_k)$ initial vertex and end vertex are $\alpha(ES)=v_0$, $\omega(ES)=v_k$, respectively. For a vertex $v \in V$, $N^+(v)$ denotes the set of all *successors* of v, and $N^-(v)$ denotes the set of all *predecessors* of v. Note that $N^-(v)$ is empty for an entry $\xi \in \Xi$ and $N^+(v)$ is empty for an exit $\gamma \in \Gamma$.

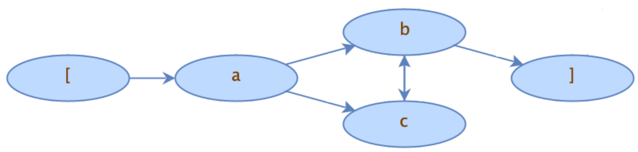


Figure 1. An ESG with a as entry and b as exit and pseudo vertices '[' and ']'.

Definition 2. Let V, E be defined as in Definition 1. Then, any sequence of vertices $\langle v_0, \ldots, v_k \rangle$ is called an *event sequence* (*ES*) iff $(v_i, v_{i+1}) \in E$, for $i=0, \ldots, k-1$.

The function l(*length*) of an ES determines the number of its vertices. If l(ES)=1 then ES=(v_i) is an ES of length 1. Note that the pseudo vertices [and] are not considered in generating any ESs. Neither are they included in ESs nor considered to determine the initial vertex, end vertex, and length of the ESs. An ES = $\langle v_i, v_k \rangle$ of length 2 is called an *event pair* (EP).

Definition 3. An *ES* is a *complete ES* (or, it is called a *complete event sequence*, *CES*), if $\alpha(\text{ES})=\xi\in\Xi$ is an entry and $\omega(\text{ES})=\gamma\in\Gamma$ is an exit.

A CES may not invoke interim system responses during user-system interaction. If it does not, that means that it consists of consecutive user inputs and only a final system response. CESs represent walks from the entry of the ESG to its exit, realized by the form (initial) user inputs → (interim) system responses → ⋯ (interim) user inputs → (interim) system responses → ⋯ → (final) system response.

ESGs are hierarchical models enabling sub-ESGs, or sub-models, which are also ESGs. A hierarchical ESG can be refined or flattened to one layer. Please see [12]-[15] for further details. Therefore, we can say that ESGs support and manage large models by following the divide-and-conquer approach in computer science.

D. Connectivity in Directed Graphs

Two vertices u and v in a graph G are connected if $u = v$, or $u \neq v$ and a u-v path exists in G [16]. A graph G is connected if every two vertices of G are connected; otherwise, G is disconnected [16]. The ESG obtained after graph transformation of BDATs and their composition might be a disconnected directed graph. By improving the connectivity of this graph, we would like to make it a connected ESG, so that test sequences can be automatically generated and reset operations are minimized.

III. Proposed Method

The proposed method improves connectivity of a BDAT test suite by NLP analysis, graph-based modeling, and model composition. The proposed method not only improves connectivity but also enables coverage-based test sequence generation by ESGs.

With the assumption that Gherkin clauses can be represented by events, the proposed method suggests the use of ESGs for modeling BDATs. To model a BDAT as an ESG, ESGs are extended with tags. This is explained first in this section. Then, how BDATs are combined using tagged ESGs is presented. After that, elimination of tags by combination process that is used to find missing BDATs is outlined. This section concludes with an example where all BDATs, i.e., original, missing, and additional BDATs, are composed into one ESG without any tags.

A. Extension of BDATs with tags

Best practice for Gherkin scenarios is to describe behavior rather than functionality. A behavior driven acceptance test is a specification of the behavior of the system, which verifies the interactions of the objects rather than their states [17]. A scenario that makes up a BDAT is composed of several steps. A step is an abstraction that represents one of the elements in a scenario which are: contexts, events, and actions [2]. So, a Gherkin scenario template is as follows:

Given context
When event
Then action

Contexts, events, and actions can be represented by events. A context, or state, is formed after a sequence of events. For instance, the line Given I am on the homepage in a scenario indicates that the context is being on the homepage and the user can reach the homepage by a sequence of events. So, we can say that a context is the result of a sequence of events. Sometimes, the sequence of events may be empty. An action is an event or results in an event depending on your standpoint. For instance, the line Then product list is displayed in a scenario is the action of the software, but for the user it is an event. We conclude that all Gherkin clauses are either events or a result of an or a series of events and therefore, we claim that all Gherkin clauses can be expressed as events.

Algorithm 1 defines the steps for extracting semantic relations between step definitions by utilizing NLP techniques. Given a set of Gherkin scenarios, a corpus is constructed by aggregating all step definitions from all scenarios. Punctuation and stop words are removed from the corpus to highlight more important words. Furthermore, words are stemmed to reduce them to their root form. After the conversions, pairwise cosine similarity is calculated for all items in the corpus by using the TF-IDF transformation [6]. For every Given and Then step definition, their similarity scores with every other Given and Then step definition are collected to a list. The list is sorted in the descending order of similarity score. The resulting collection lists best matching step definitions for each step definition. Output of the algorithm can be interpreted as a list of match suggestions for each step definition.

Algorithm 1 is applied to the running example. Number of correct matches present in each list length is also presented in Table I up to a list length of 5. Table I shows that, out of 17 possible matches, 12 of them are correctly identified as the best match by Algorithm 1. All possible matches are identified within the first 5 suggestions by the algorithm.

Algorithm 1. Match Gherkin Then Statement and Given Statement Pairs

```
Input: Scenarios
Corpus ← { }
For each scenario in Scenarios:
    Corpus = Corpus U scenario.stepDefinitions
Endfor
Corpus.removePunctuation()
Corpus.removeStopWords()
SimilarityScores = Corpus.calculatePairwiseSemanticSimilarity()
sortedMatchesForStepDefinitions = { }
For each stepDefinition in Corpus:
    Scores = SimilarityScores.getScoresForStepDefinition(stepDefinition)
    Scores.sortDescending()
    sortedMatchesForStepDefinitions[stepDefinition] = Scores
Endfor
Output: sortedMatchesForStepDefinitions
```

TABLE I. CORRECT TAG MATCH COUNTS IN A GIVEN LIST LENGTH FOR THE RUNNING EXAMPLE

Tagged step definition count	*List length 1*	*List length 2*	*List length 3*	*List length 4*	*List length 5*
17	12	14	15	15	17

As an example of Algorithm 1's results, consider the two Given step definitions from the running example below. Step definitions are best match for each other according to Algorithm 1. Both Given step definitions describe the same state in the program execution. Therefore, the match is considered to be correct.

Scenario: srch01- Do a valid search with a single keyword
Given I am on the homepage to do a single keyword search
Scenario: srch02- Do a valid search with multiple keyword
Given I am on the homepage to do a search with multiple keywords

Algorithm's output is plotted in Figure 2, where the y-axis shows the cumulative number of correctly matched step definitions when the length of the possible matches list is limited with a given number (i.e., only x of the most semantically relevant step definitions is considered).

B. *Representation of BDATs with tagged ESGs*

This work utilizes event sequence graphs for modeling BDATs. To model a BDAT as an ESG, ESGs are extended with tags [1].

Definition 4. A tagged ESG is an ESG, where a node or vertex may contain a tag instead of an event.

A tagged ESG is useful in transforming Gherkin scenarios or BDATs to ESGs. Contexts and actions are represented by tags and this way, tags become connection or composition points for ESGs.

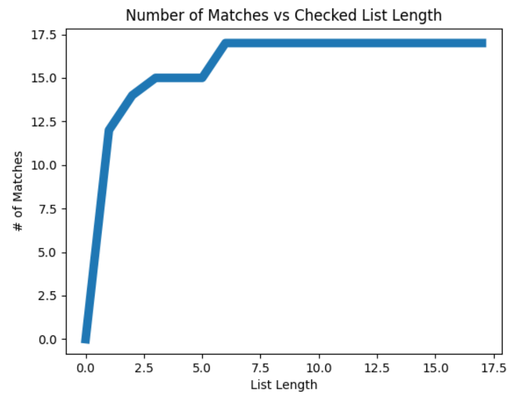


Figure 2. Number of correct matches for a given list length.

For instance, in the following Scenario cart02, *Given* event is tagged with #productPage and *Then* event is tagged with #shoppingBasket. Its ESG representation is shown in Figure 3.

Scenario: cart02 - Adding a product to cart
Given I am on a product detail page #productPage
When I select the amount
And I click the add to cart button
Then the product is added to my shopping cart #shoppingCart

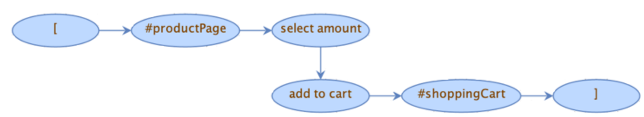


Figure 3. Tagged ESG for Scenario cart02.

Annotating Gherkin clauses with tags and representing BDATs with tagged ESGs enable us to combine BDATs.

C. *Combining two BDATs on tagged ESG*

To combine two BDATs, the following method is proposed. Ending Gherkin clause can be combined with starting Gherkin clause if they have the same tag. This means two Gherkin scenarios can be run in a sequence. We can connect Scenario cart02 with Scenario check01 presented below, where *Given* event is tagged with #shoppingBasket and *Then* event is tagged with #orderConfirmed. ESG representation of Scenario check01 is shown in Figure 4.

Scenario: check01 - Successful checkout
Given I have added an item to my shopping bag #shoppingCart
When I proceed to the check out
And I enter valid delivery details
And I select a payment method
And I confirm the order
Then I am redirected to the thank you page #orderConfirmed

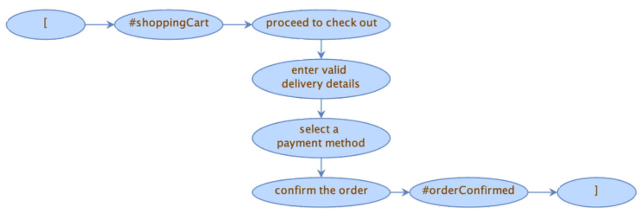


Figure 4. Tagged ESG for Scenario check01.

As seen, tags are used as connection points. Following the method presented in Section III-A, we can combine these two BDATs on a tagged ESG, since both are represented as a tagged ESG. The resulting tagged ESG is shown in Figure 5.

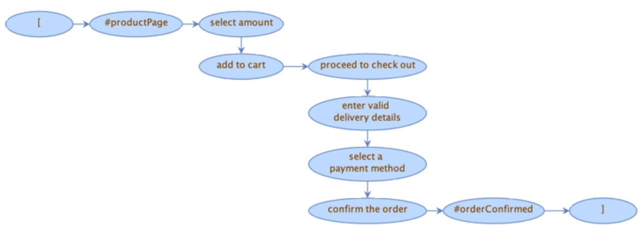


Figure 5. Tagged ESG for combined Scenarios cart02 and check01.

Algorithm 2 outlines the steps explained above.

Algorithm 2. Create Tagged ESG Segments

```
Input: Scenarios
Segments ← { }
For all scenario in scenarios
    entryNode ← Ø
    entryNode.label ← scenario.entryTag
    entryNode.isTag ← true
    prevNode ← entryNode
    For all stepDefinition is scenario.stepDefinitions
        stepNode ← Ø
        stepNode.label ← stepDefinition.text
        stepNode.isTag ← false
        prevNode.next ← stepNode
        prevNode ← stepNode
    exitNode ← Ø
    exitNode.label ← scenario.exitTag
    exitNode.isTag ← true
    prevNode.next ← exitNode
    Segments ← Segments U {entryNode}
Endfor
Output: Segments
```

D. *Finding missing BDATs*

To find missing BDATs, elimination by combination is proposed [1]. As seen above, once two BDATs are combined using a tag, that tag is eliminated. Therefore, first all possible tagged scenarios or their graphical representations, i.e., tagged ESGs, are merged. Algorithm 3 outlines the process of this operation.

Algorithm 3. Merge Tagged ESG Segments

```
Input: Segments
discoveredTags ← { }
For segment in segments
    For node in segment
        if(node.isTag)
            if(node.label in discoveredTags)
                // replace tag node with matched tag node
                // by adding node's descendents to matched node
                discoveredTags[node.label].takeoverNeighbors(node)
            else
                discoveredTags ← discoveredTags U {node}
            Endif
        Endif
    Endfor
Endfor
// remove orphan tags after matching
For segment in segments
    if(segment.length = 1)
        Segments ← Segments / {segment}
    Endif
Endfor
Output: Segments
```

It should be noted that a merged tagged ESG may be merged with another simple or merged tagged ESG. The goal is to reach an ESG without any tags, as shown in Figure 6. After all possible combinations are completed, if a tag remained on a tagged ESG indicates that there is a missing BDAT. If there are more than one tag, that may mean more missing BDATs. The process of tag removal is given in Algorithm 4.

Algorithm 4. Remove Tags from ESG Segments

```
Input: segments
discoveredTags ← { }
For segment in segments
    For node in segment
        if(node.isTag and node.hasAncestor and node.hasDescendant)
            For neighbor in node.neighbors
                node.label ← node.label U neighbor.label
                node.takeoverNeighbors(neighbor)
            Endfor
        Endif
    Endfor
Endfor
Output: segments
```

For instance, in the following Scenario acc03, *Given* event is tagged with #atHome and *Then* event is tagged with #orderDetail.

Scenario: acc03 - Check orders
Given I am logged in on the site #atHome
When I navigate to my orders
Then I see a list of my orders
And I can open an order to see the order details #orderDetail

This BDAT is the only Gherkin scenario that has the tag #orderDetail. Since there is no match, it indicates that a BDAT that starts with #orderDetail tag is missing. We can complete this missing BDAT as follows:

Scenario: acc10 - Back to order list page
Given #orderDetail
When I press OK button
Then order list page is displayed #orderList

As seen in the running example, elimination by combination gives us clues about the connectivity of BDATs. The method proposed here is to check whether all tags are combined. Any tag that is not eliminated suggests a missing BDAT.

E. Composition of BDATs on tagged ESG

After completing the missing BDATs and improving existing BDATs, the BDATs are composed on an ESG. The resulting ESG is shown in Figure 6. Elimination by combination enables us to find five missing BDATs, which are drawn in red on the resulting ESG in Figure 6.

IV. Tool Support

Algorithms 1 to 4 are implemented using Python and are provided at https://github.com/esg4aspl/Connectivity-Improvement-for-Behavior-Driven-Acceptance-Tests along with test data. For NLP operations, Natural Language Processing Toolkit (NLTK) [18] is used. Algorithm 1 is implemented in scenario_matcher.py, where the script takes a list of directories containing Gherkin scenarios and for each directory outputs the per step definition list of step definitions sorted according to semantic similarity. In addition to the output, match rate vs list length is plotted. Algorithms 2 to 4 are implemented in scenario_to_esg.py, where the script takes a directory containing tagged Gherkin scenarios and applies Algorithms 2 to 4 in order, converting scenarios to ESG segments and merging those segments by connecting them at the matching tags.

Once an ESG is ready then CES for edge and for edge-pair coverage can be generated for BDATs. The details of CES generation can be found in [14]. We utilized the TSD tool [19] to generate CES for both coverage criteria. The results are given in Section V-A.

V. Evaluation

For evaluation, the proposed method is applied to an existing test suite for an e-commerce software [20], which is also used as a running example in Section III, and the results are explained in Section V-A. For further evaluation, we asked five teams of graduate students to write BDATs for the same bank ATM software [21] after learning Gherkin and BDATs in a software testing graduate course. Their results are given Section V-B.

A. Evaluation of an E-commerce Software Test Suite

For the existing test suite for an e-commerce software [12], six features out of eight are taken for evaluation. The features locale and newsletter are left. The existing test suite has 15 scenarios, or BDATs, with 64 Gherkin clauses. Clause per scenario ratio is 4.26.

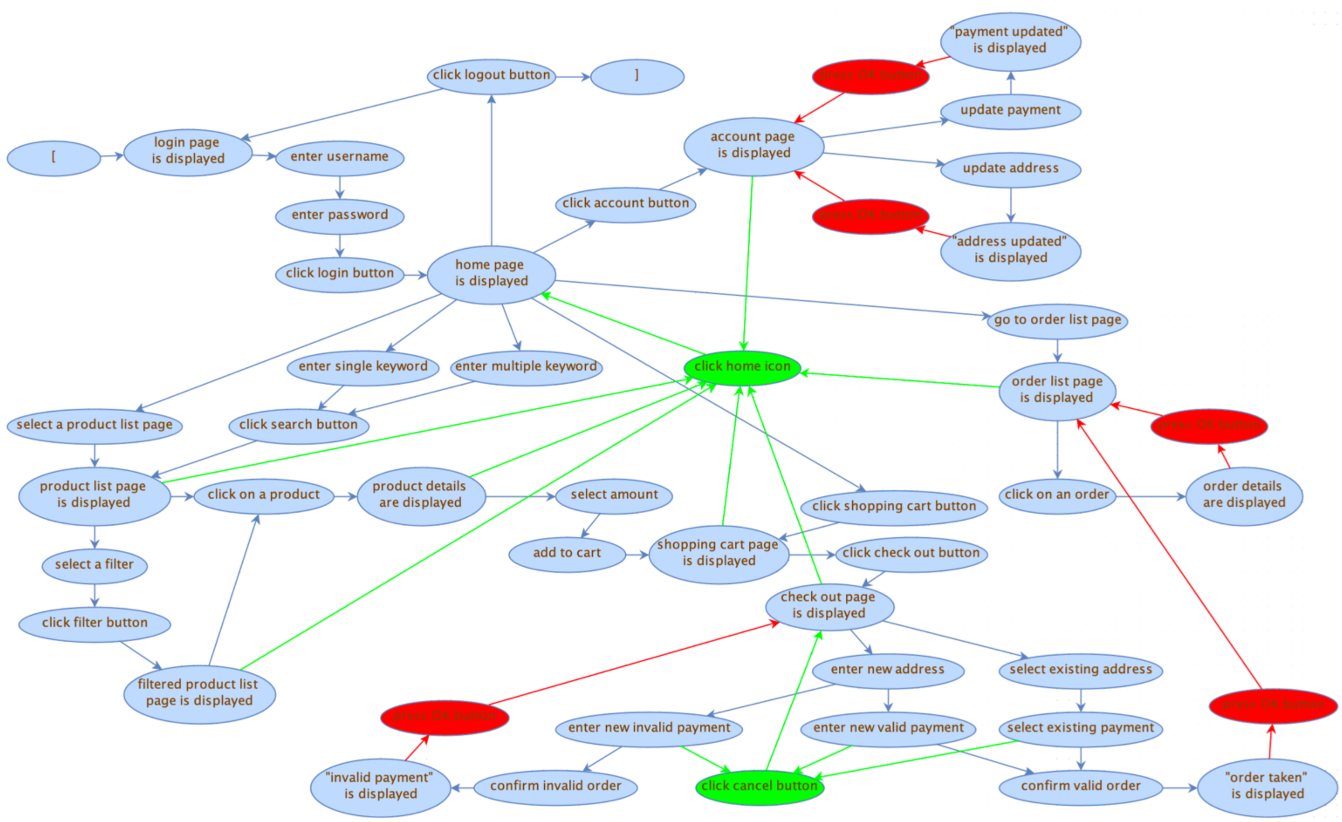


Figure 6. Composed ESG.

After applying the proposed method, we end up with 24 BDATs and 85 Gherkin clauses. There are 9 new scenarios but only 5 of them are missing scenarios. The other 4 scenarios are introduced to simplify and standardize some original scenarios. So, clause per scenario ratio is decreased to 3.54 from 4.26. The comparison of before and after the proposed method is given in Table II. The resulting test suite has the scenarios that are simplified, standardized, and tagged. Moreover, they become composable.

A further analysis of the resulting ESG shows that event sequences are stuck in the child pages of home page. There is no return to home page from child pages, which means that features of the software cannot be tested in sequence. In addition, it is discovered that there is no scenario about cancellation of the check-out process. Those BDATs are added in green to the resulting ESG in Figure 6. It should be noted that the graphical representation of BDATs enables us to perform such an analysis. Without tool support, it is very hard for test designers to conduct such analysis on text represented BDATs.

TABLE II. COMPARISON OF BEFORE AND AFTER PROPOSED APPROACH

Criteria	*Before*	*After*
Number of scenarios	15	24
Number of clauses	64	85
Clause per scenario ratio	4.26	3.54

There is another advantage of the proposed method. Since BDATs are transformed to ESGs and then combined, we have an ESG from which we can automatically generate test sequences, i.e., sequences of BDATs. CES for edge coverage computed by the TSD tool is shown below. There is only one test sequence for the whole BDATs. This shows that the proposed method improves the connectivity in such a way that there is no need for reset operations in test execution.

CES 111 events:
[, login page is displayed, enter username, enter password, click login button, home page is displayed, go to order list page, order list page is displayed, click on an order, order details are displayed, press OK button, order list page is displayed, click home icon, home page is displayed, click shopping cart button, shopping cart page is displayed, click check out button, check out page is displayed, enter new address, enter new invalid payment, confirm invalid order, "invalid payment" is displayed, press OK button, check out page is displayed, enter new address, enter new invalid payment, click cancel button, check out page is displayed, enter new address, enter new valid payment, click cancel button, check out page is displayed, select existing address, select existing payment, click cancel button, check out page is displayed, enter new address, enter new valid payment, confirm valid order, "order taken" is displayed, press OK button, order list page is displayed, click home icon, home page is displayed, enter multiple keyword, click search button, product list page is displayed, select a filter, click filter button, filtered product list page is displayed, click on a product, product details are displayed, select amount, add to cart, shopping cart page is displayed, click home icon, home page is displayed, enter single keyword, click search button, product list page is displayed, click on a product, product details are displayed, click home icon, home page is displayed, select a product list page, product list page is displayed, click home icon, home page is displayed, click account button, account page is displayed, update payment, "payment updated" is displayed, press OK button, account page is displayed, update address, "address updated" is displayed, press OK button, account page is displayed, click home icon, home page is displayed, click shopping cart button, shopping cart page is displayed, click check out button, check out page is displayed, select existing address, select existing payment, confirm valid order, "order taken" is displayed, press OK button, order list page is displayed, click home icon, home page is displayed, select a product list page, product list page is displayed, select a filter, click filter button, filtered product list page is displayed, click home icon, home page is displayed, click shopping cart button, shopping cart page is displayed, click check out button, check out page is displayed, click home icon, home page is displayed, click logout button, login page is displayed, enter username, enter password, click login button, home page is displayed, click logout button,],

CES for edge-pair coverage computed by the TSD tool has a complete event sequence of 224 events. The CES for edge-pair coverage is not given here because of space limitations.

B. *Evaluation of an E-commerce Software Test Suite*

Five teams of graduate students wrote BDATs for a bank ATM software [21], which can be found at https://github.com/esg4aspl/Connectivity-Improvement-for-Behavior-Driven-Acceptance-Tests. We applied Algorithm 1 to all five BDAT suites to compare them. The first three list length match percentages (rounded to two digits) are given in Table III. Table III shows that, on the average, 84% of possible tag matches are identified as the first suggestion by Algorithm 1. Average match percentages increase to 87% and 90% respectively, when second and third suggestions are added to the consideration. For TS4, all step definition matches are identified as the first result by the algorithm; while for TS2, all matches are present in top three suggestions.

All the results obtained after applying Algorithm 1 to five BDAT suites are drawn and shown in Figure 7. List length is normalized to account for varying step definition counts. Figure 7 shows that, apart from TS1, all TS have a match rate over 90% within the 5% of the list length (i.e., a correct match is present for 90% of the step definitions within the top 5% of suggested matches list). Match rate further increases to 95% for a 10% list length. For TS1, 90% and 95% match rates are possible at 40% and 65% of the list length respectively.

TABLE III. SCENARIO COUNT, TAG COUNT AND ALGORITHM 1 RESULTS FOR STUDENT GENERATED GHERKIN SCENARIOS

ID	*Scenario count*	*Tagged step definition count*	*Match rate for list length*		
			L=1	*L=2*	*L=3*
BDAT TS1	24	46	57%	61%	63%
BDAT TS2	18	25	88%	92%	100%
BDAT TS3	48	87	87%	92%	94%
BDAT TS4	17	31	100 %	100 %	100%
BDAT TS5	55	109	89%	91%	93%
AVERAGE	32.4	59.6	84%	87%	90%

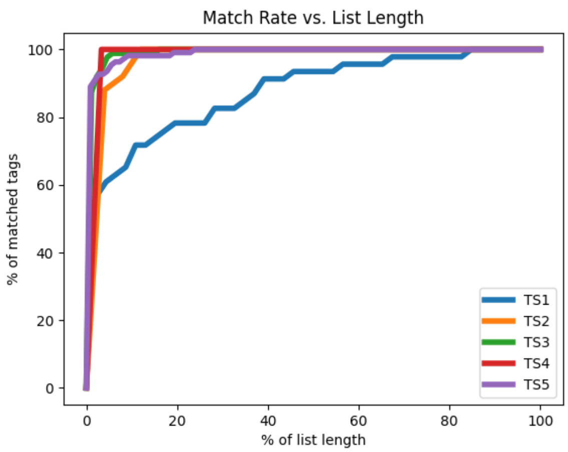


Figure 7. Number of correct matches for five BDAT suites under consideration.

Due to space constraints, we chose ESG of BDAT suite 2 for presentation. Elimination by combination enables us to find one missing BDAT, which is drawn in red on the resulting ESG in Figure 8. Further analysis showed that three scenarios are stuck and cannot lead to exit. For those, scenarios represented in green on the ESG in Figure 8 are added to improve connectivity. Finally, CES for edge coverage computed by the TSD tool is given below. This time there are four CESs. The TSD tool minimizes both the number of CESs and the number of events for efficiency.

CES 3 events: [, insert invalid cash card to ATM, ATM shows error message, eject the card,],

CES 5 events: [, insert valid cash card to ATM, redirect to password page, password page is shown, user clicks take card button, eject the card,],

CES 11 events: [, insert valid cash card to ATM, redirect to password page, password page is shown, enter wrong password, click correction password, clear password, password page is shown, enter wrong password 3 times, freeze the account, display frozen account warning, eject the card,],

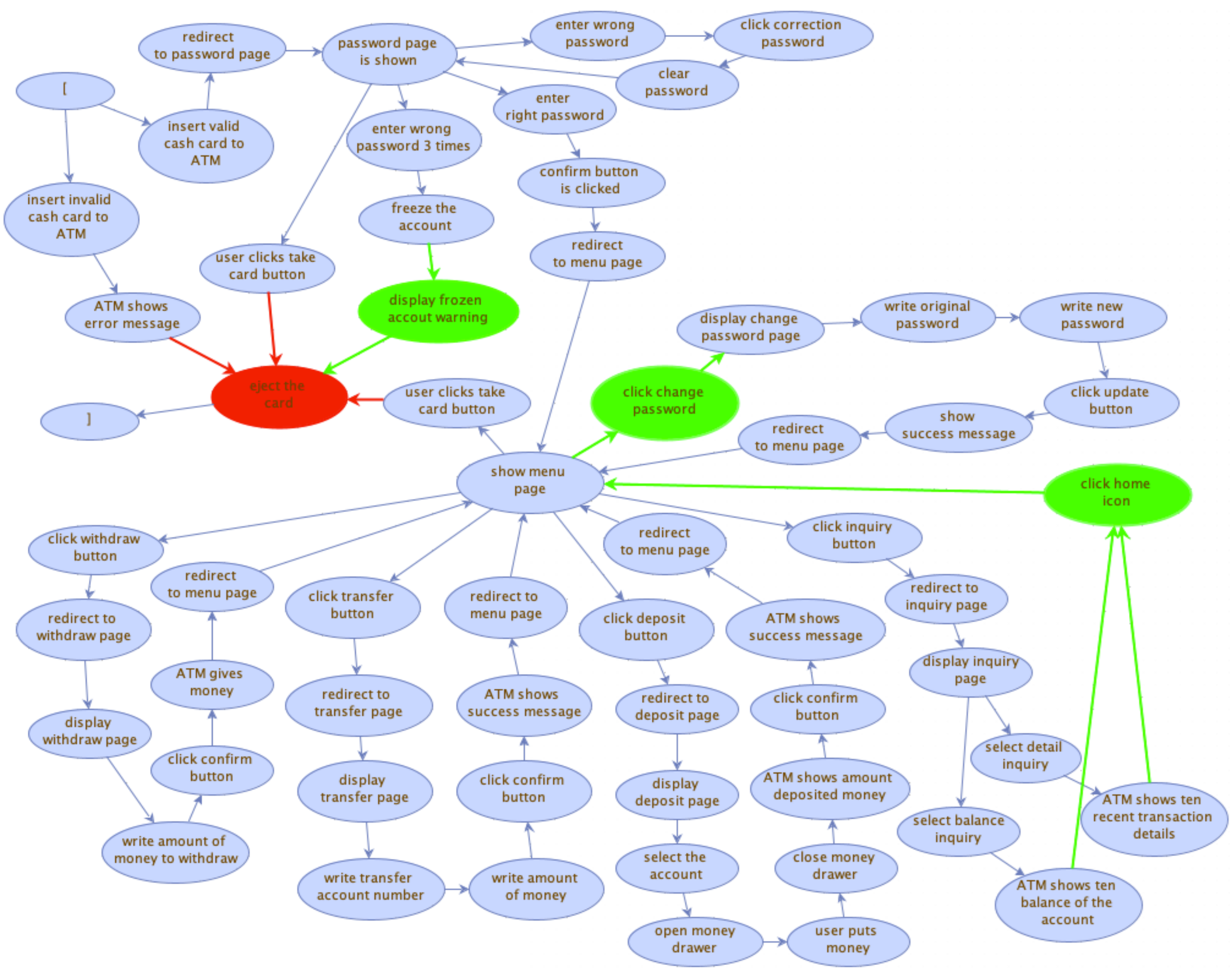


Figure 8. Composed ESG.

CES 60 events: [, insert valid cash card to ATM, redirect to password page, password page is shown, enter right password, confirm button is clicked, redirect to menu page, show menu page, click change password, display change password page, write original password, write new password, click update button, show success message, redirect to menu page, show menu page, click inquiry button, redirect to inquiry page, display inquiry page, select balance inquiry, ATM shows ten balance of the account, click home icon, show menu page, click transfer button, redirect to transfer page, display transfer page, write transfer account number, write amount of money, click confirm button, ATM shows success message, redirect to menu page, show menu page, click deposit button, redirect to deposit page, display deposit page, select the account, open money drawer, user puts money, close money drawer, ATM shows amount deposited money, click confirm button, ATM shows success message, redirect to menu page, show menu page, click withdraw button, redirect to withdraw page, display withdraw page, write amount of money to withdraw, click confirm button, ATM gives money, redirect to menu page, show menu page, click inquiry button, redirect to inquiry page, display inquiry page, select detail inquiry, ATM shows ten recent transaction details, click home icon, show menu page, user clicks take card button, eject the card,],

VI. Discussion

The proposed method assumes that Gherkin clauses can be represented by events. This assumption holds for the selected two BDAT suites used in the evaluation. We were able to represent all possible Gherkin clauses by events.

In the previous section, evaluation of Algorithm 1 results for five BDAT suites showed that Algorithm 1 performs significantly worse for TS1. In fact, excluding TS1 increases the first and third result hit rate in average to 91% and 97% up from 84% and 90%, respectively. To explain this discrepancy, TS1 Gherkin scenarios were examined, and the issue was traced back to the original set of requirements. In the original set of requirements, the system is described as two interacting modules. The team of the TS1 converted these requirements into 2 separate Gherkin features with different perspectives and terminologies. As a result, matching step definitions' semantic similarities were severely weakened across features. 85% of the missed first suggestion matches for TS1 were observed to be between step definitions from different features. This observation also leads us to a known fact that BDATs should be written without considering any design or implementation issues.

The evaluation of the second case reveals that using NLP techniques on written BDATs helps us improve the connectivity of BDATs. Moreover, the proposed method shows that through modeling BDATs, it is possible to generate test sequences automatically. UML use case diagrams and activity diagrams can also be used for modeling BDATs and then automatically generate tests. The research in this area is explained in the related work section.

Scalability of the models is an important concern. ESGs allow us to work on some small and modular models through sub-ESGs [12]-[15] like subroutines. The TSD tool is also designed to support sub-ESGs. This way, it is possible to generate manageable large models. Moreover, these sub-ESGs can be flattened into one large ESG if necessary.

VII. Threats to Validity

One threat to validity is internal validity, which deals with the effects on the evaluation. The selection of BDAT test suite used in evaluation is obtained by searching GitHub repositories. This cannot be considered as random selection. Moreover, the proposed method is applied to the selected BDAT test suite by the author.

Another threat to validity is external validity, which deals with the generalizability of the results. The evaluation in this study is based on a single BDAT test suite. Although this test suite is developed for e-commerce software, which may represent business software generally, evaluation of other BDAT test suites from different domains with the proposed method will help generalize the results.

VIII. Related Work

Tuglular [22] proposed a model-based approach for feature-oriented testing using Event Sequence Graphs (ESGs). In this approach, ESGs are extended to save state and pass it to the following ESG. This way, tests written for features can be combined on state information. However, capturing state is not always possible for acceptance tests.

UML use case diagrams can also be used for modeling BDATs and then automatically generate tests. Gutierrez et al. [23] proposed an approach for working with Gherkin scenarios using UML use case models. They transform from the UML use case diagrams to the Gherkin plain text syntax. They also developed a tool for running Gherkin scenarios in UML as test cases.

Alferez et al. [24] proposed an approach, named AGAC (Automated Generation of Acceptance Criteria), which supports the automated generation of AC specifications in Gherkin. They used UML use case diagrams and activity diagrams to create specifications, derive acceptance criteria from them, and then generate test cases from derived acceptance criteria. UML activity diagrams are not formally defined as directed graphs and therefore, in this work we choose to use formally defined ESGs to benefit from existing algorithms in directed graphs. However, with the help of some theoretical background UML activity diagrams can be used instead of ESGs.

Kudo et al. [25] proposed the software pattern meta model that bridges requirement patterns to groups of scenarios with similar behaviors in the form of test patterns. This meta model is used to describe the behavior of a requirement pattern through a time executable and easy-to-use language aiming at the automatic generation of test patterns.

Wanderley and da Silveria [26] proposed using a mind model specification, which serves as a basis for transforming the definitions of the scenario and generating a conceptual model represented by a UML class diagram. The mind model functions as a bond that represents the business entities, and enables simple association, aggregation, and composition relationships between the entities.

An adjacent area is process discovery in business process management literature. Rozinat and van der Aalst [27] worked on whether event logs conform to the process model and vice versa. They proposed two dimensions of conformance, namely fitness and appropriateness, to be checked along with corresponding metrics. They developed a Conformance Checker within the ProM Framework.

Beschastnikh et al. [28] proposed algorithms for inferring communicating finite state machine models from traces of concurrent systems, and for proving them correct. They also provided an implementation called CSight, which helps developers find bugs.

Pecchia et al. [29] proposed an approach that employs process mining for detecting failures from application logs. Their approach discovers process models from logs; then it uses conformance checking to detect deviations from the discovered models. They were able to quantify the failure detection capability of conformance checking despite missing events, and its accuracy with respect to process models obtained from noisy logs [29].

As a novel approach, this work aims to transform executable specification in Gherkin language to an ESG. Additionally, this work introduces a novel methodic analysis on BDATs that can reveal missing BDATs.

IX. Conclusion

This paper proposes a method to improve the connectivity of behavior-driven acceptance tests. The method utilizes NLP techniques and ESGs. With the proposed method, the test designer not only finds and completes missing BDATs, but also combines them to know which BDAT can be executed after which BDAT. When the final composition is supplied to the TSD tool, it automatically generates a test sequence that covers all BDATs. So, the proposed method improves the connectivity of BDATs.

As future work, we plan to enhance the developed tool with new capabilities to further aid in the design and application of acceptance tests. Also, as future work, our goal is to enhance the tool with ontologies so semantically related scenarios are easily decoded. Moreover, we plan to use UML activity diagrams instead of event sequence graphs and compare their advantages and disadvantages. Finally, we will apply all these improvements to large Gherkin-based specifications and acceptance criteria.

References

[1] T. Tuglular, “On the Composability of Behavior Driven Acceptance Tests,” in The Seventh International Conference on Advances and Trends in Software Engineering (SOFTENG 2021), 2021, pp. 1–4.

[2] M. G. Cavalcante and J. I. Sales, "The Behavior Driven Development Applied to the Software Quality Test," Proc. 14th Iberian Conference on Information Systems and Technologies (CISTI), IEEE, 2019, pp. 1–4.

[3] Cucumber Gherkin. https://cucumber.io/docs/gherkin/reference/. [retrieved: March, 2021].

[4] Gauge. https://gauge.org. [retrieved: March, 2021].

[5] T. Tuglular and S. Şensülün, “SPL-AT Gherkin: A Gherkin Extension for Feature Oriented Testing of Software Product Lines,” in IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), 2019, vol. 2, pp. 344–349.

[6] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Information processing & management, vol. 24, no. 5, pp. 513–523, 1988.

[7] P. Sitikhu, K. Pahi, P. Thapa, and S. Shakya, “A comparison of semantic similarity methods for maximum human interpretability,” in 2019 artificial intelligence for transforming business and society (AITB), 2019, vol. 1, pp. 1–4.

[8] A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, “Cosine similarity to determine similarity measure: Study case in online essay assessment,” in 2016 4th International Conference on Cyber and IT Service Management, 2016, pp. 1–6.

[9] S. Kannan et al., “Preprocessing techniques for text mining,” International Journal of Computer Science & Communication Networks, vol. 5, no. 1, pp. 7–16, 2014.

[10] S. Vijayarani, M. J. Ilamathi, and M. Nithya, “Preprocessing techniques for text mining-an overview,” International Journal of Computer Science & Communication Networks, vol. 5, no. 1, pp. 7–16, 2015.

[11] T. Tuglular, F. Belli, and M. Linschulte, "Input contract testing of graphical user interfaces," International Journal of Software Engineering and Knowledge Engineering, 26(02), 2016, pp. 183–215.

[12] F. Belli and C. J. Budnik, "Test minimization for human-computer interaction," Applied Intelligence, 26(2), 2007, pp. 161–174.

[13] F. Belli, C. J. Budnik, and L. White, "Event based modelling, analysis and testing of user interactions: approach and case study," Software Testing, Verification and Reliability, 16(1), 2006, pp. 3–32.

[14] F. Belli and C. J. Budnik, "Minimal spanning set for coverage testing of interactive systems," International Colloquium on Theoretical Aspects of Computing. Springer, Berlin, Heidelberg, 2004, pp. 220–234.

[15] T. Tuglular, C. A. Muftuoglu, F. Belli, and M. Linschulte, "Event-based input validation using design-by-contract patterns," Proc. 20th International Symposium on Software Reliability Engineering, ISSRE’09, IEEE Press, 2009, pp. 195–204.

[16] G. Chartrand, “Introductory Graph Theory,” Courirer Corporation, 1977.

[17] E. Evans, "Domain-Driven Design: Tackling Complexity in the Heart of Software," Addison-Wesley Professional, 2003.

[18] S. Bird, E. Klein, and E. Loper, Natural language processing with Python, 1st ed. Beijing ; Cambridge [Mass.]: O’Reilly, 2009.

[19] TestSuiteDesigner. http://download.ivknet.de/. [retrieved: March, 2021].

[20] Barzilay, "Example of an ECommerce cucumber web test automation suite,". https://github.com/spriteCloud/ecommerce-cucumber-web-test-automation-suite. [retrieved: March, 2021].

[21] Team-111, “Bank ATM software,” https://github.com/Team-1111/bankbankatm. [retrieved: June, 2021]

[22] T. Tuglular, "Event sequence graph-based feature-oriented testing: A preliminary study," 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE, 2018, pp. 580–584.

[23] J. J. Gutiérrez, I. Ramos, M. Mejías, C. Arévalo, J. M. Sánchez-Begines, and D. Lizcano, "Modelling Gherkin Scenarios Using UML," Proc. 26th International Conference on Information Systems Development (ISD), 2017.

[24] M. Alferez, F. Pastore, M. Sabetzadeh, L. Briand, and J. R. Riccardi, "Bridging the gap between requirements modeling and behavior-driven development," 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS), IEEE, 2019, pp. 239–249.

[25] T. N. Kudo, R. F. Bulcão-Neto, and A. M. Vincenzi, "A conceptual metamodel to bridging requirement patterns to test patterns," Proc. of the XXXIII Brazilian Symposium on Software Engineering. 2019, pp. 155–160.

[26] F. Wanderley and D. S. da Silveria, "A framework to diminish the gap between the business specialist and the software designer," 2012 Eighth International Conference on the Quality of Information and Communications Technology. IEEE, 2012, pp. 199–204.

[27] A. Rozinat and W.M.P. van der Aalst, "Conformance testing: Measuring the fit and appropriateness of event logs and process models," Proc. 4th Business Process Management Workshops, Springer, 2006, pp. 163–176.

[28] I. Beschastnikh, Y. Brun, M.D. Ernst, and A. Krishnamurthy, "Inferring models of concurrent systems from logs of their behavior with CSight," Proc. 36th International Conference on Software Engineering, ACM, 2014, pp. 468–479.

[29] A. Pecchia, I. Weber, M. Cinque, and Y. Ma, “Discovering process models for the analysis of application failures under uncertainty of event logs,” Knowledge-Based Systems, vol. 189, p. 105054, 2020.