COMPUTING

edge

- Data Engineering
- Sustainable Computing
- Internet of Things
- Careers

FEBRUARY 2024
www.computer.org

IEEE COMPUTER SOCIETY
IEEE

IEEE COMPUTER SOCIETY D&I FUND

Drive Diversity & Inclusion in Computing

Supporting projects and programs that positively impact diversity, equity, and inclusion throughout the computing community.

DONATE TODAY!

IEEE COMPUTER SOCIETY

IEEE Foundation

COMPUTING edge

IEEE COMPUTER SOCIETY computer.org

IEEE COMPUTER SOCIETY IEEE Printed with inks containing soy and/or vegetable oils

STAFF

Editor
Lucy Holden

Periodicals Portfolio Senior Managers
Carrie Clark and Kimberly Sperka

Director, Periodicals and Special Projects
Robin Baldwin

Production & Design Artist
Carmen Flores-Garvey

Periodicals Operations Project Specialist
Christine Shaughnessy

Senior Advertising Coordinator
Debbie Sims

Circulation: *ComputingEdge* (ISSN 2469-7087) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821 4010; IEEE Computer Society Headquarters, 2001 L Street NW, Suite 700, Washington, DC 20036.

Postmaster: Send address changes to *ComputingEdge*-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *ComputingEdge* does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity, and space.

Reuse Rights and Reprint Permissions: Educational or personal use of this material is permitted without fee, provided such use: 1) is not made for profit; 2) includes this notice and a full citation to the original work on the first page of the copy; and 3) does not imply IEEE endorsement of any third-party products or services. Authors and their companies are permitted to post the accepted version of IEEE-copyrighted material on their own Web servers without permission, provided that the IEEE copyright notice and a full citation to the original work appear on the first screen of the posted copy. An accepted manuscript is a version which has been revised by the author to incorporate review suggestions, but not the published version with copy-editing, proofreading, and formatting added by IEEE. For more information, please go to: http://www.ieee.org/publications_standards/publications/rights/paperversionpolicy.html. Permission to reprint/republish this material for commercial, advertising, or promotional purposes or for creating new collective works for resale or redistribution must be obtained from IEEE by writing to the IEEE Intellectual Property Rights Office, 445 Hoes Lane, Piscataway, NJ 08854-4141 or pubs-permissions@ieee.org. Copyright © 2024 IEEE. All rights reserved.

Abstracting and Library Use: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee indicated in the code at the bottom of the first page is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

Unsubscribe: If you no longer wish to receive this *ComputingEdge* mailing, please email IEEE Computer Society Customer Service at help@computer.org and type "unsubscribe *ComputingEdge*" in your subject line.

IEEE prohibits discrimination, harassment, and bullying. For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

IEEE Computer Society Magazine Editors in Chief

Computer
Jeff Voas, *NIST*

Computing in Science & Engineering
İlkay Altintaş, *University of California, San Diego (Interim EIC)*

IEEE Annals of the History of Computing
David Hemmendinger, *Union College (Interim EIC)*

IEEE Computer Graphics and Applications
André Stork, *Fraunhofer IGD and TU Darmstadt*

IEEE Intelligent Systems
San Murugesan, *Western Sydney University*

IEEE Internet Computing
Weisong Shi, *University of Delaware*

IEEE Micro
Hsien-Hsin Sean Lee, *Intel Corporation*

IEEE MultiMedia
Balakrishnan Prabhakaran, *University of Texas at Dallas*

IEEE Pervasive Computing
Fahim Kawsar, *Nokia Bell Labs and University of Glasgow*

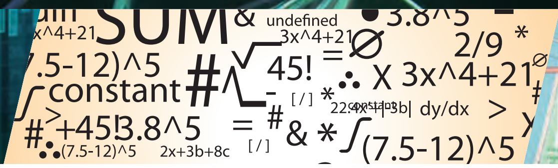
IEEE Security & Privacy
Sean Peisert, *Lawrence Berkeley National Laboratory and University of California, Davis*

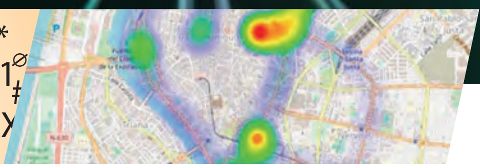
IEEE Software
Sigrid Eldh, *Ericsson*

IT Professional
Charalampos Z. Patrikakis, *University of West Attica*

FEBRUARY 2024 · VOLUME 10 · NUMBER 2

COMPUTING edge





Subscribe to *ComputingEdge* for free at www.computer.org/computingedge.

Magazine Roundup

The IEEE Computer Society's lineup of 12 peer-reviewed technical magazines covers cutting-edge topics ranging from software design and computer graphics to Internet computing and security, from scientific applications and machine intelligence to visualization and microchip design. Here are highlights from recent issues.

Computer

Teaching Edge AI at the Undergraduate Level: A Hardware–Software Co-Design Approach

The authors of this article from the November 2023 issue of *Computer* have built a hardware prototype specifically designed for teaching edge artificial intelligence (AI), enabling students to easily develop software for edge devices. This complements the University of Texas curriculum with a foundational edge AI course where students can experiment with real edge devices directly.

Computing in Science & Engineering

A Finite-Element-Based Cohesive Zone Model of Water-Filled Surface Crevasse Propagation in Floating Ice Tongues

The authors of this article from the May/June 2023 issue of *Computing in Science & Engineering* performed several numerical studies to explore the parametric sensitivity of surface crevasse depth to ice rheology, cohesive strength, density, and temperature for different levels of meltwater depth. They found that viscous (creep) strain accumulation promotes crevasse propagation and that surface crevasses propagate deeper in ice shelves/tongues if they consider depth-varying ice density and temperature profiles. Therefore, ice flow models must account for depth-varying density and temperature-dependent viscosity to appropriately describe calving outcomes.

IEEE Annals of the History of Computing

Socialist AI? Societal Use, Economic Implementation, and the Tensions of Applied Computer Science in Late Socialist GDR

AI research and development in the German Democratic Republic (GDR) can be seen as paradigmatic for the phase of late socialism. The GDR adopted a pragmatic approach to AI, described in this article from the July–September 2023 issue of *IEEE Annals of the History of Computing*. By analyzing AI objects, particularly chess computers, as indicators of the proliferation of AI within the everyday practices of the GDR, and through a thorough examination of a widely circulated computer magazine, this overview illustrates the diversity of AI applications, spanning areas such as planning, industrial management, state administration, science, public health, and pervasive surveillance.

IEEE Computer Graphics and Applications

Identifying Visualization Opportunities to Help Architects Manage the Complexity of Building Codes

While visualizations have been used to support general architectural design exploration, existing computational solutions treat building codes as separate from the design process, creating challenges for architects. Through a series of participatory design studies, the authors of this article from *IEEE Computer Graphics and Applications*' November/December 2023 issue found that interactive visualizations have the potential to aid design exploration and sensemaking in early stages of architectural design. To tackle challenges relating to complexity and ambiguity in building codes, the authors propose various user-driven knowledge management mechanisms for integrating,

2469-7087/24 © 2024 IEEE

negotiating, interpreting, and documenting building code rules.

IEEE Intelligent Systems

Effectively Modeling Sentence Interactions With Factorization Machines for Fact Verification

Fact verification is a very challenging task that requires retrieving multiple evidence sentences from a reliable corpus to authenticate claims. To alleviate limitations with existing fact-verification models, the authors of this article from the September/October 2023 issue of *IEEE Intelligent Systems* propose select and fact verification modeling (SFVM) using a multihead self-attention mechanism combined with a gating mechanism to facilitate sentence interaction and enhance sentence embeddings. They then use factorization machines to effectively express the compressed alignment vectors, which are then used to expand the representations of the base evidence.

IEEE Internet Computing

Measuring the Energy of Smartphone Communications in the Edge-Cloud Continuum: Approaches, Challenges, and a Case Study

As computational resources are placed at different points in the edge-cloud continuum, not only is the responsiveness on the client side affected, so too is the amount of energy spent during communications. In this article from *IEEE Internet Computing*'s November/December 2023 issue, the authors summarize the main approaches used to estimate the energy consumption of smartphones and the main difficulties typically encountered, and they present a case study on how to put such approaches into practice.

IEEE micro

MetaE2RL: Toward Meta-Reasoning for Energy-Efficient Multigoal Reinforcement Learning With Squeezed-Edge You Only Look Once

In this article from *IEEE Micro*'s November/December 2023 issue, the authors propose meta-reasoning for energy efficiency of multigoal RL, a hardware-aware framework that incorporates low-power preprocessing solutions and meta-reasoning to enable deployment of multigoal RL on tiny autonomous devices. For this aim, a meta-level is proposed to allocate resources efficiently in real time by switching between models with different complexities.

IEEE MultiMedia

Bandwidth-Aware High-Efficiency Video Coding Design Scheme on a Multiprocessor System on Chip

The experimental and implementation results found by the authors of this article from the July–September 2023 issue of *IEEE MultiMedia* demonstrate that the proposed CB-TB (coding bandwidth/transmission bandwidth) rate-coding distortion algorithm modeling and the very large-scale integration hardware architecture are applicable for CB- and TB-constrained H.265/high-efficiency video coding (HEVC) encoder design within multiprocessor system-on-chip systems (MPSoCs).

IEEE pervasive COMPUTING

Exploiting Radio Fingerprints for Simultaneous Localization and Mapping

The authors of this article from the July–September 2023 issue of *IEEE Pervasive Computing* propose a novel approach for localization and mapping of autonomous vehicles using radio fingerprints, for example wireless fidelity or long term evolution radio features, which are widely available in the existing infrastructure. The effectiveness of their system is demonstrated in three different environments, namely outdoor, indoor building, and semi-indoor environment.

IEEE SECURITY & PRIVACY

Challenges of Producing Software Bill of Materials for Java

Software bills of materials (SBOMs) promise to become the backbone of software supply chain hardening. The authors of this article from the November–December 2023 issue of *IEEE Security & Privacy* deep-dive into six tools and the SBOMs they produce for complex open source Java projects, revealing challenges regarding the accurate production and usage of SBOMs.

IEEE Software

Resist the Hype!: Practical Recommendations to Cope With Résumé-Driven Development

The authors of this article from the September–October 2023 issue of *IEEE Software* initiate a more serious debate about the consequences of Résumé-Driven Development on software development practice. The authors explain how this previously anecdotally and somewhat humorously reported phenomenon may constitute a harmful self-sustaining dynamic and provide practical recommendations for the hiring and applicant perspectives to change the current situation for the better.

IT Professional

Enhancing Communication Among Remote Cybersecurity Analysts With Visual Traces

This article from the September–October 2023 issue of *IEEE Professional* addresses the communication challenges in collaborative cybersecurity analysis amid the rise of remote work during the postpandemic era. The authors introduce a novel approach that leverages visual traces of analysts' analytical processes to enhance information transmission and processing in collaborative cybersecurity analysis, and they demonstrate the advantages of visual traces in supporting communication in a case study.

Join the IEEE Computer Society
computer.org/join

THE IEEE APP:
Let's stay connected...

Stay connected by discovering the valuable tools and resources of IEEE:

- Create a personalized experience
- Get geo and interest-based recommendations
- Schedule, manage, or join meetups virtually
- Read and download your IEEE magazines
- Stay up-to-date with the latest news
- Locate IEEE members by location, interests, and affiliations

Download Today!
Download on the App Store
GET IT ON Google Play
IEEE

How to Effectively Manage Data Engineering

When the initial excitement of any great scientific advancement fades and that advancement falls into regular use, scientists must turn their attention to an ordinary, but integral, problem—management. Management is key to enabling smooth, efficient, and effective processes in the long term. This issue of *ComputingEdge* explores relevant questions and possible solutions around how to manage data, including its migration, collection, and maintenance. The articles also touch on how to approach carbon-aware computing, particularly in regard to high-energy consumption activities, such as bitcoin mining.

The question of how to effectively manage data engineering poses complex problems for both individuals and large organizations on a global scale. The authors of “Migrating the Communication Protocol of Client–Server Applications,” from *IEEE Software*, propose a tool-based approach for how to migrate services and data structures for an international IT company. In *IT Professional*’s “Doing for Data What the Internet Did for Networking,” the authors analyze an approach to implementing FAIR, an international movement to effectively manage scientific data.

In recent years, the high energy costs inherent to blockchain, data centers, and supercomputing has led to increased public scrutiny. This may force engineers to assess the impact of their work more closely. In *Computer*’s “‘Frameworking’ Carbon-Aware Computing Research,” the authors discuss how to approach carbon-aware computing. *Computer* article “On the High-Energy Consumption of Bitcoin Mining” explains why high energy consumption is necessary to maintain and mine Bitcoin’s blockchain.

The expansion of the Internet-of-Things requires better systems for handling data. The authors of “Elastic Data Analytics for the Cloud-to-Things Continuum,” from *IEEE Internet Computing*, suggest a strategy for managing cloud and communications infrastructure by achieving elastic analytics. *IEEE Intelligent Systems*’ “Data-Driven Predictive Maintenance” underscores the importance of predictive data maintenance in today’s industry.

Finally, this issue closes with the remarkable stories of two individuals who accomplished great feats in computer science. In *IEEE Computer Graphics and Applications*’ “From Holocaust Hidden Child to Computer Animation Laboratory,” Marceli Wein shares how he escaped the Holocaust and grew up to help create *Hunger/La Faim*, which was nominated for an Academy Award as the first computer-animated film. “The Legacy of Mary Kenneth Keller, First U.S. Ph.D. in Computer Science,” from *IEEE Annals of the History of Computing*, introduces readers to Sister Kenneth who was a nun, a high school math teacher, and the first person to earn a Ph.D. in computer science.

EDITORS: Cesare Pautasso, University of Lugano, c.pautasso@ieee.org
Olaf Zimmermann, Eastern Switzerland University of Applied Sciences (OST), Rapperswil, olaf.zimmermann@ost.ch

This article originally appeared in IEEE Software vol. 40, no. 4, 2023

DEPARTMENT: INSIGHTS

Migrating the Communication Protocol of Client–Server Applications

Gabriel Darbord, Benoît Verhaeghe, Anne Etien, Nicolas Anquetil, Anas Shatnawi, Abderrahmane Seriai, and Mustapha Derras

FROM THE EDITORS

To live long and prosper, information systems need to survive shifts in their underlying technology platforms. This often requires switching integration styles, API designs, and communication protocols. In this "Insights," the authors discuss how they leveraged design patterns and coding conventions to develop a semiautomated migration tool to advance four large management systems (financial, customers, human resources), implemented in Java and Spring, from GWT-RPC and RMI to JSON over HTTP, presenting a pragmatic, scalable, and cost-efficient way to remove technical debt. —*Cesare Pautasso and Olaf Zimmermann*

Berger-Levrault is an international software publisher providing management solutions for citizens, families, elected officials, health-care providers, educators, and schools. Its applications rely on various technologies, some of which are very old and/or no longer maintained.[1,2] The company has been developing software systems using Java technologies for more than 25 years, and with GWT since its release. As part of a modernization effort to escape outdated legacy technologies, the company wants to migrate its applications to newer technologies. We are participating in a project to migrate client–server applications (Swing or GWT) to Angular (which uses TypeScript) for the client part, and Spring Boot for the server part. In this article, we consider four industrial applications, with sizes between 300 and 2700 thousands of lines of code (KLOC) and counting between 500 and 4400 services. The legacy client–server Swing application uses Java's RMI for communication between the Java Swing client and the Java server. The legacy GWT applications use the GWT implementation of Remote Procedure Call (GWT-RPC) for communication between the Java GWT client and the Java server. For the new client–server communication, the company chose to develop an HTTP application programming interface (API) using JavaScript Object Notation (JSON) to exchange data because these technologies are widely accepted de facto standards and easy to implement in both Angular and Spring Boot. An HTTP API corresponds to a low level of REST maturity (https://martinfowler.com/articles/richardsonMaturityModel.html). Aiming for a higher level would have required deeper changes and made the migration more difficult.

In all of these projects, the server will remain largely the same with only minor adjustments to handle the HTTP requests. The client part of the applications will be migrated to Angular using the approach proposed in Verhaeghe et al.[2] and will not be discussed here.

In this article, we focus on the automatic migration of client–server communication from the legacy RMI and GWT-RPC to the planned HTTP API. To design a

Digital Object Identifier 10.1109/MS.2023.3263019
Date of current version: 14 July 2023

 2469-7087/24 © 2024 IEEE

communication migration tool, we had to address the following four issues: 1) identify all of the services and data exchange classes 2) adapt how clients invoke server-side services 3) rewrite data exchange classes in a different language, and 4) in some cases, we had to minimize the amount of exchanged data to solve introduced performance issues. Data exchange classes correspond to the type of data structures that are transferred between the clients and the server, and data exchange objects correspond to their instances. They can be thought of as data transfer objects (https://martinfowler.com/eaaCatalog/dataTransferObject.html), but in our case some data exchange objects contain behavior: methods that are more complex than simple accessors. In addition, for the remainder of this article, the term *calling convention* will be used to refer to the manner clients invoke server-side services.

We experimented with our approach on four applications from Berger-Levrault. We report here some results of the migrations. This is a work in progress, and the migrated applications are still undergoing validation by their respective development teams. Throughout the process, we are assisted by static analysis tools, including a custom linter that helps ensure the quality and consistency of the code to be migrated.

This experience report is structured as follows: First, we present the migration context and define the vocabulary used. Then, we describe the issues encountered in this project. We continue by describing our solution. And finally, we report some data from running the tool on four industrial applications.

MIGRATION CONTEXT

A migration approach to completely rewrite a monolithic GWT application into two separate projects, a JavaScript client and a Java server, communicating via a RESTful HTTP API is proposed by Zirkelbach et al.[3] This approach allows to eliminate the technical debt in the old application and create a client–server architecture with well-designed services. Berger-Levrault rejected this solution because it is costly and carries all of the usual risks of software development projects. Instead, the company opted for an automated migration that might result in a less optimal application but would deliver results faster and at a lower cost. In this context, understanding the client–server communication is essential for a successful migration. This section introduces the factors to consider.

Client–Server Communication refers to the way both sides of a distributed application interact and exchange data. To migrate client–server communication, we must consider the client's web *pages*, the *services* provided by the server, and the *data exchange objects* that are exchanged between them.

The *pages* contain the representation of the GUI and the behavioral code that is executed when the user interacts with it. Their migration is beyond the scope of this article (see, for example, Verhaeghe et al.[2]) These pages display data obtained from the server's *services*.

The *services* are server-side constructs that must also be described in the client to ensure interoperability. This enables the connection between the two parts of the applications. For example, in a human resources application, an "employee service" could be used to query employees. Technologies such as RMI and GWT-RPC in Java use proxies on the client to transparently manipulate objects created remotely on the server. When invoked, a client "service" (actually a service descriptor) creates a request to the corresponding server service and waits for the response. When it receives that response, it passes it back to the original caller (i.e., a page or another client service descriptor).

The application client and server exchange data through what we call *data exchange objects*. A service response typically contains one or more instances of a data exchange class, such as a list of employees associated with a particular request.

MIGRATION ISSUES

To automatically migrate client–server communication from RMI and GWT-RPC to an HTTP API, the following issues must be addressed as follows:

- All services must be automatically identified, as well as their calling and receiving locations in the source code.
- Data exchange classes, used to transfer data to and from services, must also be identified and their representation converted from Java or TypeScript to JSON (serialization/deserialization).

- The service calling convention must be adapted: With RMI and GWT-RPC, the client invokes a Java method directly, whereas with an HTTP API in Spring Boot, the client sends an HTTP request to the server, which is handled by a controller.
- In addition, we needed to minimize the amount of exchanged data (payload) in some cases.

Identifying the Services

To migrate the client–server communications, one must identify all calls to services in the client and all of the definitions of those services in the server. We will see in a next section that this step is not difficult in the cases of RMI or GWT-RPC because they follow specific source code conventions (class annotation or class inheritance).

Identifying the Data Exchange Classes

The exchanged data must also be considered. First, one needs to identify what data are being exchanged, i.e., what classes define data exchange objects. Then, one must make sure that it can be serialized and deserialized correctly. RMI and GWT-RPC technologies rely on a proprietary binary serialization format that is handled transparently by the frameworks themselves. On the other hand, REST is typically associated with a generic and standard serialization format such as JSON, and data (de)serialization must be handled explicitly by the developer. This introduces some difficulties, which we explore in the following paragraphs.

A data exchange class corresponds to a *type* in Java, TypeScript, or any other language. The JSON format, which has its own Multipurpose Internet Mail Extensions type of "application/json," contains only primitive types, arrays, and dictionaries. JSON primitive types are string, number, and Boolean; JSON arrays can contain any number of "JSON types," and JSON dictionaries are key-value collections where the keys are strings, and the values are "JSON types." There is no structural description in JSON for runtime data. Objects are transferred as field-name/field-value dictionaries. This is a problem because dictionaries and objects that are serialized to JSON have exactly the same format.

Therefore, the serializer and the deserializer must agree on the structure of the transferred data beforehand. This is a known issue of "data format" described in the introduction to integration styles by Hohpe et al.[4] Even then, there are still difficulties in the case of transferring polymorphic objects. For example, to deserialize a collection of `AbstractEmployee`, a deserializer would need to distinguish instances of `FullTimeEmployee` from instances of `PartTimeEmployee`.

There is also the case of object references. If a field of a serialized object contains another object, the latter must be serialized recursively. If the second object contains a reference back to the first, this creates a known issue with serializing objects with circular references.

A final problem is that RMI and GWT-RPC ignore field visibility. When an object is sent from the server to the client, the frameworks are able to serialize all of the object fields, even the private ones. On the other hand, some modern JSON libraries, such as Jackson, only (de)serialize fields with public accessors.

Adapting the Calling Convention

The calling conventions for a service are different between RMI/GWT-RPC and an HTTP API. RMI and RPC invoke a service by executing a remote class method. On the other hand, modern RESTful solutions invoke a service using an HTTP request comprising a resource locator (the URL) and an HTTP verb (POST, GET, PUT, DELETE, *and so on*, see https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html). Therefore, client-side invocations must be adapted to the conventions of an HTTP API.

Minimize Amount of Exchanged Data

After the initial experiments, we encountered some performance issues. While evaluating the performance of a migrated application, we noticed that the exchange of some messages became noticeably slower (from near instantaneous, to tens of seconds). This often happened on pages displaying a table of objects, where the full objects are transferred (as described by the embedded entity pattern by Zimmermann et al.[5, p.314]) while only a few fields are actually displayed. Profiling showed that this was due to deserialization in the client for multimegabyte payloads. This is a known problem with JavaScript JSON processing libraries, as noted in previous work on web front-end migration.[6] Because RMI and GWT-RPC do not have this problem, the application developers did not focus on making smaller payloads.

MIGRATION PROCESS

This article proposes a process for migrating client–server communication that addresses all of the previous issues. We describe here our solutions for 1) identifying the services and data exchange classes, 2) adapting the calling conventions of services, 3) migrating data exchange classes, and 4) pruning data exchange objects for performance issues.

Services and Data Exchange Classes Identification

The services are identified using coding conventions, by static analysis of the applications. In the case of RMI, the framework requires service classes to implement the `Remote` interface, and their methods to be able to throw a `RemoteException` to be exposed. For GWT-RPC, service classes are required to implement the `RemoteService` interface, and all of the public methods are exposed. Once a method implementing a service is identified, its return and parameter types can be identified as data exchange classes.

Adapt Calling Conventions

In the legacy applications, services are implemented by methods in specific classes (as previously mentioned). For ease of implementation, we decided to keep the same convention for the migrated applications. Thus, each class in a legacy implementation is migrated as a class in its new implementation, and each method in RMI or GWT-RPC is migrated as an HTTP API endpoint. To simplify the migration, we also use only the HTTP verb: POST. This results in operating at REST maturity level one.

On the server side, we automatically generated wrapper services which make use of Spring Boot, where they are called *controllers*. The generated controller for a `methodX` will accept HTTP requests and handle JSON deserialization, then delegate the execution of the service to the old `methodX`. This is illustrated in Figure 1 and Listing 1.

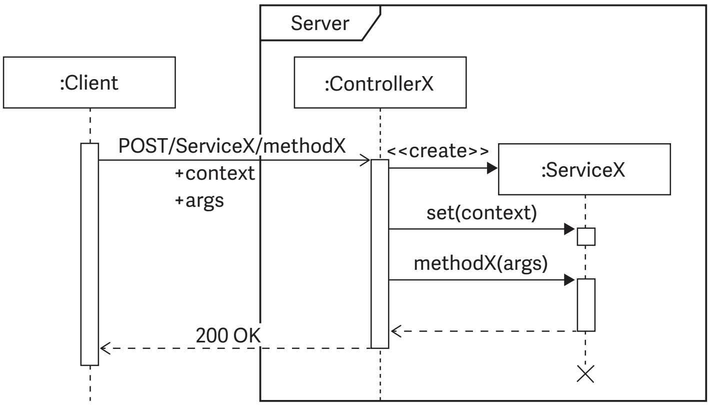


FIGURE 1. Wrapping of legacy RMI service into a new Spring controller.

```
@RestController
@PostMapping(path = "/ServiceX",
  consumes = "application/json",
  produces = "application/json")
public class ServiceXController {
@PostMapping("/methodX")
public ResultType methodX(
  @RequestBody Map<String, Object> args)
      throws RemoteException {
    //get arguments from JSON payload
    final ArgType1 arg1 = args.get("arg1");
    final ArgType2 arg2 = args.get("arg2");
    //get other arguments…
    final User _user = args.get("_user");

    final ServiceX service = new ServiceX();
    service.setUser(_user);
    return service.methodX(arg1, arg2,…);
  }
}
```

LISTING 1. Automatically generated controller with a `methodX` API endpoint. The strings "arg1," "arg2," and so on represent the argument names of the old service's method.

Note that the `RemoteException` on line 10 is a remnant of the RMI implementation that required services to be able to throw this exception. The exception is there because we avoided modifying the original services, but it will never be thrown because RMI is no longer used. Another way to handle it would be to add a try block with an empty catch statement. If we had not decided to keep as much of the original code as possible, we could have refactored the original service to remove the reference to the RMI interface and not declare the throwing of the RMI exception.

In the legacy applications, services have a small context: an object that represents the authenticated user of the application. Our solution is to use the context representation pattern,[5, p.96] and embed this small context in each request as a specific argument (line 15) of the service invocation. This context is then set in the `ServiceX` instance (line 18) before the method is called.

Data Exchange Class Migration

We discussed earlier that there are several problems associated with using JSON to transfer data: no structure description, polymorphic objects, and circular references. We need to use a JSON library that handles these problems for us. But since the language is not the same for both communicating parts, Java on the server side and TypeScript on the client side, we need compatible libraries so that exchanged data are understood in the same way by both. We chose the Jackson library embedded in Spring Boot for the server side, and the Jackson-js library, an implementation of the Jackson library in JavaScript, for Angular.

As described previously, there is also a difference between serialization mechanisms based on attributes (ignoring their visibility) or based on public getters/setters. The Jackson library accesses attributes through public accessor methods. Again, because RMI and GWT-RPC give more freedom, application developers did not strictly implement all of the accessors. We decided that this was a violation of good programming rules and programmed it into our (custom made) linter so that developers should fix all violations before the migration occurs.

As mentioned previously, the data exchange classes of the original applications sometimes contained behavioral code. While it is not complex to automatically migrate the data exchange class hierarchy and fields from Java to TypeScript, these methods had to be ignored in the process due to the difficulty of transpilation. The development teams will need to reimplement them manually, and again we use our linter to identify all occurrences of such methods.

Pruning Data Exchange Objects

The last problem we had to deal with was the pruning of data exchange objects. This is necessary when the server sends too much data to the client, resulting in a noticeable wait while the client deserializes it.

To solve this problem and reduce the payload size, we use a solution inspired by GraphQL (https://graphql.org) and follow the Wish List pattern.[5, p. 335] The idea is to reduce the size of the data exchange objects returned by the server, by not serializing data that is ignored by the client. In practice, when the client performs a request to the server, it adds, as a parameter, the list of fields of interest in the returned objects. Note that this list is not shown in Listing 1 (line 14) for the sake of clarity. Then, on line 20, before returning it to the client part, the result of the method call is passed to a special function along with the list of fields of interest. This function discards the uninteresting fields so that they do not clutter the JSON data. Again, for the sake of clarity, this function call is not shown in the listing.

GraphQL itself is a comprehensive technology that would have replaced Spring controllers and required substantial changes to the applications' source code. Therefore, we chose the small *Jackson-AntPathFilter* project (https://github.com/Antibrumm/jackson-antpathfilter), which provides the same pruning functionality but makes fewer assumptions about the host code.

The applications we worked on are all "standard management applications." As such, a large part of the UI displays tables of data. The cases where pruning was necessary were for these tables, when the service returns a list of objects of which only a few fields are actually displayed. For each displayed table, we identify the associated service invocation (which collects the data to display) and we extract the fields that the table actually requires.

To do this, we used static analysis of the code, which was facilitated by the fact that the developers followed code style guidelines. First, we need to associate a displayed table with the service that provides its data as follows:

- The table widget is identified because it is a class that inherits from `AbstractTableWidget`.
- We look for the creation of instances of this class and the use of the `setModel(new ATableModel())` method, which associates a model with the widget.
- Table model classes inherit from an `AbstractTableModel` superclass that provides a `loadData`

method. We look for the call to this method in `ATableModel` since it usually takes as parameter the invocation of a service (which will return the required data).

This tells us which service returns the data that will be displayed in the table. We must now identify which fields are actually displayed so that we can prune the result of the service. To do this, we look at the constructor of the table widget, which contains calls to the `addColumn` method (see Listing 2). Each parameter of `addColumn` is an attribute of interest.

This solution is not completely reliable, as it may sometimes be difficult to track down the service invocation from the `loadData` call. When we are not able to follow a complex data flow from the call to `loadData`, we insert a "TODO" comment in the generated code next to this call. The application developers will have to go back to the migrated client, to resolve all these comments by manually tracking down the service invocation and inserting the list of fields of interest.

EVALUATION

We tested our solution on four real industrial applications. While we manually verified that the migrated applications do work as intended, they have not yet been accepted by their respective development teams.

Table 1 gives some data about the subject applications. They are management applications (financial, customers, human resources). Three of them are large, with hundreds of service classes (in the server part), and thousands of service methods (actual services). Prior to the migration, three of the applications used GWT-RPC and one used RMI.

To evaluate the usefulness of data exchange object pruning in terms of execution time and memory consumption, we analyzed two different web pages. The first contains a seven-column table widget. It displays

```
class FolderTableWidget
 extends AbstractTableWidget {

  public FolderTableWidget() {
   addColumn(Folder.class, "date");
   addColumn(Folder.class, "city.cityRef");
 }
}
```

LISTING 2. Initialization of a Table

three rows after the service invocation. The second one contains an eight-column table widget. It displays nine rows after the service invocation. For each web page, we evaluate three aspects: the size (in kilobytes) of the message returned by the server, the time (in milliseconds) the server spent executing the request and pruning the result, and the "user time" (in milliseconds) from before the call to `loadData` to after it. This is the time the user waits for the empty displayed page to load the data. It includes the call to the service and the time to deserialize the data exchange object.

Table 2 shows the results of this experiment. In the first case, the payload size is reduced by 40%, going from 64 kB to 39 kB. The server takes more time due to data pruning (24 ms instead of 18 ms), but the difference is negligible from a human perspective. The user time decreases from 1301 ms to 822 ms (about 0.5 s), which is a small difference for the end user.

In the second case, the improvement in object size is huge (98%), going from 4249 kB to 83 kB. The server time is also reduced, going from 204 ms to 92 ms (55%), probably because there is so much less data to serialize. The user time decreases from 12,126 ms to 1401 (89%). This last case is a concrete example of a service for which the end user would have to wait more than 10 s, which drops to 1.5 s after pruning (still noticeable, but more acceptable).

The overall inferior performance of RESTful HTTP compared to RPC technologies is causing some

TABLE 1. Four applications on which we tested our approach.

	KLOC	Service classes	Service methods	Data Ex. Classes	Legacy comm.
app1	1090	323	2991	6705	RMI
app2	2735	448	3355	4611	GWT-RPC
app3	1125	645	4474	5099	GWT-RPC
app4	322	63	534	479	GWT-RPC

TABLE 2. Time and memory consumption with and without object pruning (values averaged over several calls).

		Payload size (kB)	Server time (ms)	User time (ms)
Case 1	no pruning	64	18	1301
	w/ pruning	39	24	822
	Improvement	40%	–25%	37%
Case 2	no pruning	4249	204	12,126
	w/ pruning	83	92	1401
	Improvement	98%	55%	89%

companies to migrate back to RPC.[7] If performance becomes a significant issue in validating migrations, a more efficient JSON-compliant solution could be used at the expense of readability.[8]

In summary, we have succeeded in designing a semiautomated tool to migrate services from legacy RMI/GWT-RPC to a modern Spring HTTP API. Using data exchange object pruning, we also addressed the performance issue of (de)serialization.

In this article, we exposed a concrete problem of client–server communication migration. We presented an approach to migrate the client–server communication of applications and applied it to applications of Berger-Levrault.

Client–server migration is a complex and time-consuming process that requires careful planning and execution. To successfully migrate client–server communications, it is necessary to identify the technologies and protocols used by the existing system, as well as any potential roadblocks or compatibility issues. It is also important to have a clear understanding of the data exchanged between the client and server, as well as any dependencies or interactions between different components of the system.

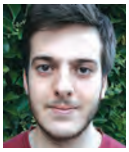
We propose to use a semiautomated tool to identify and update code that is specific to the legacy system, while leaving more general code intact. In this context, standardizing the source code according to coding conventions can make the migration process smoother and more efficient. This reduces the manual effort of discovering and mapping source widgets to their target counterparts, and it improves the maintainability of the code.

Finally, it is important to thoroughly test the migrated system to ensure that it works correctly and meets the needs of end users. We provide links at https://github.com/badetitou/Casino to several importers and generators that can help migrate applications and provide details about our approach.

REFERENCES

1. S. Bragagnolo, N. Anquetil, S. Ducasse, S. Abderrahmane, and M. Derras, "Analysing Microsoft access projects: Building a model in a partially observable domain," in *Proc. Int. Conf. Softw. Syst. Reuse (ICSR'20)*, Dec. 2020, pp. 152–169, doi: 10.1007/978-3-030-64694-3_10.
2. B. Verhaeghe et al., "From GWT to angular: An experiment report on migrating a legacy web application," *IEEE Softw.*, vol. 39, no. 4, pp. 76–83, Jul./Aug. 2022, doi: 10.1109/MS.2021.3101249.
3. Zirkelbach, A. Krause, and W. Hasselbring, "On the modernization of ExplorViz towards a microservice architecture," in *Proc. Workshops German Softw. Eng. Conf. (CEUR)*, 2018, pp. 1–4.
4. G. Hohpe and B. Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Englewood Cliffs, NJ, USA: Prentice Hall, 2004. [Online]. Available: https://books.google.com.au/books?id=dH9zp14-1KYC
5. O. Zimmermann, M. Stocker, D. Lübke, U. Zdun, and C. Pautasso, *Patterns for API Design: Simplifying Integration with Loosely Coupled Message Exchanges*. Reading, MA, USA: Addison-Wesley, 2022, p. 96.
6. B. Verhaeghe et al., "A hybrid architecture for the incremental migration of a web front-end," in *Proc. 17th Int. Conf. Softw. Technol. (ICSOFT)*, SciTePress, 2022, pp. 101–110.
7. Y. Lee and Y. Liu, "Using refactoring to migrate REST applications to gRPC," in *Proc. ACM Southeast Conf.*, New

York, NY, USA: Association for Computing Machinery, 2022, pp. 219–223, doi: 10.1145/3476883.3520220.
8. J. C. Viotti and M. Kinderkhedia, “Benchmarking JSON BinPack,” 2022, *arXiv:2211.12799*.



GABRIEL DARBORD is a Ph.D. student at the University of Lille, F-59000 Lille, France. Contact him at gabriel.darbord@inria.fr.

BENOÎT VERHAEGHE is a research engineer at Berger-Levrault, 69760 Limonest, France. Contact him at benoit.verhaeghe@berger-levrault.com.

ANNE ETIEN is a full professor at the University of Lille, F-59000 Lille, France. Contact her at anne.etien@inria.fr.

NICOLAS ANQUETIL is an assistant professor at the University of Lille, F-59000 Lille, France. Contact him at nicolas.anquetil@inria.fr.

ANAS SHATNAWI is a senior research engineer at Berger-Levrault, 34470 Pérols, France. Contact him at anas.shatnawi@berger-levrault.com.

ABDERRAHMANE SERIAI is a senior research and development engineer at Berger-Levrault, 34470 Pérols, France. Contact him at seriai.abdelrahmene@gmail.com.

MUSTAPHA DERRAS is a research director at Berger-Levrault, 31670 Labège, France. Contact him at mustapha.derras@berger-levrault.com.

ADVERTISER INFORMATION

Advertising Coordinator

Debbie Sims
Email: dsims@computer.org
Phone: +1 714-816-2138 | Fax: +1 714-821-4010

Advertising Sales Contacts

Mid-Atlantic US, Northeast, Europe, the Middle East and Africa:
Dawn Scoda
Email: dscoda@computer.org
Phone: +1 732-772-0160
Cell: +1 732-685-6068 | Fax: +1 732-772-0164

Southwest US, California:
Mike Hughes
Email: mikehughes@computer.org
Cell: +1 805-208-5882

Central US, Northwest US, Southeast US, Asia/Pacific:
Eric Kincaid
Email: e.kincaid@computer.org
Phone: +1 214-553-8513 | Fax: +1 888-886-8599
Cell: +1 214-673-3742

Midwest US:
Dave Jones
Email: djones@computer.org
Phone: +1 708-442-5633 | Fax: +1 888-886-8599
Cell: +1 708-624-9901

Jobs Board (West Coast and Asia), Classified Line Ads

Heather Buonadies
Email: hbuonadies@computer.org
Phone: +1 623-233-6575

Jobs Board (East Coast and Europe), SE Radio Podcast

Marie Thompson
Email: marie.thompson@computer.org
Phone: +1 714-813-5094

EDITOR: George Strawn, gostrawn@gmail.com

DEPARTMENT: MASTERMIND

This article originally appeared in IT Professional vol. 24, no. 6, 2022

Doing for Data What the Internet Did for Networking

George Strawn, *National Academy of Sciences, Washington DC, 20001, USA*

FAIR DIGITAL OBJECTS (FDOS)

THE acronym FAIR, (findable, accessible, interoperable, and reusable), refers to an international movement to effectively manage scientific data. The FAIR digital object (FDO) is an approach to implement FAIR data.[1,2]

FDOs seek to do for data what the Internet did for networks. That is, the Internet created a software network that interconnects existing hardware networks, and FDOs will create a software data system that makes diverse data objects interoperable. FDOs can encapsulate data, programs, work flows, and other digital artifacts. This article will focus on FDOs encapsulating data objects. Among other things, FDOs enable programmers to work with unknown data, as they do with known data.

In the current state of development, one can speak of FDOs as a general concept that might be implemented in different ways (Digital Object Architecture, Semantic Web, etc.) or as a specific implementation of the concepts. This article focuses on one approach that is currently under development. Moreover, for us techies, "what something is" is best described in the context of "how it's done." So both the "what" and the "how" of the particular implementation of FDOs are described in the following.

FDOs Are an Extension of the Internet

The second version of the Arpanet was based on the Transmission Control Protocol/Internet Protocol (TCP/IP). It was developed as specialty and local area networks were starting to interconnect computers. The goal of what became the Internet was to create a "network of networks" that would interconnect all of the computers on these networks. This was accomplished by utilizing the important computer science concept of creating a software system on top of a hardware system. In this case, creating a software network (the Internet) on top of the existing hardware networks.

The first requirement for the Internet was that each computer has a globally unique identifier, which was called its IP address. This requirement enables any two computers to identify each other to send and receive messages. The second requirement was that all the specialty networks needed to be interconnected, not each-to-each, but such that there was at least one path between any two networks. This interconnection was achieved by introducing special computers, called routers, each connected to two or more hardware networks. There are enough routers to create a path between any two Internet computers.

How Does the Internet Send and Receive Messages?

When an application on computer1 wants to send a message to that same application on computer2, it invokes its copy of TCP with the IP addresses of both computer1 and computer2. The message is then sent over computer1's specialty network to its connected router. The router consults its IP table, which tells it over which specialty network to next send the message, en route to computer2. After traversing a number of specialty networks, the message arrives at computer2 and its copy of TCP sends it to the named application.

Components of an FDO

This explanation of the Internet process is important for two reasons: first, since FDOs are the extension of the Internet, the Internet process is also a part of the FDO process; and second, FDOs have addresses and protocols like the Internet. Just as every Internet computer has globally unique address, so every FDO requires a globally unique address. It would be possible, but undesirable, to tie an FDO address to a computer address by extension of its files. But, by creating a unique address space for FDOs, a given FDO can be replicated on different computers, and FDOs become first-class objects,

1520-9202 © 2023 IEEE
Digital Object Identifier 10.1109/MITP.2022.3222712
Date of current version 11 January 2023.

which are entities that can be dynamically created, destroyed, passed to a function, returned as a value, and have all the rights of other first-class objects, such as network connected computers.

FDO addresses: there is an address space H, similar to the IP address space and each FDO has a persistent, globally unique address in H. Importantly, if this address space is administered by a separate, persistent organization, it would be immune to changes in other related technologies.[2]

FDO address protocol: an address resolution protocol that can both create a new address in H and resolve a given FDO address in H to the current "access logistics" of that FDO. While the address in H is persistent, the results of the resolution can change both in the short term, e.g., the FDO is moved from one repository to another, or the long term, e.g., a completely new set of technologies replace current repository technology. This level of indirection, essential for persistence and future proofing, is the primary reason to create an identifier to the FDO with editable resolution results.

FDO content: just as an Internet message contains a bit string pay load, an FDO contains a bit string pay load. An Internet pay load is "only" a bit string because the receiving application must know how to interpret it. An FDO, however, must include metadata to interpret the bit string.

Web Pages: Similar to FDOs, but Much Simpler

A web page (originally) is a document to be read by humans. It is stored in a mark-up language format (html) that a browser can process and display on a variety of computers. The hypertext transfer protocol (HTTP) is invoked when a user clicks on a hyperlink that points to a web page. HTTP then invokes TCP, which establishes a connection to the computer containing that page. The page is then downloaded (as a bit string) and displayed by the browser, which understands the html syntax. A similar but more complicated protocol, digital object interface protocol (DOIP), is invoked when an application wants to access an FDO.[2]

FDO Access Protocol DOIP: In this approach to FDOs, an application invokes DOIP to access an FDO. Unless access restrictions apply, any application can access any FDO. As stated previously, metadata is required to interpret its bit string.

What Makes Digital Objects FAIR?

As is now well known, FAIR is a data acronym for findable, accessible, interoperable, and reusable.[1] Findable and interoperable depend on technical solutions, whereas accessible and reusable depend on economic and legal matters. Other dimensions of data may also be required for "fairness" as well, but they would damage a great acronym! Since this article focuses on technical matters, only findable and interoperable will be considered.

Web data are the current poster child of find ability. That is, Google and other search engines have made web data searchable in a fraction of a second, and since so much textual data is on the web, "googling" has become an important activity for many of us. But even for textual data, find ability has more mountains to climb. An example of more complex find ability is described in the following. In addition, pictures, music, numerical data, software, and other digital artifacts are or need to be findable. New algorithms for generalized find ability will be required to fully enable the system of FDOs.

Automated data interoperability will be the killer app of the FDO system. Current estimates of the manual effort to make noninteroperable data interoperable are usually 80% of project time, obviously leaving only 20% for analysis and other uses of the data. By including all necessary characteristics of the data as FDO metadata, an initial FDO system goal should be to reduce the data preparation time from 80% to 20%. Advanced metadata techniques might ultimately achieve the goal of complete machine actionability of the data preparation phase.

Known Data Versus Unknown Data and Metadata

As a programmer creates a "normal" program, i.e., one with only local data, the syntax and semantics of the data are known. This applies to internal data structures, data files, and abstract data type (ADT) data. Thus, the programmer can tailor the program to the data characteristics. In the case of ADT data, the syntax is hidden, but operations/subroutines are provided to access and update that data.

Metadata components of an FDO: As a programmer creates a program that will utilize one or more FDOs, it is necessary to obtain the FDO data syntax and semantics, and other relevant characteristics from the metadata of the FDO. To begin with, the programmer might invoke DOIP with the address of the FDO and a request to return its type and the operations that are available to access it.

The metadata should have a fixed form, such as provided by the Semantic Web. If the metadata are in the RDF format, then the SPARQL search language is available for advanced processing.[3] An example of advanced searching is given next.

Advanced Metadata Processing

Consider an FDO that (recursively) contains other FDOs that consist of the titles and abstracts of biomedical research articles (e.g., Medline at NIH). If one constructs an FDO that consists of the key sentences of all the abstracts (average four per article). These key sentences explain what are the article's science results). If these sentences are then manipulated into RDF format and contain only words from a controlled vocabulary, the SPARQL search language can, among many other things, discover "new" science results that are spread across multiple articles (e.g., Semantic Medline at NIH). That is, unless the same scientist has read all those articles, it may not be realized that the result is at hand.

An example of such metadata processing is as follows. The question was asked whether two specific medical facts were related. Fact one is that aging men have less testosterone, and fact two is that aging men have more sleep problems. An SPARQL query easily discovered one article that showed that testosterone inhibits cortisone and a second article showed the cortisone disturbs sleep. This is a real example from NIH.[4]

One Computer, One Data Object

The Internet inspired Sun Microsystem's slogan that "the network is the computer." FDOs' goal is to make all digital objects interoperable, (as well as findable, accessible, and reusable, of course). So, poetically speaking, the Internet gave us a world with only one computer and FDOs will give us a world with only one data object. No one doubts that the Internet has changed the world. FDOs may very well do the same.

REFERENCES

1. [Online]. Available: https://www.go-fair.org/fair-principles/
2. [Online]. Available: https://www.dona.net/digitalobjectarchitecture
3. [Online]. Available: https://www.ontotext.com/knowledgehub/fundamentals/what-is-sparql/
4. [Online]. Available: https://lhncbc.nlm.nih.gov/ii/tools/SemRep_SemMedDB_SKR/SemMed.html

GEORGE O. STRAWN is the Director Emeritus of the NAS Board on Research Data and Information, National Academy of Sciences, Washington DC, 20001, USA. Contact him at gostrawn@gmail.com.

IEEE COMPUTER SOCIETY

www.computer.org

PURPOSE: Engaging professionals from all areas of computing, the IEEE Computer Society sets the standard for education and engagement that fuels global technological advancement. Through conferences, publications, and programs, IEEE CS empowers, guides, and shapes the future of its members, and the greater industry, enabling new opportunities to better serve our world.

OMBUDSMAN: Direct unresolved complaints to ombudsman@computer.org.

CHAPTERS: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

AVAILABLE INFORMATION: To check membership status, report an address change, or obtain more information on any of the following, email Customer Service at help@computer.org or call +1 714 821 8380 (international) or our toll-free number, +1 800 272 6657 (US):

- Membership applications
- Publications catalog
- Draft standards and order forms
- Technical committee list
- Technical committee application
- Chapter start-up procedures
- Student scholarship information
- Volunteer leaders/staff directory
- IEEE senior member grade application (requires 10 years practice and significant performance in five of those 10)

PUBLICATIONS AND ACTIVITIES

***Computer*:** The flagship publication of the IEEE Computer Society, *Computer*, publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications.

Periodicals: The Society publishes 12 magazines, 19 journals.

Conference Proceedings & Books: Conference Publishing Services publishes more than 275 titles every year.

Standards Working Groups: More than 150 groups produce IEEE standards used throughout the world.

Technical Communities: TCs provide professional interaction in more than 30 technical areas and directly influence computer engineering conferences and publications.

Conferences/Education: The society holds more than 215 conferences each year and sponsors many educational activities, including computing science accreditation.

Certifications: The society offers three software developer credentials.

COMPUTER SOCIETY OFFICES

Washington, D.C.:
2001 L St., Ste. 700,
Washington, D.C. 20036-4928;
Phone: +1 202 371 0101;
Fax: +1 202 728 9614;
Email: help@computer.org

Los Alamitos:
10662 Los Vaqueros Cir.,
Los Alamitos, CA 90720;
Phone: +1 714 821 8380;
Email: help@computer.org

MEMBERSHIP & PUBLICATION ORDERS
Phone: +1 800 272 6657; **Fax:** +1 714 816 2121;
Email: help@computer.org

EXECUTIVE COMMITTEE

President:	Jyotika Athavale
President-Elect:	Hironori Washizaki
Past President:	Nita Patel
First VP:	Grace A. Lewis
Second VP:	Nils Aschenbruck
Secretary:	Mrinal Karvir
Treasurer:	Darren Galpin
VP, Membership & Geographic Activities:	TBA
VP, Professional & Educational Activities:	Cyril Onwubiko
Interim VP, Publications:	Jaideep Vaidya
VP, Standards Activities:	Edward Au
VP, Technical & Conference Activities:	Terry Benzel
2023–2024 IEEE Division VIII Director:	Leila De Floriani
2024–2025 IEEE Division V Director:	Christina M. Schober
2024 IEEE Division V Director-Elect:	Cecilia Metra

BOARD OF GOVERNORS

Term Expiring 2024:
Saurabh Bagchi, Charles (Chuck) Hansen, Carlos E. Jimenez-Gomez, Daniel S. Katz, Shixia Liu, Cyril Onwubiko

Term Expiring 2025:
İlkay Altintaş, Mike Hinchey, Joaquim Jorge, Rick Kazman, Carolyn McGregor, Andrew Seely

Term Expiring 2026:
Megha Ben, Terry Benzel, Mrinal Karvir, Andreas Reinhardt, Deborah Silver, Yoshiko Yasuda

EXECUTIVE STAFF

Executive Director:	Melissa Russell
Director, Governance & Associate Executive Director:	Anne Marie Kelly
Director, Conference Operations:	Silvia Ceballos
Director, Information Technology & Services:	Sumit Kacker
Director, Marketing & Sales:	Michelle Tubb
Director, Membership Development:	Eric Berkowitz
Director, Periodicals & Special Projects:	Robin Baldwin

IEEE BOARD OF DIRECTORS

President & CEO:	Thomas M. Coughlin
President-Elect:	Kathleen A. Kramer
Director & Secretary:	Forrest (Don) Wright
Director & Treasurer:	Gerardo Barbosa
Past President:	Saifur Rahman
Director & VP, Educational Activities:	Rabab Ward
Director & VP, Publication Services & Products:	Sergio Benedetto
Director & VP, Member & Geographic Activities:	Antonio Luque
Director & President, Standards Association:	James E. Matthews III
Director & VP, Technical Activities:	John Verboncoeur
Director & President, IEEE-USA:	Timothy T. Lee

IEEE

revised 3 January 2024

EDITOR: Nir Kshetri, University of North Carolina at Greensboro, nbkshetr@uncg.edu

This article originally appeared in Computer vol. 56, no. 5, 2023

DEPARTMENT: COMPUTING'S ECONOMICS

"Frameworking" Carbon-Aware Computing Research

Phil Laplante, *The Pennsylvania State University*
Jeffrey Voas, *IEEE Fellow*

Computing economics is partially a function of energy availability and energy costs. We argue for the need to create a framework(s) for how to view the problem from energy considerations.

Computing economics is partially a function of energy availability and energy costs. Human factor issues, clearly, must also be considered. However, here, we argue for the need to create a framework(s) for how to view the problem solely from energy considerations.

The energy consumption costs of computing activities such as blockchain, data centers, supercomputing, and cryptographic applications have recently come to increased public and media attention. Nations, states, and cities are beginning to mandate energy-efficient computing in public works, and meanwhile, corporations, investors, and others are demanding the same from the private sector. We term any activity in response to these demands and mandates as *carbon-aware computing*, although we can also say "power-aware computing" or "energy-efficient computing" because the reduction of power consumption is an important problem regardless of the origin of the power.

The focus on carbon-aware computing presents challenges and opportunities. A successful response to these opportunities and challenges should theoretically result in a reduced carbon footprint of major computing activities and could lead to the discovery of new ways to optimize computing. However, these outcomes would need to be supported by a priori and posteriori research.

Carbon-aware computing includes hardware, software, and systems. Addressing the energy concerns of these three could provide government and industry with new sustainable solutions and business opportunities. For example, integration of wireless charging, energy harvesting, or "greener clouds" (cloud computing services that are competitively advantaged due to reduced energy consumption.)

Enablers for carbon-aware computing include retooling of existing optimization techniques for time and memory tradeoffs that can be adapted for energy efficiency. Algorithm efficiency is also important. Inhibitors to carbon-aware computing stem from the traditional computing paradigm, which evolved largely without regard to the costs of energy consumption (an exception is embedded computing systems). There is also a historical precedent for energy-aware computing in embedded systems that we shouldn't forget. Let's look at carbon-aware computing from a number of these perspectives.

CARBON-AWARE HARDWARE

From the dawn of computing through the mid-1980s, machines were severely limited in memory capacity and computation speed. The early devices were physically large in comparison to today and required substantial energy, so much so that cooling for data centers received significant research attention.

In the latter part of the early computing era, embedded systems, with limited memory capability (for example, the Space Shuttle Inertial Measurement computer program resided in 64 K of random-access memory), relied on a time-space tradeoff principle. This informal principle held that, to use less memory you had to increase computation time and vice versa. In those days, all kinds of tricks were used to

Digital Object Identifier 10.1109/MC.2023.3240482
Date of current version: 3 May 2023

 2469-7087/24 © 2024 IEEE

save time (like loop unrolling, which removed the overhead of counting the loop iterations) but at the expense of additional storage requirements. Many other approaches used in compiler optimization were employed, although these methods were specifically oriented toward real-time, embedded systems. The typical techniques included liberal use of arithmetic identities, constant folding, loop jamming, and alternative implementations such as lookup tables, and self-modifying code.[1] These techniques could be adapted for carbon-aware optimization by compilers.

In the early days of portable computing, researchers focused on battery-saving strategies. Various power conservation techniques were proposed that have now become commonplace, such as turning off the computer's screen during disuse, optimizing hard drive input–output, and slowing down the CPU with clocking schemes. Process migration over networks from high power consumption to more efficient computational devices can also be used.[2] Although most of these early hardware energy principles are still with us today, perhaps we need a new set of power-saving and "power-aware" principles for current computing paradigms.

CARBON-AWARE SOFTWARE ENGINEERING

Carbon-aware software engineering involves the design and development of software with a focus on reducing the carbon footprint. Related practices include reducing energy consumption of the software during development as well as its runtime energy consumption. It also includes considering the environmental impact of the hardware used to run the software. The techniques used in carbon-aware software engineering include energy profiling, power management, and use of renewable energy sources. (A small example here is the pixel colors on a smartphone screen; not all pixel colors drain the same amount of energy from the phone's battery.)

Carbon-aware software engineering takes us back to the old days of embedded computing with 64 K executable memory stores, self-modifying code, and space/time tradeoffs. Through the 1980s, software engineers had to be aware of the underlying hardware and target devices, components even to the transistor level, which is anathema to the concept of hardware-agnostic software engineering, computer architecture (hat tip to the late, great Fred Brooks).

But the new computing paradigm requires software engineers to rethink these principles. Fonseca et al.[3] proposed the following nine principles for energy-aware software engineering:

1. Public awareness is key for widespread adoption.
2. Incentives for software stakeholders.
3. Energy-aware software engineering should be a priority for every stakeholder.
4. Education and professional training should cover energy-aware software by default.
5. Broad adoption requires attention to usability.
6. Energy awareness should be engineered throughout the lifecycle.
7. Software quality should not come at the expense of energy awareness.
8. Energy awareness demands dynamic adaptability.
9. We value measures over beliefs (and reliable trends over precision).

All of these are self-explanatory and important. Incidentally, any of these principles could be adopted for cybersecurity.

To promote awareness, perhaps software engineers should adopt some sort of carbon-aware labeling system: green, yellow, orange, and red for energy usage for typical usage or exceptional profiles. Such a system could lead to carbon-aware compilers. Input to a "carbon-aware compiler" would include a description of the hardware configuration, a set to operational profiles and their probabilities, and the output would be (in addition to the usual outputs) a power consumption profile. Of course, this approach would necessitate an accurate description of the underlying hardware configuration and power requirements under an operational profile, accurate descriptions of these operational profiles and software trace executions, and accurate knowledge of the likelihood of these profiles.

CARBON-AWARE SYSTEMS

One of the primary concerns in carbon-aware systems engineering is, "Does the computing system, through action or inaction, cause another entity to use excessive energy?" The early approaches to carbon-aware system design involved process migration across carbon-aware computing ecosystems where processes were executed at the lowest (energy) cost destination (fluidified infrastructure). In the modern paradigm, we have roving cloud data centers on trucks moving from location to location to get the best utility rates. But similar work can be found much further back, for example, where assignment of processes and resources across large, distributed and heterogeneous computing systems based on multiple objectives (such as power saving) is presented.[4]

Bash et al.[5] propose the following principles for managing the energy consumption of computing hardware.

- Supply side:
  - A cradle-to-cradle analysis and design in minimizing the available energy in Joules required to extract, manufacture, mitigate waste, transport, operate, and reclaim components.
  - The use of local resources, such as local power generation or a water microgrid, to minimize the destruction of available energy as opposed to traditional systems in which available energy is lost in the transmission and distribution over long distances from a centralized plant.
  - Seek available energy in waste streams, for example, available energy in exhaust heat from turbines, or available energy in waste from a farm or municipality.
- Demand side:
  - Minimize the consumption of available energy by optimal provisioning of resources based on the needs of the user.
  - On-demand provisioning using flexible hardware building blocks, pervasive sensing, communications, knowledge discovery, and policy-based control.
  - Hardware-software co-design and management.

These principles can be used to identify and mitigate higher-order and emergent effects of carbon consumption.

METRICS, MONITORING, AND COSTS

IT incurs high energy requirements. Some service providers resort to extraordinary steps to manage these costs. For example, cloud-based providers may move from region to region simply based on the costs of electricity (and the associated regional electricity taxes). And it is well known that cryptocurrency and blockchain mining [and machine learning (ML)] seek cheaper energy. But what is not well known are the numbers, that is, what are the real electricity demands? We need IT electricity metrics to better plan, and unfortunately, surveys of existing literature might be all we have access to today.[6]

Home consumer power usage monitoring meters can monitor individual device usage. Whole-home sensor systems are also available. Constant monitoring and awareness is important, but it comes at a price. Monitoring devices for single appliances are relatively inexpensive (~US$50) and roughly US$300 for whole-home monitoring. But this is too pricey for universal use in the Third World unless grants are made. No one knows what the energy costs are for most of the software systems, and that's the problem.

Nothing is free. Assume that the underlying algorithms are optimized in terms of execution time, then using less energy means costs somewhere else, and we need to understand these costs. Is there a fundamental equation that prevails? For example,

$$T * S * A * E = \text{Constant}$$

where "T" equals the execution time, "S" is the memory space used, "A" represents accuracy, and "E" is the energy consumed, all for an assumed operational profile or weighted combination of profiles.

PROPOSED FRAMEWORK FOR CARBON-AWARE COMPUTING RESEARCH

To help organize and focus research efforts into carbon-aware computing, we recommend an Association for Computing Machinery computing classification systems-like taxonomy. For example,

- Hardware
  - energy profiling
  - data centers
  - energy utilization of clouds
  - energy usage in small computing environments
  - power costs of security
  - quantum computing energy costs
  - cryptocurrency and data mining energy utilization
  - alternative energy sources
  - energy-efficient design tradeoffs.
- Software
  - energy usage profiling
  - comparative energy costs of programming languages
  - AI/ML energy consumption
  - costs of cybersecurity
  - software/carbon-aware metrics and labeling
  - carbon-aware compilers
  - energy-efficient design tradeoffs.
- Systems
  - energy usage profiling
  - systems carbon-aware metrics
  - hardware/software co-design for energy optimization
  - ergonomic design to reduce carbon consumption
  - carbon-aware constructive cost estimators
  - ethical, societal, and environmental concerns.

We acknowledge that this taxonomy is incomplete; it is a starting point for discussion.

In summary, today's public increasingly demands greater energy efficiency for services and products, and this creates the need for a new set of principles and a research agenda with a framework that is actionable for computing professionals.

DISCLAIMER

The authors are completely responsible for the content in this message. The opinions expressed here are their own.

REFERENCES

1. P. A. Laplante and E. R. Dougherty, "Optimization techniques for real-time image processing," in *Proc. SPIE, Int. Soc. Opt. Eng.*, 1995, vol. 2421, pp. 116–122.
2. A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, "The remote processing framework for portable computer power saving," in *Proc. ACM Symp. Appl. Comput.*, 1999, pp. 365–372, doi: 10.1145/298151.298385.
3. A. Fonseca, R. Kazman, and P. Lago, "A manifesto for energy-aware software," *IEEE Softw.*, vol. 36, no. 6, pp. 79–82, Nov./Dec. 2019, doi: 10.1109/MS.2019.2924498.
4. T. J. Marlowe et al., "Multiple-goal objective functions for optimization of task assignment in complex computer systems," *Contr. Eng. Pract.*, vol. 4, no. 2, pp. 251–256, Feb. 1996, doi: 10.1016/0967-0661(95)00232-4.
5. C. Bash, N. Hogade, D. Milojicic, G. Rattihalli, and C. D. Patel, "Sustainability: Fundamentals-based approach to paying it forward," *Computer*, vol. 56, no. 1, pp. 125–132, Jan. 2023, doi: 10.1109/MC.2022.3219173.
6. C. Li and J. Voas, "Carbon 'Footprinting' IT," *Computer*, vol. 55, no. 11, pp. 12–14, Nov. 2022, doi: 10.1109/MC.2022.3197913.

PHIL LAPLANTE is a professor of software and systems engineering at The Pennsylvania State University, State College, PA 16801 USA, a Fellow of IEEE, and an associate editor in chief of *Computer*. Contact him at plaplante@psu.edu.

JEFFREY VOAS, Gaithersburg, MD 20899 USA, is the editor in chief of *Computer*. He is a Fellow of IEEE. Contact him at j.voas@ieee.org.

EDITOR: Doron Drusinsky, Naval Postgraduate School, ddrusins@nps.edu

DEPARTMENT: ALGORITHMS

This article originally appeared in Computer vol. 55, no. 1, 2022

On the High-Energy Consumption of Bitcoin Mining

Doron Drusinsky, *Naval Postgraduate School*

A well-known fact is that Bitcoin's blockchain maintenance using proof-of-work mining consumes a lot of energy. This article explains how this energy expense is inherent to Bitcoin's design whose purpose is to induce a truly distributed peer-to-peer currency ledger (the blockchain) that is nevertheless trustable.

The market capitalization for Bitcoin and other digital currencies is valued at hundreds of billions of dollars. Although the price of Bitcoin has fluctuated greatly over the years, its long-term phenomenal growth is obvious. Ever since its inception, entrepreneurs and speculators have realized they can make a profit by mining coins, a process that requires only computing hardware and the energy to power it. In the early days of Bitcoin, one could mine coins using a laptop and the rather low amount of energy it uses. The reward in those days was 50 bitcoins (in 2009) for each successful mining operation, albeit with a rather low dollar exchange rate per bitcoin at the time. Over the years, the number of bitcoins received by mining has halved every 210,000 blocks mined, down to 6.25 today, yet the exchange rate has grown from a few cents to close to US$30,000 in 2021. This trend has induced increased motivation for entrepreneurs to mine for bitcoins. The other side of the coin (no pun intended) is the fact that the energy required to mine bitcoins has increased significantly. The Digiconomist's Bitcoin Energy Consumption Index[3] estimates that the annualized total Bitcoin footprint is approximately 130 TW/h, comparable to the power consumption of Sweden; it produces 64 megatons of carbon dioxide (CO_2), equivalent to the carbon footprint of Serbia and Montenegro.

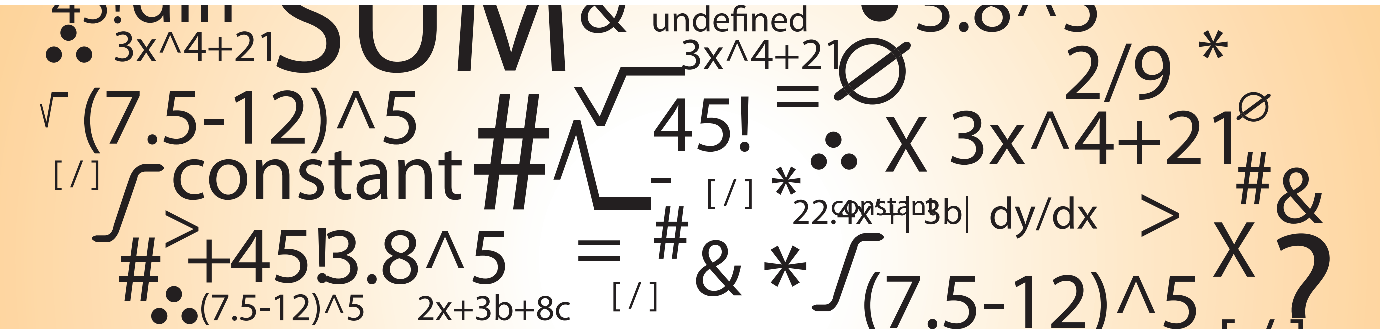
Digital Object Identifier 10.1109/MC.2021.3123781
Date of current version: 12 January 2022

As explained later in this article, a coin, or fraction thereof, is but a transaction in the Bitcoin ledger. Bitcoin ownership is proved using a public key (pk), for which only the valid owner has the corresponding secret private key. Hence, it is only the valid owner who can sign the transfer of such a coin to a new owner. There is one key difference between such a signature and the digital signature many of us routinely do as part of our daily business: the identity certificate or the lack thereof. The pk signatures used in a business setting typically have an accompanying certificate, signed by a trusted authority, that associates the pk with a person's or an organization's identity. The certificate therefore assures a bank, for example, as it verifies the digital signature of a customer claiming to be John Doe using his pk, that that pk is indeed John Doe's pk. In contrast, Bitcoin pks have no associated certificates, thereby providing certain, but not perfect, privacy.

BITCOIN BASICS

The Bitcoin blockchain, as detailed in the next section, is but a chain of collections of transactions, called *blocks*. The design goal of the blockchain is for each record, be it an individual transaction or a block of transactions, to become immutable once the Bitcoin community reaches a consensus that it is a valid record. Immutability is a result of two processes: the construction of the blockchain as a chain of hash codes, as described later in this section, and the process of repeatedly generating a

 2469-7087/24 © 2024 IEEE



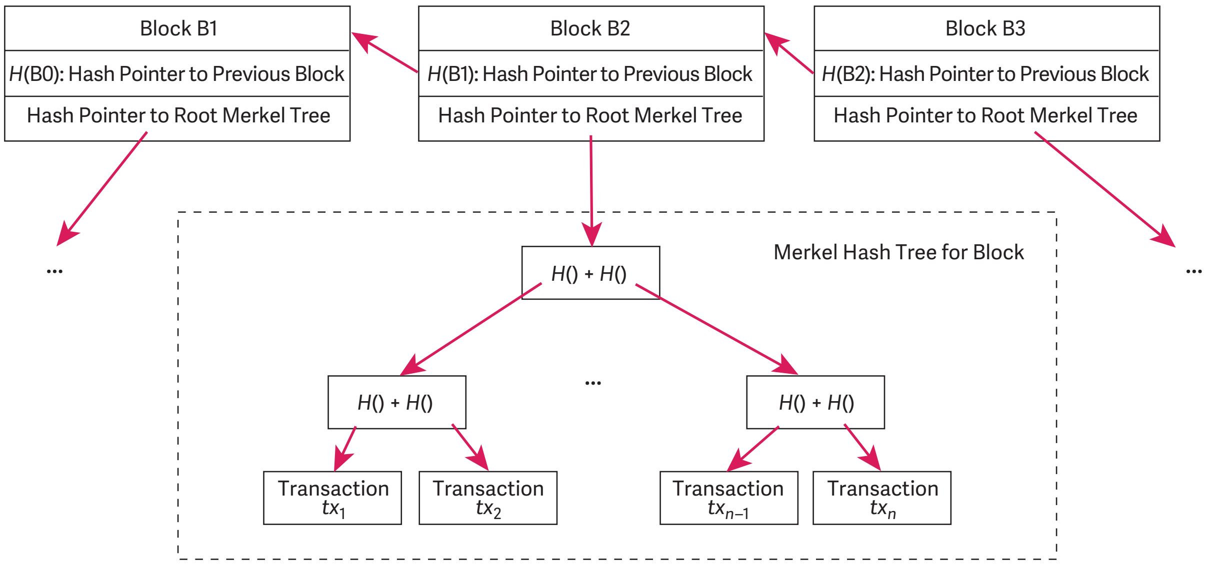


FIGURE 1. The Bitcoin blockchain. The top-most H() inside each record represents a hash pointer to the previous block in the blockchain, that is, it is the hash code of the previous block and also identifies ("points to") it. The bottom *H*() is the Merkel-tree root hash code; it is the hash code of the root of the block's Merkel tree and also points to it. Blocks are added every 10 min with blocks depicted on the right being ones added later in time.

networkwide consensus, using the energy expensive proof-of-work-based mining process described in the following sections.

Every record, be it an individual transaction or a block of transactions, has its own corresponding cryptographic hash code (SHA-256)[6] stored in a neighbor record, as illustrated in Figure 1. The transaction hash codes within a block are connected as a Merkel tree,[5] whereas one block's hash code is stored in the next block once such a block is created. As explained in the "The Bitcoin Blockchain—An Overview" section, this chain of hash codes effectively seals all the records in the blockchain because any change made to any record results in a hash-code mismatch that will be noticed by other nodes when they perform a periodic due-diligence record validation, as described in upcoming sections. Therefore, henceforth, the process of hashing a record (transaction or block) and storing that hash in another record is referred to as *sealing the record*.

In addition to sealing the record, any hash codes *H(R)* can also be used to locate the record *R* it hashed, that is, as a pointer to *R*; henceforth, *H(R)* is referred to as a *hash pointer* to *R*. Hash pointers are used as a part of the transaction-validation process described in the coming sections. The hash-pointer implementation details are too low level to be included in this article, yet one important relevant fact is that a Bitcoin cryptographic hash-code value is assumed to be a unique 256-bit value, that is, the likelihood of two records in the blockchain having the same hash code is negligible; therefore, such a hash code *H(R)* can also

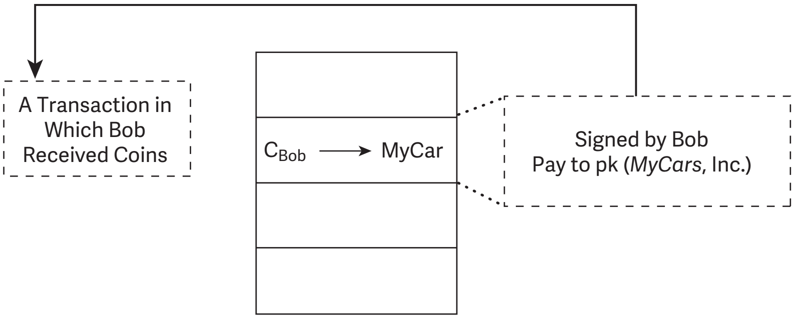


FIGURE 2. Transaction *tx*(*B*→*M*) transferring *c* coins from Bob to *MyCars*. The reference to the transaction in which Alice received coins is by means of hash pointer.

be used for a dictionary-like or database query that locates record *R*.

A cryptographic hash puzzle is a quest for a value that forces a hash-function result to fall into a certain range. Bitcoin's hash puzzle is defined for a given block of transactions *tx*1, ... , *txn*, as follows. Given a small target value Ta and the hash value *H*(prev) of the previous block, find a value nonce such that $H(\text{nonce} \;||\, H(\text{prev}) \;||\; tx1 \;||\; \dots \;||\; txn) < Ta$, where "||" is the concatenation operation. Because of cryptographic nature of *H*, such as SHA-256, the only known way for solving this problem is to perform repeated random guesses of nonce. Nevertheless, given a candidate nonce value, it is very easy to plug it in to $H(\text{nonce} \;||\; H(\text{prev}) \;||\; tx1 \;||\; \dots \;||\; txn)$ and check whether the inequality is satisfied.

Target value *Ta* is adjusted roughly every two weeks (2,016 blocks, to be precise); the adjusted value is calculated so that the hash puzzle is solved by some computer in the Bitcoin network roughly every 10 min.

THE BITCOIN BLOCKCHAIN—AN OVERVIEW

As with a conventional centralized financial institution, the Bitcoin ecosystem maintains a ledger of transactions, with a transaction being the ledger's basic element. A transaction is a record that details and subsequently proves the transfer of a certain number of coins among users. Consider an example in which Bob and *MyCars,* Inc., (henceforth *MyCars*) a car dealership, agree on a transaction wherein Bob buys a new car for a certain price. First, consider the transaction made using the conventional financial system, with the price being US$30,000. Bob needs to provide *MyCars* with a secured financial note, such as a money order made out to *MyCars,* guaranteeing that he owns the funds. Bob must also sign the note or an associated document attesting to his agreement to the transfer of those US$30,000 to *MyCars*.

Now suppose that the transaction is performed in bitcoins, with the car's price being *c* coins; this transaction is depicted in Figure 2 and denoted as *tx*(*B*→*M*). As Bob claims to own those c coins, he must have received them from someone, say Alice, in a transaction denoted as *tx*(*A*→*B*). Therefore, to prove that Bob owns the *c* coins he is using in tx(*B*→*M*), *tx*(*B* →*M*) includes a hash pointer to transaction *tx*(*A*→*B*). Bob then signs *tx*(*B*→*M*) using his private key and includes his pk in *tx*(*B*→*M*) for subsequent signature verification. Note that in practice, a transaction could have more than one input (rather than Bob alone) and more than one output (rather than *MyCars alone*). In fact, if *tx*(*A*→*B*) contains more than amount *c'>c,* then one such output will be Bob himself, where Bob pays Bob the change c'–c. Such transaction details and others are specified using the Bitcoin script language, which is how transactions are specified.[2] Clearly, this chain of coin transfers has to begin somewhere; those initial coin-creation transactions occurred when Bitcoin was created and also occur when a miner mines a block—mining being the process this article is all about.

The public face of a transaction such as *tx*(*A*→*B*) in Figure 2, is the pair of associated pks, called *addresses*, one for Alice, and one for Bob. Alice and Bob can have a large plurality of addresses; Bob can use one key pair to transfer coins to *X* and a different key pair to transfer coins to *Y*.

Each Bitcoin network node maintains a list of unspent transaction outputs (UTXO); transaction *tx*(*A*→*B*) is included in the UTXO before Bob buys his car, meaning that the amount detailed in that transaction is unspent; *tx*(*A*→*B*) is removed and *tx*(*B*→*M*) is added after the sale transaction is added to the blockchain, a process described later in this section. Given a hash pointer for a transaction like *tx*(*A*→*B*), it is but a key-value dictionary-like search used to locate that transaction in the UTXO.

An important clarification is that *tx*(*B*→*M*) is not a transaction between the person whose identity is Bob and the known entity *MyCars*; neither is it a transaction between a device owned by Bob to one owned by *MyCars*. Rather, this transaction transfers coins from a pseudonymous address for which Bob has the private key to an pseudonymous address for which *MyCars* has the private key; both Bob and *MyCars* can own a large plurality of such addresses. Cryptocurrency wallets are used by individuals and corporations to securely manage their collections of addresses and associated private keys.

A block in the blockchain of Figure 1 is a collection of transactions that occurred (and were sent to all nodes in the network) within a 10-min interval, using a process called *mining*, the energy-consuming process this article is all about. During mining, all the verification steps performed as a part of transaction tx(*B*→*M*), such as the verification of ownership of funds, are applied to all transactions in the block. As discussed in the next section, this verification process is done by every node in the Bitcoin network, without the help of a trusted, centralized financial institution.

Each node in the peer-to-peer Bitcoin ecosystem has a copy of the ledger blockchain, being the nodes' own view, or instance, of the blockchain. A consensus in Bitcoin is achieved when all honest nodes share the same view; the emphasis here is on honest nodes because dishonest nodes are obviously unpredictable. After each successful mining operation, that is, roughly every 10 min, the miner's blockchain is extended with a new block *B1* proposed by the miner; this is done by including a hash pointer to block *B0*, the last block of the blockchain. Note that at this point, the extended blockchain is but a proposal because a consensus has not yet been reached. The miner then transmits the new block to the rest of the network.[1] The other network nodes can decide, for any reason, to not accept that block, that is, not extend their blockchain instance with it and leave B_0 as the last node of the chain. When doing so, they can also try to deny the propagation of B_1 through the network by not transmitting it to their neighbors. However, if all honest nodes successfully verify the block, they will perform the Bitcoin protocol, add the block to their blockchain instance, and propagate it through the network. Ten minutes later, a newer block B_2 with a hash pointer to B_1, will be proposed to the network by some miner. As more blocks are verified and accepted into honest nodes' blockchain instances, older ones increasingly become part of a consensus blockchain. A common practice is that when B_6 is added to blockchain instances, then B_1 is considered being in the consensus blockchain.

Verification, both of individual transactions and of blocks of transactions, goes to the heart of trust that the Bitcoin blockchain ledger is secure. Specifically, one needs to verify that no transaction or block of transactions is modified, faked, denied, or duplicated. All honest nodes will perform the Bitcoin protocol and complete verification. We will discuss the verification operation in greater detail in a subsequent article.

When a newly mined block is being verified by network nodes, they make use of the Merkel tree structure (see Figure 1) for efficient verification of each transaction in the block. One verification aspect is seal verification, which verifies the seal of *tx*(*B*→*M*) or *tx*(*A*→*B*) in our running example; it is done as follows. The transaction is located using its hash pointer; its hash-code chain is then checked for modifications. As illustrated in Figure 1, that chain begins with the hash of the transaction itself (and its neighbor), goes up the tree, and then all the way to the right of the blockchain. In each transaction-verification step, the sum of the hash code of the tree node and that of its neighbor is compared to the hash pointer stored in the Merkel-tree node above it. This process is logarithmic in the size of the block because transactions within a block are organized as a (Merkel) tree. In contrast, if blocks were organized as linear chains of transactions resembling blockchain's chain of blocks, then the worst- and average-case cost would be linear in the number of transactions n_t contained in the block (n_t ~2,700). When applied to all transactions is the block, clearly $O(n_t * log_2(n_t))$ is much better than $O(nt^2)$.

BITCOIN—A DECENTRALIZED SYSTEM WITH CONSENSUS

The key objectives of most cryptocurrencies, including Bitcoin, are decentralization, security, and value. Decentralization was one of Bitcoin's original motivations, as evident from Satoshi Nakamoto's famous white paper[1] published on a cryptography mailing list, which describes

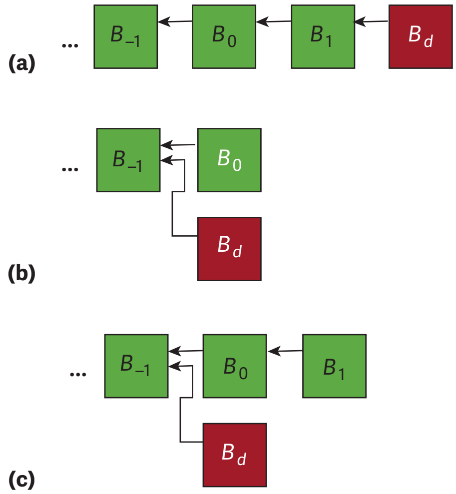


FIGURE 3. The blockchain on dishonest miner node *Md* after it added block B_d, which double spends *tx*(*A*→*B*) (which was already spent in B_0). The boxes represent blocks, and the arrows represent hash pointers. (a) The blockchain on node M_d after adding block B_d, which double spends *tx*(*A*→*B*) (already spent in B_0). (b) An alternative possibility for the blockchain on node M_d after adding block *Bd* which double spends *tx*(*A*→*B*). (c) The same as Figure 3(b), but whereas block B_1 was added to all nodes in the network before M_d attempts to add B_d to the blockchain.

digital currency that would allow secure, peer-to-peer transactions without the involvement of any centralized entity, whether that be the government, financial institution, a company, or some other organization.

A direct consequence of the decentralized, peer-to-peer architecture of the Bitcoin ecosystem is that the computers (called *nodes*) participating in the Bitcoin ecosystem do not have identities. This is because there is no centralized authority that assigns identities to nodes, and because nodes can join the ecosystem without the consent of such a centralized authority. Moreover, the peer-to-peer network itself is random, that is, a new node joining the network is assigned network neighbors randomly. Rather than having a single centralized ledger, the Bitcoin network has a large plurality of blockchain ledger instances, one in each participating node. Bitcoin's design objective is that all the honest nodes achieve implicit consensus about their respective ledger instances by following the same Bitcoin protocol.

Value is an obvious goal for any currency. Initially, the hope for Bitcoin was probably for it to gain value, whereas at present, with the large plurality of investors, there is probably more than one goal, with the minimal one being for Bitcoin to not lose value. Clearly also, to achieve and maintain value, such a currency must be widely accepted as being secure. Assuming that owners safeguard their private keys, then security means individual coins cannot be stolen from their owners or used by others while not owning them, and that transactions cannot be modified, faked, denied, or duplicated.

The network-node nodes that check all the aforementioned security properties are called *miner nodes*. Miners perform mining for profit, which is hereby considered an honest motivation. They all perform a *proof-of-work* operation which refers to attempting to solve a computationally expensive random hash puzzle. Lucky miners get to solve the hash puzzle; they then add their hash-puzzle solution to their newly proposed block, add that block to their blockchain instance (see Figure 1), and start propagating it to the rest of the network. Note that at this point, the newly proposed block is but a proposal. The distributed consensus process described in this section is designed so that this candidate block is accepted by the entire network only if the network agrees that it is an honestly verified block of transactions and that it includes all the latest transactions. The difficulty, however, is that absent a centralized investigative authority, miner nodes can be honest or dishonest. Striving toward an ecosystem with mostly honest nodes is at the core of the motivation behind the proof-of-work approach, as discussed below.

A dishonest node can affect the consensus blockchain only on the relatively rare occasions that it solves the hash puzzle because all honest nodes in the network check whether a newly proposed block contains a correctv hash-puzzle solution; the blocks that do not contain a solution will not be propagated through the network. In addition, when honest node N_h receives the newly created block it will easily notice if the proposed block B_e is not kosher in some other way, such as B_e's hash pointer is not pointing to the last node of N_h's blockchain instance, B_e is missing some transactions, B_e's Merkel tree is invalid, some transaction *Tr* inside B_e fails signature verification, and so on. Any such N_h that solves the hash puzzle next will propose valid block B_v instead of B_e.

Although changes in transaction details, such as coin amounts or recipient addresses, are easily detectable by honest nodes, more nuanced attacks by potentially dishonest miners are proposing a block with a double spending, and denying a valid transaction by omission, namely, by mining a block that does not contain that transaction.

Double-spending attacks can be done in more than one way. The most straightforward case is depicted in Figure 3(a) and represents a snapshot of the blockchain after dishonest miner M_d attempts to extend the blockchain with block B_d, which contains double-spending transaction $tx(B \rightarrow H)$; suppose it is double spending because it spends $tx(A \rightarrow B)$, which was used as a part of transaction $tx(B \rightarrow M)$ in block B_0. M_d then broadcasts block B_d to the network, but honest nodes will notice that $tx(B \rightarrow H)$ cannot use $tx(A \rightarrow B)$ because it is not in the UTXO (specifically, it has already been spent). They will then deny the propagation of B_d and not add it to their blockchain instances.

Miners in general and attackers in particular can add their mined block anywhere in the chain, that is, either to its end, as implied by Figure 1, or somewhere else. Hence, a double-spending (or omission) attack by an M_d might also look like that of Figure 3(b) or (c), depending on how soon after B_0 was created M_d attempted the attack, that is, whether B_1 was added to the chain before that event or not. Either way, M_d is trying to convince network nodes to verify block B_d and prefer it to B_0, B_1, and so on, which were added to those nodes' blockchain view at an earlier time. To that end, prevailing Bitcoin software applies the longest path rule, whereby honest nodes accept valid blocks that extend the longest path in the blockchain; therefore, M_d has the smaller chance of success in the case of Figure 3(c) than in the case of Figure 3(b). Either way, extending after B_{-1} does not conform to the longest path rule. Another form of attack is a variation of the situation presented in Figure 3(a), where the transactions contained in B_d are all valid, yet B_d is missing some transactions that posted in the last 10 min. Honest nodes, however, would notice that omission because they too, like B_d, received all transactions, and therefore treat B_d much like they would any invalid block.

It is expected for both honest and dishonest nodes to exist on an open, decentralized network. Dishonest nodes will presumably mine invalid blocks, add them to their blockchain instance, and send them to other nodes in the network, hoping they will add them to their own blockchain instance. Honest nodes will perform honest-block validation, detect that the proposed block is not valid, and will therefore not extend their blockchain copy with it. As an ad hoc rule, transactions are considered belonging to the consensus blockchain when their container block is extended six or more times, that is, roughly 1 h later.

Clearly, Bitcoin's challenge is to create an ecosystem of mostly honest pseudonymous miners. More specifically, the challenge is to 1) incentivize as many computer owners as possible to participate in this distributed process as nodes, thereby motivating a large plurality of unrelated nodes and 2) discourage nodes from creating crony nodes that might affect the outcome of the distributed agreement. After all, the more nodes that decline an invalid mined block, that is, a block with modifications (of transactions or entire blocks), double spending, or omissions, the lower the chance that the whole network will accept it in their blockchain view.

The Bitcoin proof-of-work process provides answers to these challenges. A node is chosen at random from all working nodes within the ecosystem. This random node selection is not done by a governing authority; rather, it is done by having all nodes try to solve a hash puzzle. As hash puzzles are randomized, the only way a node can solve the problem is by repeatedly tossing a coin and trying a new candidate solution. Being energy costly with almost no reward (rejected blocks are not rewarded even though a miner solved the hash puzzle), this process addresses the aforementioned challenge number 2). In contrast, the lucky nodes that solve the hash puzzle receive an incentive [thereby addressing the aforementioned challenge number 1)] in the form of 0.625 bitcoins (in 2021), delivered as a special Bitcoin transaction; miners will continue mining as long as this incentive offsets their expenses.

Note that most, if not all, fully operational nodes in the Bitcoin ecosystem participate in the mining proof-of-work operation, namely, they all try to solve the hash puzzle. Hash-puzzle solving power is measured using a metric called *hash rate*, which measures hashes computed per second (H/s). A personal laptop has a hash rate of a few millions of hashes per second; because the hash-puzzle target *Ta* has become so small over the years, the probability of solving it within 10 min on a personal laptop has become negligible. Therefore, serious miners must invest in expensive hardware to perform mining and hardware capable of computing a large amount of hash candidates, albeit with a high energy cost; specialized hardware can achieve close to 100 trillion hashes computed per second (TH/s). At present (August 2021), using 100-TH/s, 3,000-W hardware, a miner earns less than US$50 per day (at a Bitcoin price of approximately US$54,000); this miner's daily energy consumption is 72 kWh, inducing an expense of nearly US$9/day. The hardware's fixed cost is several thousand U.S. dollars. Miners most often use racks consisting of a plurality of such hardware devices, thereby increasing energy consumption even more.

As energy hungry as each mining operation is, multiply that by the number of mining nodes in the network (estimated to be more than one million nodes) to understand how the total electrical power consumption reached 135 TW/h in 2021. According to Digiconomist, a single Bitcoin transaction has an average carbon footprint of approximately 500 kg CO_2, the equivalent of roughly 90,000 h of watching YouTube.

DISCLAIMER

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. government. The U.S. government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright annotations thereon.

REFERENCES

1. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." Bitcoin. Accessed: Nov. 26, 2021. [Online]. Available: https://bitcoin.org/bitcoin.pdf
2. "Script," Bitcoin. Accessed: Nov. 26, 2021. [Online]. Available: https://en.bitcoin.it/wiki/Script
3. "Bitcoin Energy Consumption Index," Digieconomist. Accessed: Nov. 26, 2021. [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption
4. H. Auer and T. Perger, "Energy efficient route planning for electric vehicles with special consideration of the topography and battery lifetime," *Energy Efficiency*, vol. 13, no. 8, pp. 1705–1726, 2020, doi: 10.1007/s12053-020-09900-5.
5. "Merkle tree," Wikipedia. Accessed: Nov. 26, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Merkle_tree
6. National Institute of Standards and Technology, Gaithersburg, MD, USA. [Online]. Available: https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/examples/sha256.pdf

DORON DRUSINSKY is a professor in the Naval Postgraduate School's Department of Computer Science, Monterey, California, 93943, USA, and the chief science officer at Aerendir, Inc., Mountain View, California, 94040, USA. Contact him at ddrusins@nps.edu.

COMPUTING edge
WWW.COMPUTER.ORG/COMPUTINGEDGE

EDITOR: Schahram Dustdar, dustdar@dsg.tuwien.ac.at

DEPARTMENT: INTERNET OF THINGS, PEOPLE, AND PROCESSES

This article originally appeared in IEEE Internet Computing vol. 26, no. 6, 2022

Elastic Data Analytics for the Cloud-to-Things Continuum

Sergio Laso, *Global Process and Product Improvement, 10160, Cáceres, Spain*
Javier Berrocal, *University of Extremadura, 06006 Badajoz, Spain*
Pablo Fernandez and José María García, *Universidad de Sevilla, 41004 Sevilla, Spain*
Jose Garcia-Alonso and Juan M. Murillo, *University of Extremadura, 06006 Badajoz, Spain*
Antonio Ruiz-Cortés, *Universidad de Sevilla, 41004 Sevilla, Spain*
Schahram Dustdar, *TU Wien, 1040 Vienna, Austria*

The massive deployment of Internet-connected devices has led to an increase in the collection of data that are then used by companies to improve their decisionmaking processes. This growing trend demands more and more cloud and communications infrastructure. The limited resources, the need for sharing them, and the fact that many consumers are interested in the same data, call for an efficient management of the available resources. The cloud-to-things continuum can be used to execute different analytics closer to the data source so that infrastructure consumption and data circulation can be optimized. In this article, different dimensions for achieving elastic analytics and a framework for dynamically modifying their behavior are proposed.

Over the last few years, there has been a massive deployment of Internet of Things (IoT) devices, from heart rate sensors to location monitoring devices. This deployment has been especially driven by IoT applications that facilitate everyday tasks for every individual and organization. These tasks cover a wide range of use cases and applications, from managing an individual's personal health to understanding the flow of people and movement patterns in a city. This massive deployment of IoT devices has also led to the creation of networks of smart devices. This growing use of IoT devices is putting stress on current infrastructures in different dimensions.

On the one hand, it is putting stress on the amount of devices deployed. As IoT applications become more complex, they require combining devices from different domains for offering more useful functionalities. So far, new IoT applications and functionalities require the deployment of new devices. However, in this growing trend, the deployment of new IoT devices for sensing similar data is becoming unfeasible and unsustainable. For instance, a city that monitors the movement patterns of its citizens may also want to combine the location information with the heart rate in order to know what kind of activity citizens tend to do in each area of the city and, thus, provide more useful services and better plan their investments. For that purpose, instead of asking citizens to wear new heart rate monitors, it would be more feasible to ask them to share the data obtained by the monitors they already wear for other apps. Thus, in coming years, we will see how IoT applications will share large networks of devices already deployed.

On the other hand, it is putting stress on the infrastructure for data circulation and processing. Data, especially those coming from IoT devices, have become a strategic asset because of their availability to continuously monitor the environment without human intervention. IoT device networks are enablers of a new data economy in which more and more data are being exchanged not only within but also amongst

1089-7801 © 2022 IEEE
Digital Object Identifier 10.1109/MIC.2021.3138153
Date of current version 23 December 2022.

companies.[1] This fact boosts the massive deployment of IoT device networks. However, it also entails an increase in the circulation of information and the need for its processing and, consequently, a management model to support the optimal network usage. Such requirements will be even harder when different information systems require data from the same devices, for different needs and with different qualities. For example, a system to monitor an individual's heart rate during his/her activities may require very high information freshness; however, for a municipality to plan its services, such information does not have to be very fresh, but it has to be obtained from as many devices as possible.

In order to reduce the network overhead and improve the quality of service (QoS), IoT applications already attempt to take advantage of the cloud-to-things continuum to deploy services closer to information providers and consumers. However, such applications are still isolated systems that do not favor the sharing of data or analytics streaming. Thus, a sustainable data economy in this market[2] poses many challenges, among which we can count the following.

- First, the shared use of IoT devices so that data they are sensing can be used by different applications minimizing the drain on resources.
- Second, the optimization of the use of the cloud-to-things continuum infrastructure by enabling optimized deployment so IoT applications interested in the same data streaming and analytics can be deployed as close to the data as possible and together in the same nodes of this continuum.
- Third, elimination of duplicate streams by enabling applications interested in the same data or analytical streams to use the same datum.

In this article, we address these challenges by using elastic IoT data analytics whose behavior can be dynamically modified according to the quality requirements defined by each IoT system and the available resources of the cloud-to-thing infrastructure. For defining and deploying these analytics, we also provide a framework with an orchestrator managing the elasticity. This orchestrator evaluates the state of the infrastructure, the other analytics in progress, and the QoS required by each analytic. As a result, it reconfigures all the analytics to maximize the quality provided by each of them without exhausting the infrastructure nor the IoT devices.

In the following, we first discuss the different dimensions that should be taken into account for building elastic data analytics. Then, we present a framework for achieving elastic data analytics considering some of the defined dimensions. Finally, we present some conclusions and discussion about the elasticity required by IoT applications.

ENABLING ELASTIC DATA ANALYTICS

Data analytics allow companies to process large volumes of information in order to: find trends, eliminate unnecessary information, aggregate information, etc. In addition, they can enable efficient information management, as they may reduce the overhead of the transferred data. However, they may also require data storage and processing capabilities from the cloud-to-things continuum nodes. These analytics must be defined by each data consumer, since the results must be adapted to their requirements. Nevertheless, the circulation of information should be controlled to increase the efficiency in the use of both the system and the data.

Let us define a smart city scenario to better show the need of elastic data analytics. In order to maintain the simplicity, this case study is focused only on the citizens' location. Different institutions can analyze this information to identify their movement patterns, but with different goals. For instance, a healthcare system can use the real-time movements of the citizens to identify crowded areas and reduce the risk of infection for COVID-19. Instead, the city council may require long-term movement patterns to study future infrastructure investments (accessibility, bus lines, etc.).

The execution of these analytics impacts on several highly related dimensions, as Figure 1(a) shows, that must be controlled in a sustainable computing and information environment: data quality, resource consumption and cost.

First, one of the most important aspects for data consumers is the quality of the information. This quality must be appropriate for the specific functionalities they want to provide. While there are different properties to measure data quality, for information coming from IoT devices two properties are particularly relevant.[3]

1) *Accuracy:* It is the level to which data represent the real-world scenario. In this sense, in a network of IoT devices, the higher the number of devices involved in the data analytic the better they will represent the real world.
2) *Freshness* or the timeliness of the data: For some applications, the data have no value if it is

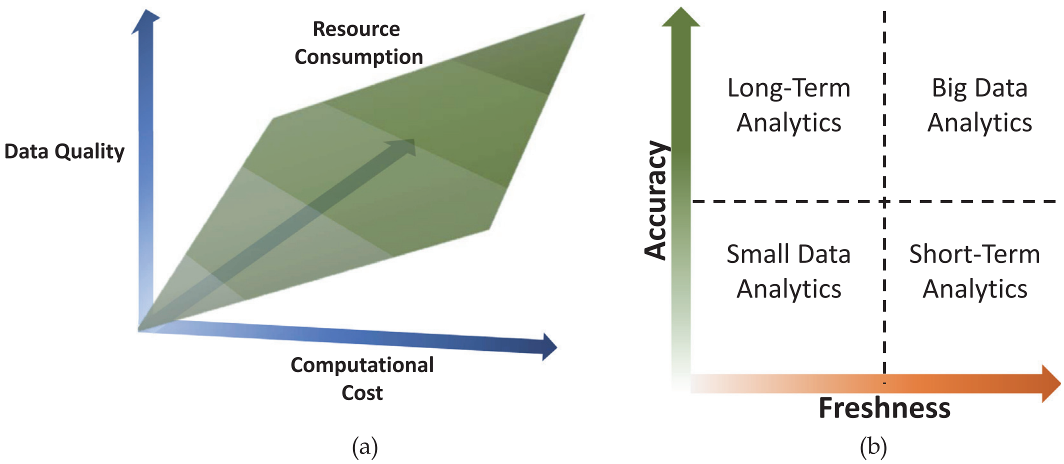


FIGURE 1. (a) Dimensions impacting the analytics. (b) Analytics depending on the data quality.

not available at the right moment. Freshness indicates the frequency at which IoT devices have to provide information for the analytic to provide value to the consumer.

Combining both dimensions, as Figure 1(b) shows, different types of analytics can be executed. When the accuracy and the freshness are low, small data analytics are executed in order to have a limited quantity of highly granular data that usually provide valuable information for the system. As the freshness and the accuracy increase, bigger data analytics are executed focusing on processing large volumes of information for business decisions. Instead, if only the accuracy increases, long-term data analytics are executed for predictive decision-making processes. In addition, if mainly the freshness is important for achieving an acceptable data quality, short-term data analytics are usually executed for developing reactive processes. For instance, for our case study, the healthcare system would require short-term analytics while the city council would require long-term analytics.

Second, this type of analytics has a direct impact on the resource consumption of the nodes in the cloud-to-things continuum. A greater accuracy means that more IoT devices will be involved in sending information, consuming their resources, and increasing the number of information flows. Edge and fog nodes can be used to distributedly process such analytics, reducing and aggregating the information flows. However, the greater the dispersion, the greater the number of nodes involved. Therefore, a high accuracy usually leads to a greater distribution of the resource consumption over the entire network. For our case study, the city council may require to use different nodes to cover the whole city, storing and aggregating the movement patterns whilst the healthcare system would only require a sample of some areas.

Finally, in addition to the data cost (which may depend on each data producer), its processing also entails some infrastructure costs associated with the use of edge and fog nodes.[4] Thus, the greater the freshness and frequency at which the information must be processed, the greater the need for processing this information in the fog or edge nodes, and the higher the infrastructure cost of the analytics. This cost may be limited to a small number of fog or edge nodes if the accuracy is low or to a larger one as the accuracy increases. For instance, for the healthcare analytics only the nodes involved in crowded areas would incur a higher cost. Furthermore, in a leveraged scenario with an open market of sensors,[5] this cost could also be dynamic depending on the fluctuations of the market.

In environments where multiple IoT applications are consuming similar information and using the same computing resources, data analytics should be able to have an elastic behavior.[6] They should be able to increase or decrease the provided quality depending on the available resources (and within the consumer's requirements). In addition, a framework would be needed to efficiently manage the distribution of all the requested analytics on the cloud-to-things continuum. For each analytic, this framework should identify in which fog or edge nodes it should be deployed to meet the accuracy and freshness requirements, without overloading the infrastructure. Likewise, when a new analytic is requested, the framework should use the defined elasticity to reconfigure the already existing analytics in order to achieve an efficient use of the

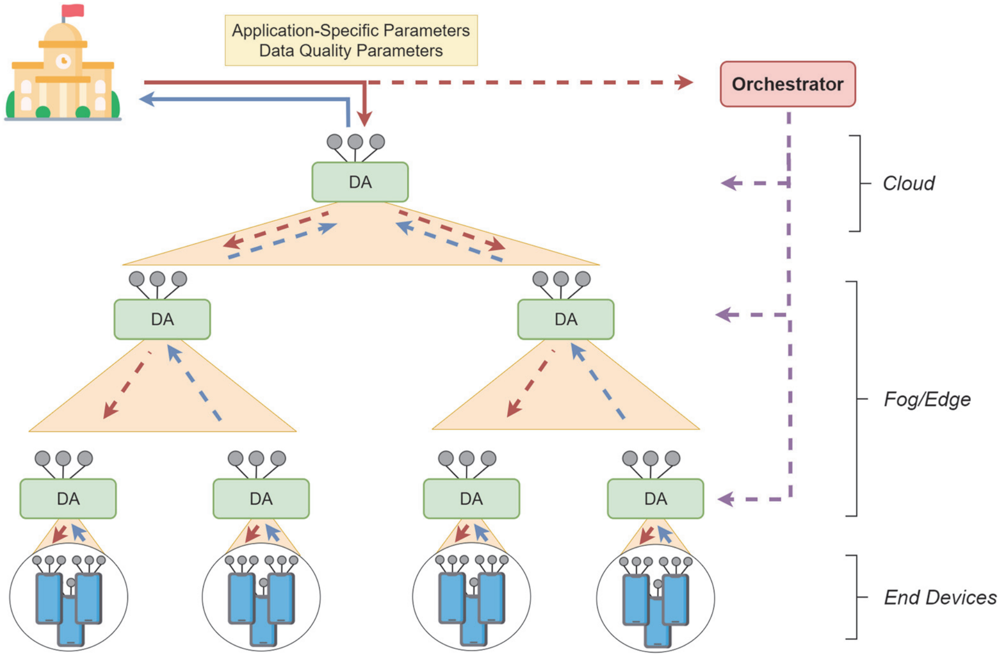


FIGURE 2. Architecture of the elastic data analytic framework.

computing resources, while maximizing the provided quality. Moreover, this distribution should also consider if several analytics should be deployed together, because they can share the same sensed data and, thus, increase the general efficiency of the system.

REALIZATION AND APPLICATION OF THE FRAMEWORK

To achieve elastic data analytics, we propose a framework orchestrating the analytics required by different data consumers and leveraging the cloud-to-things continuum to distribute them depending on the available resources and the required quality. The source code of the framework[a] for the presented case study and a video[b] showing the achieved elasticity are publicly available.

Figure 2 shows an example of the infrastructure that can be used for our smart city case study. It shows the different layers of the cloud-to-things continuum. Note that we limited the number of layers to improve readability. Data consumers (healthcare system and city council, for our case study) can request different analytics through the entry point to the infrastructure, the cloud. The deepest layer is composed by the end devices sensing and sending information to the cloud. This information goes through the fog and edge nodes on its way to the cloud where it is provided to data consumers.

Data analytics can be requested with different parameters. First, the application-specific parameters (for instance, the area to monitor in the smart city) and, second, the data quality parameters (accuracy and freshness). For the current implementation, the accuracy and the freshness can take different values from a range (low, medium, and high). The accuracy relates to the ratio of end devices involved, and the freshness to the frequency at which the information is obtained.

Figure 2 also defines the most important components of the proposed framework. First, cloud, fog, and edge nodes are composed by a data aggregator (DA) component. This component processes the deployed analytics. To that end, it requests the required information to the lower nodes depending on the data quality (mainly the freshness) required by the most demanding analytic, caches the obtained data to be reused by the other analytics, and processes it. In addition, the obtained results are also cached and always available, waiting for the higher levels to request them.

In addition, an elasticity orchestrator is proposed to balance the load of the whole infrastructure, modify the behavior of the elastic analytics depending on the previously defined dimensions, and to provide the best QoS to all parties. When a new analytic is required, it is analyzed by the orchestrator to evaluate the desired quality

[a][Online]. Available: https://doi.org/10.5281/zenodo.5793214
[b][Online]. Available: https://doi.org/10.5281/zenodo.5793189

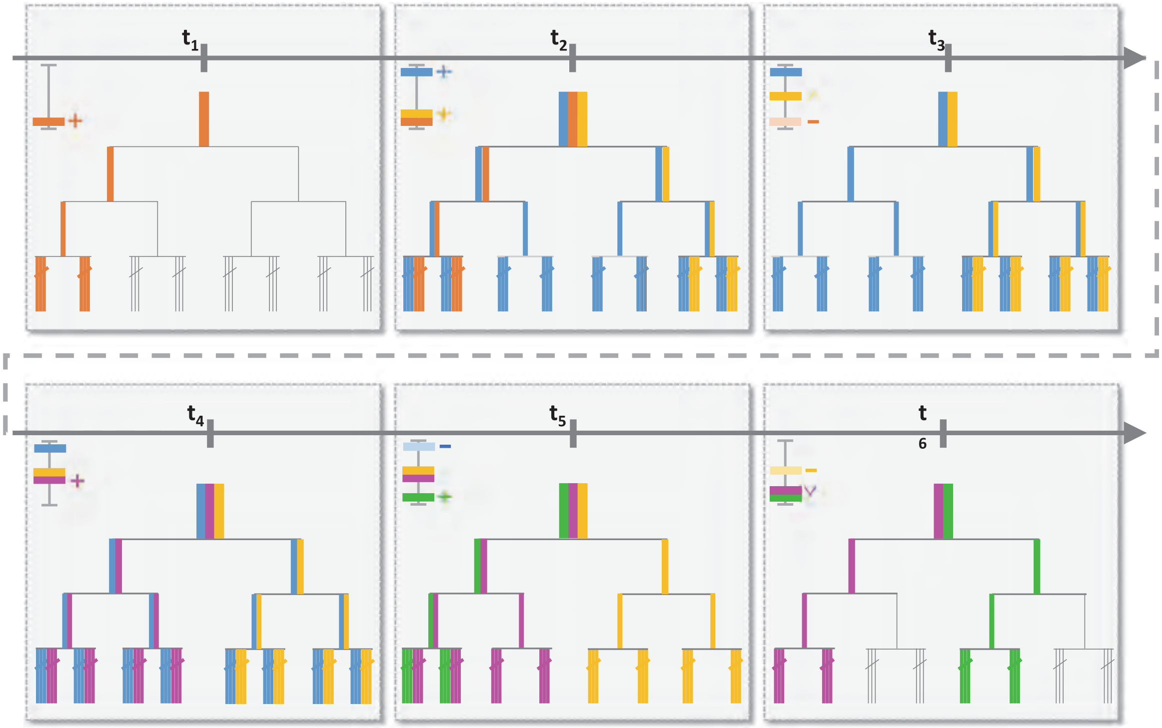


FIGURE 3. Elasticity dynamics.

(accuracy and freshness), the analytics already deployed in the architecture, and the workload of the different nodes. If there are enough resources, it directly deploys it providing the highest possible quality. If there are not enough resources, it checks if the analytics already deployed can be reconfigured to make room for the new one. This reconfiguration can be provided by redeploying existing analytics on other nodes to have a more efficient distribution or, if possible due to the quality defined by consumers, reducing the accuracy and freshness of some analytics to free up some resources.

Finally, all the DAs have the same communication interface (API). Thus, the infrastructure can be easily increased or reduced without heavily impacting the orchestrator. This also facilitates the work of developers and operators, avoiding having to create *ad hoc* solutions.

Figure 3 shows a working example of the provided elasticity. The example consists of a timeline with six

Analytic 1: Healthcare system, identification of crowded areas

Analytic 2: City council, study of movement patterns

FIGURE 4. Analytics obtained for the smart city case study.

DATA MANAGEMENT IN THE CLOUD-TO-THINGS CONTINUUM

Cloud computing has not only been a revolution at the business level to reduce IT costs. It has also fostered the massive deployment of new Internet-connected devices, with reduced computing resources but with huge capabilities for sensing and interacting with the environment. This has led to a steady increase in the data flows between these connected devices and the cloud. Paradigms such as fog and edge computing have brought the cloud environments, and hence the information processing, closer to the data sources. In fog computing, the processing is done on servers residing between the cloud and end devices, while edge computing focuses on processing the information at the edge of the network (gateways and end devices). By processing the information closer to the data source, the infrastructure overhead and the QoS can be improved.[1]

These paradigms allow the generation of distributed clouds making use of the cloud-to-thing continuum for the deployment of services throughout the entire infrastructure.[2] These services, some of them focused on data processing, can dynamically migrate from one layer to another depending on the volume of information to be processed, the workload of the nodes, or the QoS desired. Different technologies even orchestrate these distributed services to execute complex workflows.[3]

The incorporation of intelligence to the cloud-to-thing continuum is allowing researchers to take a further step in the processing of these huge data flows in order to get predictions and automate the decision-making process. However, fog and edge nodes have limited computing resources, hindering the execution of common artificial intelligence techniques. New techniques are being defined for reducing the rigidity of these mechanisms, allowing their distribution along the cloud-to-thing infrastructure.[4]

Finally, the cloud-to-thing continuum should not only manage the distribution of resources for a single IoT application but also support the integration of various applications, even from different domains. However, the different data semantics, requirements, and qualities of the different consumers complicate the interoperability and turn the collaboration across multiple vertical systems into a challenging task. New architectures are being defined to manage the infrastructure resources and share services between different systems.[5] Nevertheless, an elastic management of the information flows in the cloud-to-thing continuum is needed. This elasticity must consider the limited infrastructure resources, the interactions between applications, the data that can be shared between them, and the quality required.

REFERENCES

1. T. Lynn, J. G. Mooney, B. Lee, and P. T. Endo, *The Cloud-to-Thing Continuum: Opportunities and Challenges in Cloud, Fog, and Edge Computing*. Berlin, Germany: Springer, 2020, doi: 10.1007/978-3-030-41110-7.
2. J. L. Herrera, J. Galán-Jiménez, J. Berrocal, and J. M. Murillo, "Optimizing the response time in SDN-fog environments for time-strict IoT applications," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 17172–17185, Dec. 2021, doi: 10.1109/JIOT.2021.3077992.
3. M. Villari, M. Fazio, S. Dustdar, O. Rana, D. N. Jha, and R. Ranjan, "Osmosis: The osmotic computing platform for microelements in the cloud, edge, and Internet of Things," *Computer*, vol. 52, no. 8, pp. 14–26, 2019, doi: 10.1109/MC.2018.2888767.
4. D. R. Torres, C. Martín, B. Rubio, M. Díaz, "An open source framework based on Kafka-ML for DDNN inference over the cloud-to-things continuum," *J. Syst. Archit.*, vol. 118, 2021, Art. no. 102214, doi: 10.1016/j.sysarc.2021.102214.
5. N. Chen, Y. Yang, J. Li, and T. Zhang, "A fog-based service enablement architecture for cross-domain IoT applications," in *Proc. IEEE Fog World Congr.*, 2017, pp. 1–6, doi: 10.1109/FWC.2017.8368533.

instants (*t*). At each instant, an action is triggered that has an impact on the infrastructure. At the top left of each instant, a small legend appears with a vertical bar. This legend indicates for each analytic (each colored horizontal bar) if it is inserted/removed from the system (+/−) and, depending on its height in the vertical bar, the accuracy that it will provide. Note that we only represent the accuracy dimension to improve readability. At *t1*, the orange analytic is requested, requiring low accuracy. The orchestrator, therefore, assigns it to only a part of the infrastructure. At *t2*, two new analytics (yellow and blue) are inserted, with low and high accuracy, respectively; the orchestrator deploys the blue analytic throughout the whole infrastructure while the yellow one is deployed only in the less loaded part. At *t3,* the orange analytics ends and, because there are enough resources, the orchestrator upgrades the yellow one to a medium accuracy. At *t4,* the purple analytic is inserted and the orchestrator again assigns it the lowest loaded part of the infrastructure. At *t5*, the blue analytic ends and a new one with a low accuracy is deployed. Finally, at *t6*, the yellow analytics ends and the rest are reorganized to better distribute the resource consumption.

Finally, to show how the elasticity may affect the results obtained, Figure 4 shows two different analytics for the smart city case study. For each analytic, the data quality parameters (freshness and accuracy) are detailed. The results are the citizens' locations, which are shown with heatmaps for an easier decision-making process. The first analytic is focused on quickly identifying crowded areas for the healthcare system. Therefore, it has been requested with low accuracy and high freshness. Consequently, the orchestrator has only ordered the execution of the analytics to a part of the infrastructure and fewer citizens are involved but with real-time data. On the other hand, the second analytic would be interesting for the city council to study future investments. In this case, it has been requested with high accuracy and low freshness. Therefore, the orchestrator assigns the entire infrastructure to execute them and, therefore, gets the information from all the available citizens in the area.

CONCLUSION

The deployment of a swarm of sensors and Internet-connected devices is giving rise to a new data-driven economy. We envision an environment in which both the data to be consumed and the existing resources must be efficiently managed. In this article, we propose elastic data analytics, and we define the different dimensions affected by them. We also outline a framework to manage the dynamicity of these analytics in real time depending on the context.

Nevertheless, there are still some open challenges to be addressed to fully implement elastic data analytics. First, different policies should be generated for changing the management of elasticity depending also on the cost. For instance, trying to maximize the available computing resources, or minimize the costs to foster a sustainable economy. Second, more dimensions should be analyzed to identify how the elasticity affects them, such as the cost of the data. Currently, we work on including artificial intelligence to define self-managed analytics.

ACKNOWLEDGMENTS

This work was supported in part by the DIN2020-011586 Grant, funded by MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU/PRTR," in part by the Projects RTI2018-094591-B-I00, RTI2018-101204-B-C21 (MCIU/AEI/FEDER, UE) and 0499_4IE_PLUS_4_E (Interreg V-A España-Portugal 2014-2020), in part by the Department of Economy, Science and Digital Agenda of the Government of Extremadura under Grants GR18112, IB18030 in part by Junta de Andalucía under projects APOLO (US-1264651) and EKIPMENT-PLUS (P18-FR-2895), in part by the European Regional Development Fund, and in part by the RCIS Network under Grant TIN2016-81978-REDT.

REFERENCES

1. A. Opher, A. Chou, A. Onda, and K. Sounderrajan, *The Rise of the Data Economy: Driving Value Through Internet of Things Data Monetization*. New York, NY, USA: IBM Corporation: Somers, 2016.
2. F. Xhafa, B. Kilic, and P. Krause, "Evaluation of IoT stream processing at edge computing layer for semantic data enrichment," *Future Gener. Comput. Syst.*, vol. 105, pp. 730–736, 2020, doi: 10.1016/j.future.2019.12.031.
3. J. Medd, "Data quality fundamentals for data and analytics technical professionals," Gartner, Aug. 17, 2020. [Online]. Available: https://www.gartner.com/en/documents/3989182/data-quality-fundamentals-for-data-and-analytics-technic
4. A. Brogi, S. Forti, and A. Ibrahim, "Deploying fog applications: How much does it cost, by the way?," *Small*, vol. 1, no. 2, 2018, Art. no. 20, doi: 10.5220/0006676100680077.
5. J. M. García, P. Fernandez, A. Ruiz-Cortés, S. Dustdar, and M. Toro, "Edge and cloud pricing for the sharing economy," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 78–84, Mar./Apr. 2017, doi: 10.1109/MIC.2017.24.
6. S. Dustdar, P. Fernandez, J. M. García, and A. Ruiz-Cortés, "Elastic smart contracts in blockchains," *IEEE CAA J. Autom. Sinica*, vol. 8, no. 12, pp. 1901–1912, Dec. 2021, doi: 10.1109/JAS.2021.1004222.

SERGIO LASO is currently with Global Process and Product Improvement, Cáceres, Spain, where he is working toward the industrial Ph.D. degree in computer science. Contact him at slasom@unex.es.

JAVIER BERROCAL is currently an associate professor with the Department of Informatics and Telematics System Engineering, University of Extremadura, Badajoz, Spain. Contact him at jberolm@unex.es.

PABLO FERNANDEZ is currently an associate professor with the Applied Software Engineering Group, Smart Computer Systems Research and Engineering Lab, Research Institute of Computer Engineering, University of Sevilla, Seville, Spain. Contact him at pablofm@us.es.

JOSÉ MARÍA GARCÍA is currently an associate professor with the Applied Software Engineering Group, Smart Computer Systems Research and Engineering Lab, Research Institute of Computer Engineering, University of Sevilla, Seville, Spain. Contact him at josemgarcia@us.es.

JOSE GARCIA-ALONSO is currently an associate professor with the Department of Informatics and Telematics System Engineering, University of Extremadura, Badajoz, Spain. Contact him at jgaralo@unex.es.

JUAN M. MURILLO is currently a full professor of Software Engineering with the University of Extremadura, Badajoz, Spain. Contact him at juanmamu@unex.es.

ANTONIO RUIZ-CORTÉS is currently a full professor and head of the Applied Software Engineering Group (ISA) with the Smart Computer Systems Research and Engineering Lab, Research Institute of Computer Engineering, University of Sevilla, Seville, Spain. Contact him at aruiz@us.es.

SCHAHRAM DUSTDAR is currently a full professor of computer science (informatics) with the Distributed Systems Group, TU Wien, Vienna, Austria. Contact him at dustdar@dsg.tuwien.ac.at.

IEEE Computer Society Has You Covered!

WORLD-CLASS CONFERENCES — Over 195 globally recognized conferences.

DIGITAL LIBRARY — Over 900k articles covering world-class peer-reviewed content.

CALLS FOR PAPERS — Write and present your ground-breaking accomplishments.

EDUCATION — Strengthen your resume with the IEEE Computer Society Course Catalog.

ADVANCE YOUR CAREER — Search new positions in the IEEE Computer Society Jobs Board.

NETWORK — Make connections in local Region, Section, and Chapter activities.

Explore all member benefits www.computer.org today!

IEEE COMPUTER SOCIETY

IEEE

EDITOR: Longbing Cao, longbing.cao@uts.edu.au

This article originally appeared in Intelligent Systems vol. 37, no. 4, 2022

COLUMN: AI FOCUS

Data-Driven Predictive Maintenance

João Gama, INESC TEC and FEP-University of Porto, 4200-465, Porto, Portugal
Rita P. Ribeiro, INESC TEC and FCUP-University of Porto, 4200-465, Porto, Portugal
Bruno Veloso, INESC TEC and University of Portucalense, 4200-465, Porto, Portugal

With the growth of 5G networks, the Internet-of-Things is becoming a reality. The advances in networking, machine learning, data analytics, and robotics largely improve industrial processes. Industry 4.0 is a term for the fourth industrial revolution: the digitization and automation of manufacturing.[1] Predictive maintenance is one of the techniques with high impact in today's industry.

INDUSTRY 4.0 AND MAINTENANCE

Maintenance is the process that deals with the health of equipment and system components to ensure their normal functioning under any circumstances. Over the years, and due to technological advances, different maintenance strategies have been developed. Three main maintenance approaches can be classified in the following.[2]

- *Corrective maintenance:* It means run-to-failure, which is the simplest and the oldest method. Actions are taken only after a machine or equipment fails. It would almost always lead to high (unexpected) downtime. This method usually generates a critical situation that will demand a great cost for companies.
- *Preventive maintenance:* It provides planning of regular replacement of components and/or equipment. It considers historical failure data and/or the data provided by the equipment manufacturer. Although this approach prevents unexpected shutdown, it usually needs additional costs and an increased unexploited lifetime.
- *Predictive maintenance (PdM):* It needs direct monitoring of the mechanical condition and other parameters to determine the operating conditions over time. Given the recent technological advances, we have tools that can process real-time data acquired from different equipment parts to predict any sign of failure.

PdM differs from preventive maintenance because it relies on the actual condition of equipment, rather than average or expected life statistics, to predict when maintenance will be required. Typically, machine learning approaches are adopted for the definition of the actual condition of the system and for forecasting its future states.

DATA-DRIVEN PDM

There are two main approaches for PdM: 1) Model based, and 2) data driven. In the model-based PdM, the equipment is described by physical equations, a digital representation of the equipment. Digital twins[3] are the most popular model-based approach. Data-driven PdM monitors the mechanical conditions or other health indicators of the equipment, and uses advanced statistical methods, such as machine learning, to find a pattern of functioning and dynamically determine the operating conditions over time.

Unlike the model-based maintenance approaches (e.g., preventive maintenance) that rely on forecasting the performance degradation using stochastic models, data-driven PdM practices are based on data without prior knowledge of degradation conditions. Its performance strictly depends on the analysis of signals and data. While model-based solutions can be expensive and inaccurate for complex systems, data-driven diagnosis methods are a promising alternative to fault/anomaly detection and isolation. Machine learning and deep learning algorithms and tools are naturally relevant to the PdM practices, mainly due to the large amount of data available (specifically the unlabelled ones).[4]

Failure Prediction

Fault detection is one of the most critical components of PdM. Nevertheless, PdM goes much behind predicting a failure. It is essential to understand the

1541-1672 © 2022 IEEE
Digital Object Identifier 10.1109/MIS.2022.3167561
Date of publication 25 May 2022; date of current version 12 September 2022.

consequences and what will be the collateral damages of the failure. One of the approaches in data-driven PdM is to learn the normal operating condition of a system such that any significant deviation from that operating condition could be spotted as a potential failure. The most popular techniques include autoencoders,[5] trained with data from the normal operating condition of the system. During the model exploitation, the system signals an alarm if the reconstruction error of the autoencoder exceeds a threshold.

PdM is generally employed based on the health status of critical elements. To avoid possible interruptions or even more severe damage, based on the operational history of different components, this strategy can predict failures over time, minimizing costs, and extending the useful life of the components.

Remain Useful Life—RUL

Different maintenance management policies can be employed using anomaly detection, diagnostics, and prognostics. The RUL is strongly related to prognostics, which provides the amount of time equipment will be operational before it requires any repair or replacement. Prognostic is directly related to the likelihood of system failure occurrence. It can be regarded as a forecasting process given the current machine conditions and its historical record.

A well-known technique to decide on the maintenance approach is P–F curve analysis (cf. Figure 1), which allows understanding of the condition of equipment over time. During the time between the detection of the potential failure and the actual failure, it is crucial to perform a maintenance action to address the problem before a functional failure occurs. As a fundamental task for RUL, prediction clearly defines the difference between run-to-failure

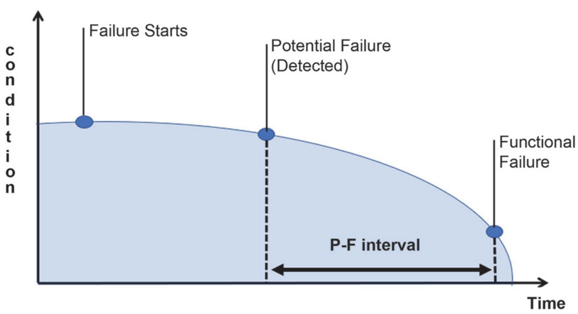


FIGURE 1. P–F curve is a graph that shows the health of equipment over time to identify the interval between potential failure and functional failure.

(corrective maintenance) and time-to-failure (prognostics) strategies. A categorization of methods and techniques for RUL can be found in Nguyen *et al.*'s work.[6]

Root Cause Analysis

This dimension looks for the causes and explanations about what could have happened when an anomaly was detected. The main concept that allowed this delimitation is inference, a fundamental task of AI. There exist many available techniques with different implementations. However, it is not common due to limitations to obtain any explanation from data.

OPEN ISSUES AND FUTURE DIRECTIONS

With the development of the Internet of Things, manufacturing technology, and mass production, the methodology of maintenance scheduling and management has become an essential topic in the industry. In these contexts, there is an urgent need for understanding and trust models and their results.[7] Understanding a predicted failure requires explaining where, why, and when a failure occurs. This is a relevant research topic aligned with the current trends in AI.[8]

Another critical issue in PdM applications is the design of the maintenance plan after a fault is detected or predicted. To elaborate the recovery plan, it is crucial to know the causes of the problem (root cause analysis), which component is affected, and the expected remain useful life of the equipment. Few works in PdM address this issue.

Given the growth in the digital virtualization of the world, the recent technology of digital twins[3] is the future for PdM. A digital twin is a virtual model designed to reflect a physical machine accurately. Vital areas of the machine are monitored using sensors. The data collected by the sensors is used to feed the digital copy of the machine. The digital twin can simulate scenarios, what-if analysis, etc.

ACKNOWLEDGMENTS

This work was supported in part by CHIST-ERA under Grant CHIST-ERA-19-XAI-012 and in part by project CHIST-ERA/0004/2019 funded by FCT.

REFERENCES

1. A. Rojko, “Industry 4.0 concept: Background and overview,” *Int. J. Inf. Manage.*, vol. 11, pp. 77–90, 2017.
2. R. K. Mobley, *An Introduction to Predictive Maintenance*, 2nd ed. Amsterdam, The Netherlands: Elsevier, 2002.

3. F. Strohmeier, C. Schranz, and G. Güntner, "I-maintenance: A digital twin for smart maintenance," *ERCIM News*, vol. 115, pp. 12–14, 2018. [Online]. Available: https://ercim-news.ercim.eu/en115/special/2095-i-maintenance-a-digital-twin-for-smart-maintenance
4. N. Davari, B. Veloso, G. de A. Costa, P. M. Pereira, R. P. Ribeiro, and J. Gama, "A survey on data-driven predictive maintenance for the railway industry," *Sensors*, vol. 21, no. 17, 2021, Art. no. 5739, doi: 10.3390/s21175739.
5. L. Wang, X. Zhao, J. Pei, and G. Tang, "Transformer fault diagnosis using continuous sparse autoencoder," *SpringerPlus*, vol. 5, no. 1, pp. 1–13, 2016.
6. D. N. Nguyen, L. Dieulle, and A. Grall, "Remaining useful lifetime prognosis of controlled systems: A case of stochastically deteriorating actuator," *Math. Problems Eng.*, vol. 2015, 2015, Art. no. 356916.
7. C. Molnar, *Interpretable Machine Learning*, 2nd ed., Munich, German, 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book
8. R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 93:1–93:42, 2019, doi: 10.1145/3236009.

JOÃO GAMA is a full professor at University of Porto, Porto, Portugal. His main research topics are learning from data streams and learning from network data. He is an EurIA fellow, IEEE fellow, fellow of the Asia-Pacific AI Association, and member of the board of directors of the LIAAD, a group belonging to INESC-TEC. Contact him at jgama@fep.up.pt.

RITA P. RIBEIRO is an assistant professor at the Department of Computer Science of the University of Porto, Porto, Portugal. She is the director of the master on data sciences and the senior researcher of LIAAD, a group belonging to INESC-TEC. Her research interests include imbalanced domain learning, outlier detection, evaluation issues on learning tasks, and problems related to social good. Contact her at rpribeiro@fc.up.pt.

BRUNO VELOSO is an assistant professor at University Portucalense, Porto, Portugal and an invited assistant professor at University of Porto. His research interests include data streams and automated machine learning. He is a member of Spanish Association for Artificial Intelligence AEPIA, and Senior Researcher of LIAAD, a group belonging to INESC-TEC. Contact him at bruno.miguel.veloso@gmail.com.

www.computer.org/cga

IEEE Computer Graphics and Applications bridges the theory and practice of computer graphics. Subscribe to *CG&A* and

- stay current on the latest tools and applications and gain invaluable practical and research knowledge,
- discover cutting-edge applications and learn more about the latest techniques, and
- benefit from *CG&A*'s active and connected editorial board.

IEEE COMPUTER SOCIETY

IEEE

EDITORS: Chris Johnson, crj@sci.utah.edu
Dave Kasik, dave.kasik@gmail.com
Mary C. Whitton, mcwhitton@gmail.com

This article originally appeared in ComputerGraphics vol. 43, no. 1, 2023

DEPARTMENT: VISUAL COMPUTING: ORIGINS

From Holocaust Hidden Child to Computer Animation Laboratory

Marceli Wein, *National Research Council of Canada, Ottawa, ON, K1A 0R6, Canada*

The article describes my long journey as a Jewish-born hidden child who was handed over to a Catholic family before the Krakow ghetto was eliminated in 1943. My father survived and I was reunited with him. We traveled to Germany in 1950 and were accepted as Canadian refugees in 1952. After an undergraduate and graduate program at McGill University, I was married in an Episcopalian/Anglican ceremony. My good fortune continued when I joined a research group at the National Research Council in the 1960s. The group worked on computer graphics and computer animation and won a Technical Academy Award for technology for the animated short Hunger/La Faim.

I am writing my story starting with my childhood in World War II as a Jewish *hidden child* raised as a Roman Catholic. Five years after the war, my father and I left Poland and ended up in Canada. I went through graduate school there where my Ph.D. project involved remote delivery of the weather radar images.[2] It was based on prior work by Tom Legg and produced CAPPI weather radar maps at selected altitudes. My work consisted of scan-converting constant altitude planar position indicator images to slow-scan raster images using 35-mm film as intermediate storage. Later I was fortunate to join a small group working on computer graphics and computer animation at the National Research Council of Canada. I worked with a Nestor Burtnyk-written program[2] that interpolated between hand-drawn key frames. The National Film Board of Canada sent Peter Foldes to create the art for the key frames. Our result, *Hunger/La Faim*, was nominated for an Academy Award as the first computer-animated film.

BACKGROUND

I am thinking back to my pre-SIGGRAPH days and recalling how I started in computer animation through a series of lucky incidents. I was born in 1934 in the historic Polish city, Krakow, into a Jewish family. My father was a tailor. He had a prosperous business before World War II. I discovered later that we lived in one of Krakow's Jewish districts.

My parents were not religious. Since my father was a successful tailor, we lived comfortably. I even had a nanny who took me to the park.

0272-1716 © 2023 IEEE
Digital Object Identifier 10.1109/MCG.2022.3218030
Date of current version 31 January 2023.

WORLD WAR II

I was four years old when World War II started. On 1 September 1939, Poland was invaded by Germany from the west and by the Soviets from the east. The invasion of Poland was very quick and very brutal but it left no impression on me—there was no fighting in my part of Krakow. In 1941, we were forced to move to a smaller walled-in ghetto called Podgorze (German for "bottom of a hill"). By then my father worked in a factory in which he probably made uniforms for the German army. Every morning he went with other workers in a group out of the ghetto to the factory. At various stages, the ghetto was reduced in size and people were more crowded.

This is the setting for my first lucky break. In the spring 1943, when I was eight years old, I was sent to a hospital in the Krakow ghetto with scarlet fever. My father heard that the hospital was being shut down and everyone in it would be killed. He visited me that night, wrapped me in a blanket, and smuggled me out.

The next day, I was smuggled out of the ghetto in Krakow. I remember walking out with my father and a group of Jewish workers as they went from the ghetto to a slave labor factory. On the way, he handed me to a woman along the route. The person to whom I was

FIGURE 1. This photograph of me was given to the German officer. My father had it at the end of the war.

handed over was a fine lady, Zofia Jezierska. I called her my aunt. We went to a flat in Krakow. After a few days I moved with her to her flat in Warsaw. I was now a "hidden child with a new name."

My father had thought that my 14-year-old brother Jerzy was old enough to manage with adults. Jerzy walked with a polio-induced limp and was shot. I learned of his death while I was still in Krakow, on the "outside" with Zofia Jezierska. I was devastated.

When my new aunt and I arrived in Warsaw, we heard that the Krakow ghetto was being liquidated and everyone was being shipped to camps. I saw fires about a kilometer away; the Warsaw ghetto was being liquidated by the Nazis.

I had no news about my parents during the war. Later I found out that both parents went to a series of concentration camps. My father was sent to Plaszow and survived Auschwitz as number 5 on Schindler's List. My mother did not survive.

My aunt made me a "hidden child" with the assumed "nom-de-guerre": Marek Czach. There happened to have been an available birth certificate of a child who had died in infancy. For reasons of expediency and safety, I was then raised as a Roman Catholic. In addition to me, Ciocia (aunt) Zofia Jezierska also hid a Jewish man, a pharmacist named Jozef Feiner. She later married Feiner.

While I was with Ciocia Zofia, a German officer came to the door and enquired if a Jewish boy lived there. The officer brought money from my father who was earning a little in the worker camp and trusted the officer. The same officer brought some photos of me

FIGURE 2. My first communion. I am on the left.

back to my father. My father had those photos at the end of the war. I still have one (see Figure 1). The event terrified me. I never discovered the officer's identity.

Soon thereafter, I said something I should not have and was immediately handed to another "aunt," Ciocia Szczepanska. I was sent to Grodzisk, about 30 km southwest of Warsaw. I stayed there for a few months. In Warsaw, I did not go to school but did in Grodzisk. I also prepared for my First Communion (see Figure 2).

In July 1944, I went back to Ciocia Zofia in Warsaw. The August 1944 Warsaw uprising gave me even more impressions of the cruelty of war. As the uprising set in, we spent most of the time in the basement. There were connecting holes broken through adjoining buildings that connected entire blocks. I can remember that there was very little to eat. The only food we had was made up of staples like flour and barley. As our part of the city was about to surrender, we had to crawl behind a barricade of sandbags across the street to a different area.

The 1944 uprising lasted until October 1944. Everyone was marched with few belongings to a sorting camp in Okecie on the Warsaw outskirts.

When the marchers stopped overnight, most slept on the road. That night, Ciocia Zofia, Jozef, and I entered a house where the residents hid us in the attic. When we woke in the morning, everyone on the road was gone. We went to a farm in the area and spent the next few months there. On 17 January 1945, the Soviets took over.

The war ended on the 9th of May 1945. I was located by a cousin, Leon Stach. My cousin was a Captain and doctor in the Polish army who came from the east with the Soviets. I learned from him that only my father survived. Figure 3 is a photograph of me in his staff car. For the month before my father came, I went to scout camp in the Grodzisk area.

AFTER THE WAR

Between 1945 and 1948, I lived in Chorzow in Upper Silesia with my (real) Aunt Helena Krenicer. She had

FIGURE 3. My photograph on Leon Stach's staff car after he found me in 1945.

FIGURE 4. B.Eng. 1958 with chaplain of newman club.

reopened her dentistry practice and we lived comfortably. I remember the Auschwitz prisoner number tattooed on her forearm. From 1948 until 1950, I lived with my father in Bytom (formerly Beuthen) in Upper Silesia.

After World War II, my father married a German from Silesia! From then on, I had a new family: a German stepmother (Elisabeth Stein Wein) and a stepbrother, Elisabeth's son (Horst Lempka). While I did not resent Elisabeth as my stepmother per se, I did resent for a few years the fact that she was German.

My stepmother and stepbrother were allowed to leave Poland in 1950 as Germans who did not want to take Polish citizenship. My father arranged for the rest of the family to be designated as Jews and received permission to travel abroad. We visited several countries to find a place we could go to live. None would accept us. We went to Munich and lived with my stepmother and stepbrother from 1950 until the end of 1951. I learned English by listening to the U.S. Armed Forces Radio Network. It was in Munich that my father applied to emigrate to Canada. Canada accepted us as refugees.

My father and I came to Montreal in January 1952. We were sponsored by the Canadian Jewish Congress. On the ship, we were given a choice to go to Montreal or to Winnipeg. Fortunately, we chose Montreal.

The desolate view of the bay entering the harbor in January shocked me. Going to Baron Byng High School (almost all Jewish) as a tenth grader shocked me almost as much as French Canadian culture and the bay. My stepmother and stepbrother arrived in Canada a few months later.

After high school, I received a small Seagram's scholarship and a McGill University Scholarship. In 1958, I got my degree in engineering physics with honors in electrical engineering. During my McGill career, I was a member of the McGill Catholic Newman club (see Figure 4) and no Jewish organizations.

All the graduating students were accompanied by large families except I was not. I was accompanied only by my father, stepmother, and stepbrother but I was soon to discover my large surviving Jewish family.

FINDING FAMILY

It was not until there was publicity around the film *Schindler's List* (1993) that I realized the significance of my family background. That is when I discovered the original Schindler's list and that my father, Wolf Wein, was on the list as Number 5. He was recorded as being a master tailor. He may have made uniforms for Schindler.

I regret deeply that I did not question my father about wartime events. As is the case with many, he was reluctant to talk about wartime.

Some years ago, I located one cousin on my father's side and several cousins on my mother's side in England and in Germany. I learned that my grandmother was Taube Ruchel Hofstadter Nachman. Through her, I discovered that my second cousin was the historian, Richard Hofstadter. I am thrilled that I have many more relatives of different religious persuasions who survived and are scattered around the world. This family history sets the stage for my long, successful, professional career.

EVOLVING TO *HUNGER*

My first engineering job after graduating from McGill (1958) was in manufacturing magnetrons that could be used in rockets and in radars. But after a few months, production was stopped and I was transferred to designing television sets.

In 1959, I had another lucky break. On 24 June (the St. Jean Baptiste Quebec holiday), I went to the McGill campus with the intention of asking a girl whom I knew and who worked in the Physics Department to have lunch with me. As I entered the building and before I met her, I was met by one of my physics professors. The professor thought I had come to see him, so he gave me a guided tour of the Stormy Weather Group, headed by J. Stewart Marshall. By the end of the afternoon, I was accepted into the M.Sc. program.

FIGURE 5. Weather radar image as transmitted to a facsimile.

MY PH.D. WORK IN WEATHER RADAR

I learned about transferring images to film, critical to my work in computer animation, while pursuing my Ph.D.

After my 1961 M.Sc., I continued to complete my Ph.D. in physics. My Ph.D. project under J. Stewart Marshall dealt with weather radar and how to send the radar images over a telephone line to a distant weather office.[2] First, the images had to be generated. The radar image appears like a modern radar image because of three factors. First, the image was created at a specified altitude by capturing rings as the antenna was raised at angle increments. Second, the stepping logarithmic amplifier created the image. Third, another process prepared the image for scan conversion. These steps were developed at McGill University by my predecessor in the group, Tom Legg.

My work on the image transmission involved scan conversion and was entirely analog. I had access to a rapid film processor that consisted of a series of three gates for 35-mm film and two 5-inch flat CRTs. The radar image was a planned position indicator radial scan, the same format still used today for weather radar displays. The image was produced on the first CRT and was captured on 35-mm film in the first gate. The film was then advanced to the second gate where it was sprayed with the developer, then fixer, and then dried. The film was advanced to the third gate where it was scanned with a slow rectangular raster scan on the second flat CRT. The analog signal was captured through an optical system by a photomultiplier tube, processed by stepping circuitry, and then transmitted over a telephone line via an analog facsimile protocol to a standard weather office facsimile machine. Figure 5 shows an example.

Scan conversion required image storage. We used 35-mm film. Digital storage was totally impractical at that time because of large memory requirements and steep cost. These activities were well before the RS232 digital data transmission standard existed. As a result, I gained a lot of experience capturing gray scale radar images on 35-mm film.

FIGURE 6. Peter Foldes at the IDI display.

The radar project was part of activities in the Stormy Weather Group in the Physics Department that was funded by United States Air Force Cambridge Research Center (AFCRL).[1] The AFCRL had a much larger program on weather radar than Canada. I recall the two leading scientists: David Atlas, the head of the program, and Ed Kessler. About the time I completed my Ph.D., Ed Kessler went to Norman Oklahoma and started the National Severe Storm Laboratory, where he introduced Doppler radar that used klystrons. Klystrons became the basis of modern weather radars.

EARLY DAYS AT NATIONAL RESEARCH COUNCIL

After a year in the Physics Department, I accepted a job at the National Research Council of Canada. I started as a Research Officer in computing, an activity that NRC was trying to start. I was lucky again because I became a junior staff member and got to work in interactive computer graphics with brilliant colleagues, Nestor Burtnyk and Ken Pulfer.

We were interested in how nontechnical people could interact with computers. We intended to investigate human–computer interaction, especially for nontechnical people. That was in 1966. We acquired a minicomputer from Systems Engineering Laboratories, a model SEL840A. The computer was the size of two refrigerators but was still called a "Mini." It had 8K 24-bit words, hence 24K bytes.

Ken Pulfer designed and built the controller for an Information Displays Inc. (IDI) point-plotting display that Carl Machover sold us. Although the IDI display (see Figure 6) came with a lightpen, we did not use it. Pulfer copied the mouse Doug Engelbart designed at Xerox in 1963 for input interaction. For animation drawing and digitization, we used a Computek tablet, which provided 10-bit *x* and 10-bit *y* packed into a single 24-bit input word.

There was no compiler for any high-level language for the computer, only an assembler for assembly language programs. Programs were entered on a teletype

FIGURE 7. Peter Foldes and Marceli Wein at the Computek tablet.

that produced paper tape. The computer had a high-speed paper tape reader and produced the machine language "object" version of the program. A loader accepted all the relevant object programs to build a runtime version of the program.

We learned assembler language programming by experimenting and created a program that would display something on the IDI screen. We then wrote a program that would read the mouse pulses to increment or decrement screen position. Eventually, we had enough to build more complex programs.

Nestor Burtnyk attended a conference in Los Angeles in 1969 at which the guest speaker was an artist from Disney Studios. The speaker suggested that computers could be used to generate the in-between cels in film animation. Upon return, Burtnyk wrote a program that generated the in-between frames for beginning and ending two-dimensional (2-D) images that were drawn on the tablet.[3] Somewhat later we defined an image as a series of strokes. Each stroke was a series of short, interconnected line segments that were drawn from pencil-down to pencil-up. Frame-to-frame interpolation was between strokes.[4] We improved the interpolation software to make it accelerate and decelerate near the end points. We used algorithms from my undergraduate textbook by Slater and Frank.[5] All the implementation was done on the SEL840A using assembly language, paper tape, the Computek tablet, and the IDI display.

MAKING *HUNGER/LA FAIM*

A visiting producer, Rene Jodoin from the French Animation Section of the National Film Board of Canada, thought that the line animation style was suitable for a script that had been submitted by Peter Foldes. Foldes was an animator in France who had submitted a script for *Hunger/La Faim* to the Film Board in Montreal. The script was reviewed and approved but was waiting for a suitable technical support and funding. Jodoin arranged for Foldes to travel to NRC in Ottawa in 1970.

FIGURE 8. Record of how the sequence of strokes in a cel was recorded.

Foldes experimented with the interpolation software (see Figure 7). He liked it enough to use it to produce an experimental film called *Metadata*. We recorded *Metadata* on 16-mm film from images displayed on a small, flat display-screen, separate from the IDI display. The Film Board released *Metadata* as a film in 1971 that is still listed in the NFB catalog (https://www.nfb.ca/film/metadata_en/).

We learned that we needed a pin-registered 16-mm camera so that the film could be backed up (in the sense of reversed) accurately for multiple exposures. We could not compose an entire frame at once because of limited processor memory. Our work-around was physical compositing on film. Also, we had to record the foreground and background separately. The pin-registered camera ensured that when the film was projected the image was steady on screen.

Jodoin arranged for Foldes to make trips from Paris to Ottawa to work on the drawings. Foldes stayed with my family for a week at a time. In preparation, we installed a set of animation registration pins on the Computek tablet so that standard, prepunched drawing sheets could be used.

While Peter Foldes stayed at our house, we made plans for multiple exposures and multiple film sequences, including raster scanned mattes for the foreground. We discussed the sequences that would be drawn the following day. Peter would input sketches with the tablet the following day. He worked on the studies in the evening at my house on each trip. For each cel, he made a separate sketch and numbered all the strokes in groups (see Figure 8). Peter then drew the cels (see Figures 9 and 10) the following day and preserved the group order.

He completed the drawing phase after a few visits. We then recorded all the parallel sequences on rolls of 16-mm film, commercially processed the film in

FIGURE 9. Cels from the film *Hunger/La Faim* by Peter Foldes, National Film Board 1974.

FIGURE 10. Film strips: Foreground line, foreground 1 matte, foreground 2 matte combined in the optical printer.

Ottawa, and then had it sent to the National Film Board in Montreal.

Once the film material was sent for processing, we had time to update the drawing program. This allowed the artist to see the previous cel in low brightness while the new cel was being drawn. The stroke that corresponded to the one to be drawn was shown in full brightness. The bright stroke advanced as soon as the pen was raised. However, there was not enough storage to establish a database of cels. The database was stored on seven-track digital tape and cels had to be selected manually.

At the National Film Board, Rene Jodoin supervised the work on the optical printer. (An optical printer consisted of a film frame projected using 1:1 optics onto a fresh film frame. Both film gates were pin registered to maintain accuracy. A color filter could be inserted to add color.) The 16-mm film was enlarged to 35-mm film and color was applied to the background. The completed short film featured music by Pierre F. Brault. *Hunger/La Faim* was released in 1974[6] and submitted to the Academy of Motion Pictures without mentioning the computer role in its production. The film was nominated for an Academy award in the Animated Shorts category. Peter Foldes and Rene Jodoin attended the Academy Awards Ceremony in Los Angeles.

It was only a couple of years later that Unix and C were published by (then) Bell Telephone Laboratories.

FIGURE 11. Technical Academy Award, 1997 with Helen Hunt.

They were available on a Digital Equipment Computer minicomputer NRC purchased to replace the SEL. We were now old timers and had to learn how to program in C to feel comfortable with newly minted and hired computer science graduates. The programming language C felt comfortable because it was close enough to the Assembler language yet high enough to be quite structured. It represented a real breakthrough for Brian Kernighan and many more advanced languages have been based on C.

I arranged to work with William Cowan at University of Waterloo in human–computer interaction and became adjunct professor at the university.

Nearly 20 years later, after *Toy Story* was released and achieved great success, the Academy searched for the earliest impact of computer animation in the motion film industry. Ed Catmull, then president of Pixar, identified *Hunger/La Faim* as an early Academy award nominee and therefore the earliest example. Catmull nominated Burtnyk and me for a Technical Academy Award. We attended the Science Technical Academy Awards Ceremony and received the award in the form of an Academy certificate. While we received a certificate (see Figure 11), the single statuette went to the creators of IMAX.

Jim Kajiya received an award for animating sand in the film *Twister* that starred the host of the event, Helen Hunt. The event was two years after my retirement.

Burnyk and I were interested in offering tools to traditional animators whom we admired as extremely talented artists. In particular, Peter Foldes was such a talented artist when he took a pen or pencil to paper. We saw the emergence of computers with 3-D animation destroying the traditional media and such talent. We were thrilled that our more modest computer tools could be successful.

I have thought how I ended up in my career. In North America, I have observed that university-educated families took it for granted that their children would attend universities. But families in the trades did not. My father had made a great effort to get me through the war. He then took it for granted that I would attend university even though he had stopped at grade four. After I entered graduate school, I felt the continued urge to survive, advance, and was grabbed by the urge to learn as much as possible. Then I was taken by the interest in how computers generated images, a new research topic. That interest blossomed in the mid 60s when I discovered computers and the opportunities they offered to produce animated films.

The activities at the National Research Council had a strong positive influence on computer graphics in Canada. In our earliest days, we worked with Ron Baecker at the University of Toronto and later with several universities. We organized and sponsored the Canadian Man Computer Communications Conference that was held at NRC in 1969. The conference attracted participants from many universities. Many showed extraordinary work that has been lost to history. We hosted a second conference in 1971. The conference became biennial and independent of NRC. The Conference went through name changes, the first Canadian Human Computer Communications Conference and, in 1980, the conference changed its name to Graphic Interface (GI). While we participated in SIGGRAPH once it started, we continued to support GI and it has continued providing a focus for activities at Canadian universities.

ACKNOWLEDGMENTS

I acknowledge the leadership of Nestor Burtnyk and the late Ken Pulfer in the leadership of the computer graphics project. I greatly appreciate the great assistance of David Kasik in preparing this manuscript.

REFERENCES

1. M. Wein, "Areal integration for weather radar," Scientific and Technical Aerospace Reports, Tech. Rep. Air Force Cambridge Laboratories AFCRL-65-339, Apr. 1965.
2. M. Wein, "Facsimile and areal integration for weather radar," PhD Thesis, McGill Univ., Montreal, QC, Canada, 1964.
3. N. Burtnyk and M. Wein, "Computer generated key frame animation," *J. Soc. Motion Picture Telev. Engineers*, vol. 80, no. 3, pp. 149–153, 1971.
4. N. Burtnyk and M. Wein, "Interactive skeleton techniques for enhancing motion dynamics in key frame animation," *Commun. ACM*, vol. 19, no. 10, pp. 564–569, Oct. 1976, doi: 10.1145/360349.360357.
5. J. C. Slater and N. Frank, *Mechanics International Series in Pure and Applied Physics*. New York, NY, USA: McGraw-Hill, 1956.
6. *Hunger /La Faim a film By Peter Foldes, 10 Minutes*, R Jodoin. Montreal, Canada: National Film Board of Canada, 1974.

MARCELI WEIN retired in 1985 from the National Research Council of Canada. He received his Ph.D. degree in physics from McGill University, Montreal, QC, Canada. Contact him at mwein@hotmail.com

Contact department editor Chris Johnson at crj@sci.utah.edu or department editor Dave Kasik at dave.kasik@gmail.com or department editor Mary C. Whitton at mcwhitton@gmail.com.

REMEMBERING FREDERICK P. BROOKS, JR. (1931–2022)

Pioneering computer scientist Frederick P. Brooks, Jr. died November 17, 2022, at his home in Chapel Hill, NC, USA.

Dr. Brooks is widely known as the author of *The Mythical Man Month: Essays on Software Engineering*. In it, Brooks clearly describes the lessons he learned while managing the development of IBM OS/360. After leaving IBM in 1964, he founded the Computer Science department at the University of North Carolina at Chapel Hill. As chair for 20 years, he led the department to international prominence in computer graphics.

Brooks' mantra was that *computer scientists are toolsmiths*: they build tools others use to solve their own hard problems. For example, UNC-Chapel Hill computer scientists built systems for displaying and manipulating molecular models for protein chemists and nanoscale structures for material scientists. Dr. Brooks cared deeply about the relationship between people and computers. His research group extensively studied system characteristics, such as latency to understand their impact on the effectiveness of virtual reality systems.

Brooks received dozens of major awards, including the ACM A.M. Turing Award, the IEEE VGTC Virtual Reality Career award, and membership in the SIGGRAPH Academy and the IEEE VGTC Virtual Reality Academy.

EDITOR: David C. Brock, dbrock@computerhistory.org

DEPARTMENT: BIOGRAPHIES

This article originally appeared in IEEE Annals of the History of Computing vol. 45, no. 1, 2023

The Legacy of Mary Kenneth Keller, First U.S. Ph.D. in Computer Science

Jennifer Head, *Sisters of Charity of the Blessed Virgin Mary, Dubuque, IA, 52003, USA*

Dianne P. O'Leary, *University of Maryland, College Park, MD, 20742, USA*

The founding of the first computer science programs at universities in the United States in the early 1960s resulted in 1965 in the first two doctoral-level degrees. On June 7, 1965, Irving C. Tang received a D.Sc. from the Applied Mathematics and Computer Science Department, Washington University, St. Louis. Tang's thesis was entitled "Radial Flow Between Parallel Planes," directed by C. David Gorman of the Mathematics Department. The same day [27], Mary Kenneth Keller received a Ph.D. in Computer Science from the University of Wisconsin, Madison. Ralph London documented this, correcting the historical record [26], [27].

Here, we focus on Mary Kenneth Keller[1] (Figure 1). She was a woman of many names. Born Evelyn Marie Keller, she entered the Catholic congregation of the Sisters of Charity of the Blessed Virgin Mary (BVMs) and was given the name Sister Mary Kenneth. In her publications, she used the name Sister Mary K. Keller [8], [19], [20], [21], [22], [23], Sister Mary Kenneth [25], Mary K. Keller [18], and (her complete name) Sister Mary Kenneth Keller, BVM [24]. In this article, we refer to her as Sister Kenneth.

LIFE BEFORE COMPUTING

Evelyn Keller was born December 17, 1913, in Cleveland, Ohio, but spent most of her youth in Chicago. Both of her parents had 8th grade educations. She graduated from The Immaculata High School, a BVM school, in 1931. Mary Jerellen Tangney, BVM, wrote in 1932 that Evelyn was an excellent student, excelling in English and Journalism. Evelyn entered the BVM congregation in September 1932, at age 18. She was formally received into the community on March 19, 1933, and it was then that her name was changed from Evelyn to Mary Kenneth.

For the next three years, Sister Kenneth focused on spiritual development and also attended Clarke College (Dubuque) and Mundelein College (Chicago), women's colleges founded by the BVMs. After taking religious vows on March 19, 1935, she began her teaching career, assigned for the next 29 years to work in elementary and high schools in Illinois and Iowa. In those times, sisters typically finished their academic degrees "on the installment plan" in summer school so as not to interfere with their teaching assignments. Sister Kenneth took additional courses from Clarke and Mundelein, as well as 13 credit hours from DePaul University (Chicago), which awarded her B.A. in mathematical sciences, with a minor in Latin, in 1943. She took 30 hours of course work at DePaul between 1946 and 1952 to earn an M.S. in mathematics.

INTRODUCTION TO COMPUTING

As a high school math teacher on the west side of Chicago in her mid-40s, Sister Kenneth "read the signs of the times and as early as 1961 responded by enrolling at Dartmouth College in Hanover, New Hampshire for her first workshop in computer education."[2] As Sister Kenneth told it, "I just went out to look at a computer one day, and I never came back.... It looked to me as if the computer would be the most revolutionary tool for doing math that I could get" [35]. Anthony Knapp, an undergraduate staff member for the summer workshop, recalls that programming of Dartmouth's LGP-30 computer was done in assembly language. The drum memory contained 4096 32-bit words, addressed by track and sector, and arithmetic was

[1]This is an abridgement of the biography at https://arxiv.org/abs/2208.01765.

[2]Dolores Marie McHugh, BVM, January 14, 1985.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/
Digital Object Identifier 10.1109/MAHC.2022.3231763
Date of current version 3 March 2023.

FIGURE 1. Sister Kenneth's high school yearbook photo and a photo from later in life. Photo credits: BVM Archives and Clarke University Archives.

fixed-point. The participants used a sign-up sheet to reserve the computer for their use and completed simple programming tasks, such as playing tic-tac-toe. Contrary to published reports (e.g., [13]), Kurtz wrote that Sister Kenneth "had absolutely nothing to do with the creation of the BASIC computer language,"[3] first released in May 1964.

PH.D. STUDY

In the early 1960s, Mary Benedict Phelan, BVM, president of Clarke College, decided that students should prepare for the information age. Sister Kenneth was sent to the University of Wisconsin, Madison (1962–1965), to complete a Ph.D. degree in order to lead this program. Her break in teaching assignments was only one year, 1964–1965, so again much of her study must have been during summers. She completed 15 credit hours in computer science and 21 credits of research, supported by a National Science Foundation Fellowship (1962–1964) and then by a university fellowship. Her academic advisor was Preston Hammer, Professor of Numerical Analysis and (as of 1964) Professor of Computer Science.

Sister Kenneth began her dissertation [24], "Inductive Inference on Computer Generated Patterns," with the declaration, "The basic problem of this thesis is the exploration of an approach to the mechanization of inductive inference." The idea was to automatically generate the nth mathematical expression in a series, given the first k expressions. As validation, Sister Kenneth implemented a FORTRAN program to determine the nth derivative of a function, given the first k derivatives. (She remarked that list processing languages such as LISP would have been more convenient but were not available to her.) She wanted to demonstrate that algorithms could perform tasks like differentiation through learning-by-example, rather than by a rule-based process. Although this approach was out of favor for many years, it has recently had a renaissance in so-called "deep learning models" in artificial intelligence and now dominates the field. Sister Kenneth did not claim that her "results obtained in the domain of analytic differentiation were important in themselves," but that they demonstrated that her programs "do mechanize an inductive inference....accomplished in less than 30 seconds per problem" [24, pp. 46–47].

She concluded with her first statement on the relation between human and artificial intelligence: "The claim that this parallels the activity of the mathematician who generalizes on a symbol manipulation routine after some number of repetitions must, of course, be qualified" [24, p. 46].

The thesis was signed on May 21, 1965, by Preston Hammer, Ralph L. London, (programming languages and verification), and Eldo Koenig (man–machine interactions and robotics). The "major department" is listed as "Computer Sciences" and the minor as "Mathematics" [24, p. 67]; see Figure 2. She listed membership in the Society for Industrial and Applied Mathematics (SIAM), the Association for Computer Machinery (ACM), and the Mathematical Association of America (MAA).

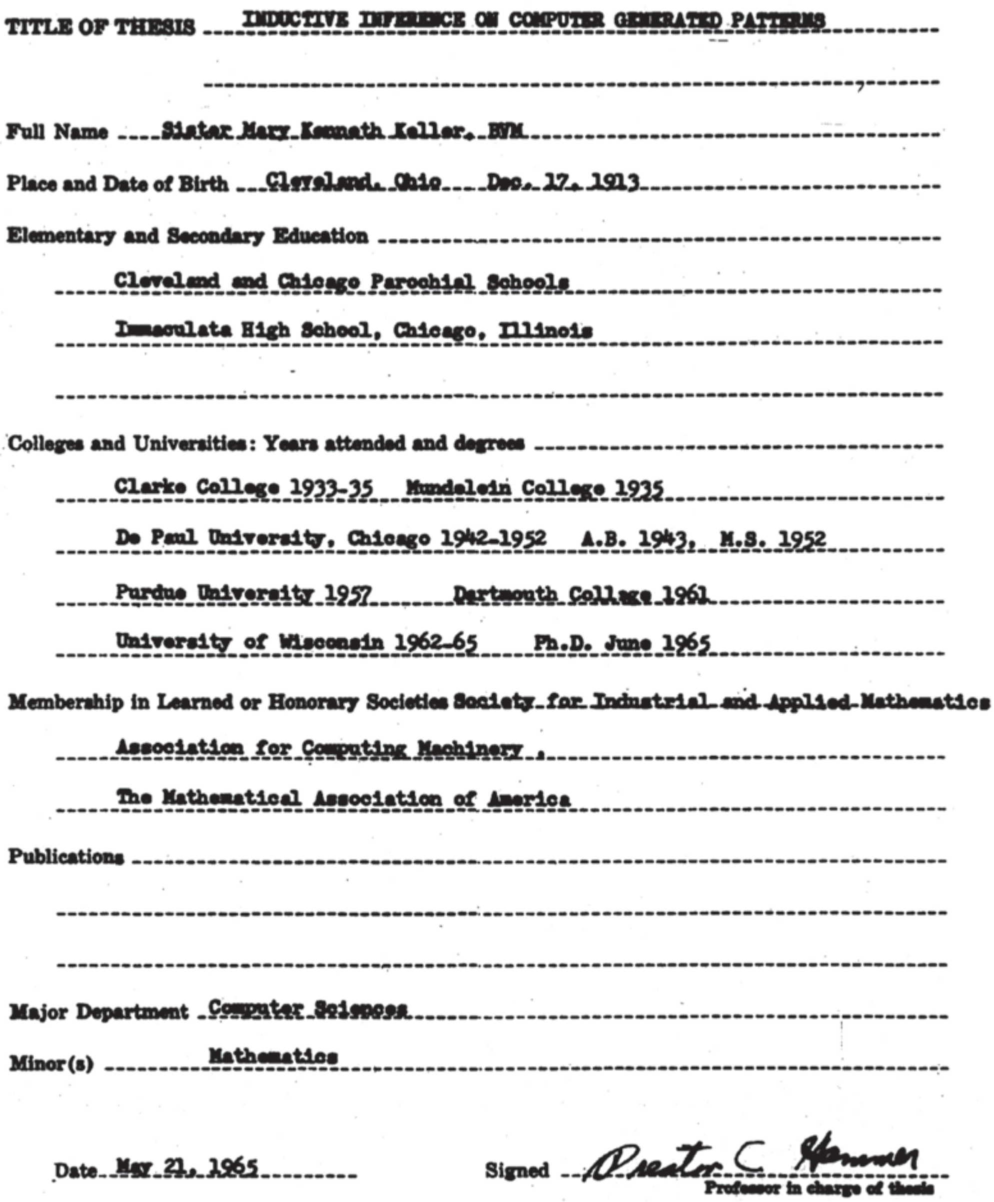
After 29 years of K-12 teaching and 33 years of college-level studies, Sister Kenneth had finally completed her formal education. She was 51 years old.

WORK AT CLARKE COLLEGE

Sister Kenneth arrived at Clarke College with a vision: "Automation, or more precisely, cybernation, is a fact of our lives. Its impact is swift and in many ways silent. Our sense of responsibility should make us wish to be informed and to inform our students.... Furthermore, the computer can give a new dimension to the education we offer" [29]. She was appointed head of the "Center for Computer Sciences" at Clarke, in charge of a desktop computer, the Bi-Tran Six, (Figure 3) with 128 6-bit words of memory, visible circuit boards, and programming in assembly language [12]. An IBM 1130 with 8 K memory and paper tape reader was installed in 1966, later upgraded to punched card input. Keeping the computer facility up-to-date on a shoe-string budget was a continuing struggle, absorbing much of her time and effort.

In addition to managing academic and administrative computing at Clarke, Sister Kenneth kept her hand in research; in 1975, she was collaborating with Raymond Martin at Wartburg Theological Seminary, using the computer to compare the Hebrew, Aramaic, and Greek texts of two books of the Bible [35], research that was related to Martin's previous work [28].

[3]Thomas Kurtz, January 11, 2019 email.



TITLE OF THESIS INDUCTIVE INFERENCE ON COMPUTER GENERATED PATTERNS

Full Name Sister Mary Kenneth Keller, BVM

Place and Date of Birth Cleveland, Ohio Dec. 17, 1913

Elementary and Secondary Education

Cleveland and Chicago Parochial Schools

Immaculate High School, Chicago, Illinois

Colleges and Universities: Years attended and degrees

Clarke College 1933-35 Mundelein College 1935

De Paul University, Chicago 1942-1952 A.B. 1943, M.S. 1952

Purdue University 1957 Dartmouth College 1961

University of Wisconsin 1962-65 Ph.D. June 1965

Membership in Learned or Honorary Societies Society for Industrial and Applied Mathematics

Association for Computing Machinery,

The Mathematical Association of America

Publications

Major Department Computer Sciences

Minor(s) Mathematics

Date May 21, 1965 Signed Preston C. Hammer
Professor in charge of thesis

FIGURE 2. Page 67 of Sister Kenneth's thesis [24].

Under Sister Kenneth, the computer center was a family-friendly environment. After Kathy Decker (Clarke '74) was hired, despite the cramped quarters, Sister Kenneth provided nursing and play space for her children [5].

By 1978, Sister Kenneth advocated using microprocessing technology in place of large, central computing systems.[4] In 1980, she received an NSF grant to equip a lab with 20 microcomputers, shown in Figure 4, for classroom use. The machines could be configured with color graphics, sound, light pens, plotters, and more, one more step toward serving liberal arts students. The newly named Keller Computer Center [6] had tripled in size from its original classroom [29] to 2750 square feet [10] reclaimed from Clarke's laundry building, and had an IBM 4331 with a megabyte of memory [33]. By 1982, there were 21 Apple computers on campus.

Contributions to Clarke College Education

Sister Kenneth became an evangelist for computer science, determined that the women graduating from Clarke

[4]Sister Kenneth letter to M.C. Rayner, May 12, 1978.

FIGURE 3. Sister Kenneth with Bi-Tran Six computer. Photo credit: Clarke University Archives.

FIGURE 4. Sister Kenneth in Clarke's Computer Center in September, 1980. Photo: Reprinted with permission of the *Dubuque Telegraph Herald*.

would be ready for the computer age. Until 1968, she was "a one-woman computer sciences department" [9]. Immediately upon taking her position at Clarke in Fall 1965, Sister Kenneth initiated a 3-credit-hour introductory course and established a minor concentration in computer science, a first for private colleges in Iowa. She initially did not want a computer science major, believing that the liberal arts should be encouraged.[5]

"Sister Computer" noted in 1965 that a broad mathematical background was not necessary to study computing; essential qualities were confidence, humility, and patience. "The computer does not make mistakes, and it is hard on one's ego to have to assume all the blame every time something goes wrong. That is where humility comes in and then you get to practice patience by correcting your mistakes" [29].

Sister Kenneth was a gifted teacher and communicator, even tutoring R. (Richard) Buckminster Fuller (designer of the geodesic dome) (Figure 5). Fuller said, "I wondered if I'd be able to do this successfully at the age of 73," but he found that he could, "slowly" [31]. "He believed that technology should be directed to 'livingry' rather than to military power and weaponry" [5], a view shared by Sister Kenneth.

[5]Mary Louise Caffery, BVM, July 2, 2019.

Sister Kenneth was a witty teacher, engaging each person as she walked around the room. In addition to regular courses, she taught evening adult education in FORTRAN and assembly language. She "coerced" her academic colleagues into learning computing, telling artists and musi-

FIGURE 5. "73-year-young R. Buckminster Fuller is taking a two-day crash course in computer science and programming from Sister Mary Kenneth" [11]. Photo credit: Clarke University Archives.

cians, "You have to know this!"[6] She got them "hooked up on that Apple computer," and even told IBM representatives that they needed to build a machine like that, rather than just producing large mainframes. She was "just so pleased if people used the computer in a new way."[7]

In 1973, Sister Kenneth moved from teaching into full-time administration. In 1977, a merger at Clarke created the Department of Computer and Management Sciences, with a Women in Management Program to integrate the disciplines. This doubled the size of the faculty, from two to four. Clarke established a Bachelor of Arts degree in 1979, with 7 graduates in 1980. The students were quite successful in finding work-study positions, summer jobs, and permanent positions [4]. A Bachelor of Science degree was added in 1982. By that time, Computer Science at Clarke had 51 students, including 30 masters candidates [1].

Sister Kenneth continued her advocacy for computer education in secondary schools, running NSF Summer Institutes in Mathematics for Secondary School Teachers from 1968 to 1972. In 1981, Clarke established a summer Masters Program in Computer Applications in Education, "among a very few such programs nationwide" [1]. Fifty students made up the first cohort, completing 9–12 credit hours in education, 15–18 in computer science, and 3–6 in electives. David Fyten, a Clarke faculty member, said, "Many teachers feel that the introduction of microcomputer education has rejuvenated their desire to stay in teaching."

In the January 1983 semester, Computer Science ran 10 courses with 19 sections averaging 25 students per section, taught by five FTE instructors. Sister Kenneth was asked to prepare a 5-year plan for the computer center. She complied but noted that it "can only be a projection from the present, and not too real. Consider that five years ago almost nothing that is in the center today existed. The field is in even greater flux today."[8] She did know one thing with certainty: "A search for a new director of the computer center should be started," since she was dealing with breast cancer after surgery in August 1982, and knew that she needed to retire.

COMMUNITY SERVICE

Service to Industry

During the late 1960s, Sister Kenneth established herself as a local expert on computing hardware, software, and applications. She was much in demand by local government [33] and industry (e.g., Figure 6) and responded by organizing seven Iowa Technical Service Seminars for Business and Industry and by consulting for companies ranging from from Ertl Toy Company to Magma Copper Mines. "People called her up, and she solved their problems."[9] The relationships she developed supported Clarke in the 1970s, for example, through the donation of a used IBM 360-40 and through contracts, such as one to develop software for John Deere Dubuque Works. "Almost single handed she equipped the computer lab by giving lectures in colleges and universities, conferences, conventions, lunches, dinners, wherever she could reach an audience and be paid to do it" [33]. In those lectures, Sister Kenneth often put considerable emphasis on ethics, saying that computers should be used to alleviate poverty and ignorance and to open up new possibilities in industry rather than replacing current workers [3].

FIGURE 6. Sister Kenneth at an August 2/3, 1967 conference at Clarke College on Hospital and Medical Applications of Computers. Photo credit: Clarke University Archives.

In a 1967 speech on "Computer Applications from a Management Point of View," Sister Kenneth noted that advances were hampered by a shortage in computer programmers and by dataflow restrictions within companies:

> The concept of a database where all items are organized on a tree with interconnected branches would seem to be the correct principle of operation for the future,... a complete departure from the structuring of data files in the traditional sense where a payroll file belongs to one department and a separate personnel file [exists] for the use of the personnel department.... There will likely be a greater use of... visual display output which is meant to be a momentary presentation of information for view by the executive on an on-call demand basis,

[6]Carol Blitgen, BVM, July 2, 2019.
[7]Mary Louise Caffery, BVM, July 2, 2019.
[8]January 25, 1983 memo
[9]Mary Louise Cafferty, BVM, July 2, 2019.

whereas the hard copy output will be limited to traditional reports and output documents.

She also highlighted the use of computers in process and production control and in simulation. All of this depended on a sufficient supply of knowledgeable programmers, and she established a pipeline of educated candidates from Clarke and other small colleges and universities.

Service to Education

Sister Kenneth was a founding member of the College and University Eleven-Thirty Users Group (CUETUG) which, as IBM 1130 s were retired, was renamed the Association of Small Computer Users in Education (ASCUE). She spoke at their meetings (e.g. [15], [16], and [17]) and represented ASCUE on the steering committee for the National Educational Computing Conference. She served as a Board member (1974–1976) and as Public Relations Director (1977–1984) until shortly before her death.[10] Sister Kenneth was active in ACM's Curriculum Committee for Undergraduate Computer Science and worked on the Masters-level curriculum. She spoke at many professional meetings and consulted on curricula and computer facilities for small colleges (e.g., [14]). The highly influential ACM *Curriculum'78*, a model for undergraduate computer science education, lists her as a contributor [2, p.166]. In April 1980, she testified before Congressional subcommittees on information technology in education.

The IEEE Computer Society honored Sister Kenneth by establishing the Mary Kenneth Keller Computer Science & Engineering Undergraduate Teaching Award, which has been given most years since 1999.[11]

TEXTBOOK AND EDUCATIONAL MODULES

Textbook

Sister Kenneth collaborated on a 1973 textbook [8] based on an earlier textbook by Dorn and Greenberg entitled *Mathematics and Computing with FORTRAN Programming* [7]. The newer version omitted applications in linear algebra and calculus and used the more accessible BASIC programming language in order to reach a broader audience.

Sister Kenneth's contributions began with a 43-page introduction to BASIC programming [8, Sec. 1.1–1.13 and 2.4], developing the ideas of an algorithm, flowchart, and code through the solution of two linear equations. Then, she systematically taught the language. Her attempts to prevent student errors were remarkable; for example, she carefully explained the rather confusing rules for variable and array names. The level of clarity was noticeably higher than that of the following section about computer arithmetic, taken from Dorn and Greenberg.

Sister Kenneth's approach to pedagogy is most evident in the book's Appendix A, where she stepped students through their first encounter with using a teletype, from sign-on to sign-off. It seemed important to her to provide students with every piece of information they would need to succeed.

Two other sections by Sister Kenneth complemented the discussion of `and-or-not` computer circuits by introducing truth tables and Karnaugh maps.

A final section explains how to use random numbers to run simulations, using a baseball batter's history as an example.

Work With UMAP

Following a 2-week curriculum development meeting in Massachusetts in 1975 [33], Sister Kenneth wrote several educational modules for the Undergraduate Mathematics and Its Applications Project (UMAP). "The goal of UMAP is to develop, through a community of users and developers, a system of instructional modules in undergraduate mathematics and its applications that may be used to supplement existing courses and from which complete courses may eventually be built" [22, facepage]. Sister Kenneth structured her contributions in pairs, one elementary and one more advanced. She included exercises, a sample test, answers, and computer programs. She aimed to make the modules attractive to students by choosing topics of interest.

Modules U105 and U109 [23] concerned food service management. In the elementary module, she showed how to compute the cost of various menu items from ingredient costs and recipes. In the advanced module, she invited the student to create a diet that met specified nutritional requirements.

Module U106 [22] concerned computer graphics. Sister Kenneth explained the process of rotating a ray in the plane through a specified angle by multiplication by a 2×2 rotation matrix. Then, she introduced homogeneous coordinates, three coordinates for each point in the plane, a common choice in computer graphics. Continuing this theme in Module U110 [22], Sister Kenneth invited students to determine how to rotate, translate, and scale points in 3-D using homogeneous coordinates.

Modules U107 and U111 [21] introduced Markov chains. Using an example of predicting student performance based on performance on a previous exam, Sister Kenneth introduced the transition matrix and its uses. In the advanced module, using an example of television viewing habits, she introduced the stationary vector of

[10] [Online]. Available: https://ascue.org/history/
[11] [Online]. Available: http://www.computer.org/volunteering/awards/cse-undergrad-teaching

the chain, the long-term probabilities of the model. Another example involving loan repayment prediction, which has absorbing states, motivated the fundamental matrix of the chain.

Module U108 [20] presented Kirchoff's laws and Ohm's law with a system of linear equations to analyze a (linear) electrical circuit. Module U112 [20] dealt with more complicated circuits.

SISTER KENNETH's PHILOSOPHY OF COMPUTING

In her unfinished book entitled, *The Computer: A Humanistic Approach*, Sister Kenneth viewed the computer as an "idea processor" for text and programs that should be used in schools as early as the fourth grade. She meant to provide a source of interesting elementary, intermediate, and advanced problems amenable to computer solution, especially with "pictorial and graphical representations for solutions."

Sister Kenneth was a visionary. Her colleagues were in awe of her, astounded by prescient predictions that wristwatches would become computers[12] and that computers would transform the arts and humanities. As early as 1966, she envisioned the use of computers in every academic discipline:

> We really do not know how people learn. For the first time, we can now mechanically simulate the cognitive process. We can make studies in artificial intelligence. Beyond that, this mechanism can be used to assist humans to learn. It reacts patiently, persistently. It can store the number of paths a student may take, and point out the successes....In the modern linguistic field, for instance, the whole science of language and grammar may be studied by this method [30].

She was also aware of the fears that jobs would be eliminated and automated decisions would be unquestioned, as well as the hopes of more meaningful work and "a super-planner, which can aid the world's growing population in production, protection, and decision. It is reasonable to stand with those who hope, but it must be with a sense of responsibility," since "man has not always used his inventions well" [25].

Perhaps her most complete exposition of her philosophy was presented in an undated address to the Cedar Rapids Conference on "The Role of the Computer in the High School." She pointed out the computer's potential as a teaching machine, librarian for information retrieval, "super-clerk" for reducing the monotony of administrative tasks, and linguist for foreign language translation.

> Of course it is a mathematician, but please do not think of it as just a special adding machine or even a super slide rule. It is just a historical accident that such connotation [is] attached. It is essentially a symbol manipulator.

She envisioned the computer improving education and freeing teachers to have "more human contact with your students and fellow teachers." Sister Kenneth gave an emphatic "yes" in answer to the question of whether computation should be taught in high school:

> Every citizen has a right and a duty to have a knowledge, commensurate with his capacity, of the important forces and instruments which shape his civilization. Not only, as I have said, would we be presenting a distorted, truncated view of subject areas [if computation were omitted], but we would be failing to capitalize on the ease of introducing basic concepts to young adults as compared to mature adults, and we would be denying many that knowledge altogether.

She believed that every high school student should be able to program a computer and to understand its internal binary logic system. Rather than separate courses in computing, Sister Kenneth envisioned a "mathematics laboratory" accessible "to every mathematics student regardless of level." On the practical side, she anticipated that the reaction of those in the audience might be "there are no funds, and that takes care of that." She noted, "This is a situation in which the experiences of life makes me a minor expert." To convince them, she had collected positive responses from potential donors in the Dubuque area. Regarding teacher training,

> The number of such courses that are available at many colleges is yearly increasing, and a single course is sufficient to bootstrap teachers into a program of self-education.... Education is, after all, a self-activity, and a patient but unrelenting electronic assistant can be a boon.

FINAL DAYS

Sister Kenneth was true to her mission to the end. A sign in her office read, "My life is a continuing changing awareness of God's will for me" [35]. Catherine Dunn, BVM remembers that even as Sister Kenneth was dying in a nursing home, she had a computer in her bedroom. She continued to work, plan nutritious meals [33], and give lessons to other sisters (Figure 7) [34]. Sister Kenneth did not want to die: she felt she still had work to do.

[12]Carmelle Zserdin, BVM, July 2, 2019.

Nevertheless, Bertha Fox, BVM, reports that her last words were, "Yes, yes." She died January 10, 1985.

Her gravestone at Mount Carmel, shown in Figure 8, is small, flat to the ground, and shared with another sister. It has her name and the three most important dates of her life: birth, entrance into the BVM congregation, and death.

CONCLUSION

Sister Kenneth was a totally remarkable woman. She made the most of every opportunity given to her, even her unlikely path to her late-in-life Ph.D. As a scholar, she has the distinction of being an early advocate of learning-by-example in artificial intelligence. Her main scholarly contribution was in shaping computer science education in high schools and small colleges. She was an evangelist for viewing the computer as a symbol manipulator, for providing computer literacy to everyone, and for the use of computers in service to humanity. She was far ahead of her time in working to ensure a place for women in technology and in eliminating barriers preventing their participation, such as poor access to education and daycare. She was a strong and spirited woman, a visionary in seeing how computers would revolutionize our lives.

FIGURE 7. Sister Kenneth (right) in April, 1984, demonstrating computing to BVM sisters Gladys Ramaley and Marian Delany. Photo Credit: Reprinted with permission of the *Dubuque Telegraph Herald*.

FIGURE 8. Shared gravestone at Mount Carmel, Dubuque, Iowa. Photo: T. J. O'Leary.

ACKNOWLEDGMENTS

The authors would like to thank BVM sisters Carol Blitgen, Mary Lou Caffery, Catherine Dunn, Bertha Fox, Diana Malone, Sara McAlpin, Sheila O'Brien, and Carmelle Zserdin, for sharing memories of Sister Kenneth. Ralph London was especially generous in providing insight. Thomas Byrne, Robert Hargraves, Arthur Luehrmann, Tracy Flynn Moloney, and especially Steven Garland, Anthony Knapp, and Thomas Kurtz helped to shed light on Sister Kenneth's connection to Dartmouth. Mary Tonoli Teister, Mary C. Beckman, BVM, Marilyn Ferdinand, David Hemmendinger, and Michael Steuerwalt also provided useful assistance. Peter J. Murdza Jr. searched Dartmouth archives for us, and Karen L. Barrett-Wilt did the same at the University of Wisconsin.

BIBLIOGRAPHY

[1] "Clarke College offers masters program," *Apple Educ. News*, vol. 3, no. 3, Oct.–Dec. 1982.

[2] R. H. Austing, B. H. Barnes, D. T. Bonnette, G. L. Engel, and G. Stokes, Eds., "Curriculum '78: Recommendations for the undergraduate program in computer science," *Commun. ACM*, vol. 22, no. 3, pp. 147–166, Mar. 1979.

[3] J. Byrne, "Expert urges computer familiarity," *Chicago Sun Times*, Jan. 1966.

[4] "Clarke College receives a $206,676 National Science Foundation grant," Clarke College Comput. Sci. Dept. Newsletter, Summer, 1980.

[5] "Sister Kenneth Keller's legacy," in *Clarke College On Campus*. Spring 1995.
[6] "Computer Sciences Department celebrates 25 years," *Clarke Comput. Sci. Bull.*, 1990.
[7] W. S. Dorn and H. J. Greenberg, *Mathematics and Computing With Fortran Programming*. Hoboken, NJ, USA: Wiley, 1967.
[8] W. S. Dorn, H. J. Greenberg, and Sr. M. K. Keller, *Mathematical Logic and Probabiity With BASIC Programming*. Boston, MA, USA: Prindle, Weber & Schmidt, Inc., 1973, 216 pages.
[9] "The nun and the computer," *Dubuque Telegraph Herald*, Oct. 1965.
[10] "Clarke unveils computer center," *Dubuque Telegraph Herald*, Sep. 1980.
[11] "Famed scientist learns new skill here," *Dubuque Telegraph Herald*, Sep. 1968.
[12] Educational Services Department, Fabri-Tek, "Computer education: An answer to the challenge!" Accessed: Jan. 25, 2020. [Online]. Available: https://archive.org/details/TNM_Computer_education_BI-TRAN_SIX_-_Fabri-Tek_20171010_0075/page/n15
[13] D. W. Gürer, "Pioneering women in computer science," *Commun. ACM*, vol. 38, no. 1, pp. 45–54, Jan. 1995.
[14] M. K. Keller, "Experience with various alternatives in undergraduate computing facilities," in *Proc. 1st Conf. Comput. Undergraduate Curricula*, 1970.
[15] M. K. Keller, "The professional development of student programmers through field experience," in *Proc. 7th Annu. CUETUG Meeting*, Greenville, SC, USA, 1974.
[16] M. K. Keller, "A business-oriented computer science curriculum," in *Proc. 10th Annu. ASCUE Meeting*, Reading, PA, USA, 1977.
[17] M. K. Keller, "Microcomputers as a versatile alternative to time sharing in education," in *Proc. 13th Annu. ASCUE Meeting*, 1980.
[18] M. K. Keller, "The role of a computer science department in the use of the computer in undergraduate curricula at a small liberal arts college," *Proc. Conf. Comput. Undergraduate Curricula, Claremont, CA*, 1973, 6 pages.
[19] Sr. M. K. Keller, "The role of the university computer center in educational research (with remote hope)," *ACM SIGUUC Newslett. (Special Int. Group Univ. Comput. Centers)*, vol. 4, pp. 12–14, 1974.
[20] Sr. M. K. Keller, *Electrical Circuits (Unit 108) and Applications of Matrix Methods: Analysis of Linear Circuits (Unit 112)*. 1978, 34 pages.
[21] Sr. M. K. Keller, *Markov Chains (Unit 107) and Applications of Matrix Methods: Fixed Point and Absorbing Markov Chains (Unit 111)*. 1981, 33 pages, In [32].
[22] Sr. M. K. Keller, *Computer Graphics (Unit 106) and Applications of Matrix Methods: Three Dimensional Computer Graphics and Projections (Unit 110)*. 1983, 43 pages, In [32].
[23] Sr. M. K. Keller, *Food Service Management (Unit 105) and Applications of Matrix Methods: Food Service and Dietary Requirements (Unit 109)*. 1983, p. 27, In [32].
[24] Sr. M. K. Keller, BVM, "Inductive inference on computer generated patterns," Ph.D. thesis, Comput. Sci., Univ. Wisconsin, Madison, WI, USA, Jun. 1965.
[25] Sr. M. Kenneth, "Man and machine–in that order," *BVM Vista*, pp. 24–25, Dec. 1964.
[26] R. London, Additional Information for Who Earned First Computer Science Ph.D.? Dec. 2012. Cited in [27] .
[27] R. London, *Who earned first Computer Science Ph.D.?* Jan. 2013. [Online]. Available: https://cacm.acm.org/blogs/blog-cacm/159591-who-earned-first-computer-science-ph-d/fulltext
[28] R. A. Martin, *Syntactical Evidence of Semetic Sources in Greek Documents*. Society of Biblical Literature, 1974. Reprinted by Wipf and Stock Publishers, Eugene, OR, USA, 2004.
[29] Sr. M. Michail, BVM, "Clarke computers, cybernetics," *BVM Vista*, pp. 26–28, Mar. 1966.
[30] "Needed: Women to run computers," *Midwest Magazine, The Chicago Sun Times*, Feb. 1966.
[31] J. Ney, "It's back to school for genius, 73," *De Moines Morning Register*, Sep. 1968.
[32] Sr. M. K. Keller, *UMAP Modules and Monographs in Undergraduate Mathematics and Its Applications Project*. Bedford, MA, USA: Consortium for Mathematics and Its Applications, 2022. [Online]. Available: www.comap.com
[33] D. Walsh, BVM, "Kenneth Keller, BVM, 1913-85 still being quoted," *BVM Newslett.*, vol. 39, no. 1, Oct./Nov. 2003.
[34] K. Walsh, "Older nuns learn new tricks on computers," *Dubuque Telegraph Herald*, Apr. 1984.
[35] "I went to look at a computer one day," *The Witness*, Nov. 1975.

JENNIFER HEAD is the archivist for the Sisters of Charity of the Blessed Virgin Mary in Dubuque, IA, 52003, USA. A Certified Archivist, she holds master's degrees in American History from the Catholic University of America and in Library and Information Science from Dominican University. Contact her at jhead@bvmsisters.org.

DIANNE P. O'LEARY is a distinguished university professor, emerita at the University of Maryland, College Park, MD, 20742, USA. O'Leary received the Ph.D. degree from Stanford University. She is a member of AWM and a Fellow of SIAM and ACM. Contact her at oleary@umd.edu.

IEEE COMPUTER SOCIETY

Publications Seek 2025 Editors in Chief

Application Deadline: 1 March 2024

The IEEE Computer Society seeks applicants for the position of editor in chief for the following publications:

- *Computer* magazine
- *Computing in Science and Engineering* magazine
- *IEEE/ACM Transactions on Computational Biology and Bioinformatics*
- *IEEE Computer Architecture Letters*
- *IEEE Intelligent Systems* magazine
- *IEEE Open Journal of the Computer Society*
- *IEEE Transactions on Big Data*
- *IEEE Transactions on Cloud Computing*
- *IEEE Transactions on Sustainable Computing*

Computer Society publications are the cornerstone of professional activities for our members and the community we serve. We seek candidates who are IEEE members in good standing, have strong familiarity with our publications, and possess an excellent understanding of the field as it relates to academic, industry, and governmental areas. Applicants must also have successful experience attracting and developing a diverse team of talented and respected individuals to serve key editorial board roles. Demonstrated managerial skills are also required, as they are necessary to ensure rich content and issue development, along with timely processing of submissions through the editorial cycle. Terms begin 1 January 2025.

For complete information on how to apply, please go to
www.computer.org/press-room/seeking-2025-editors-in-chief

Apply Today!

IEEE COMPUTER SOCIETY

IEEE

Get Published in the New *IEEE Open Journal of the Computer Society*

Submit a paper to the new IEEE Open Journal of the Computer Society covering computing and informational technology.

Your research will benefit from the IEEE marketing launch and 5 million unique monthly users of the IEEE *Xplore*® Digital Library. Plus, this journal is fully open and compliant with funder mandates, including Plan S.

Submit your paper today!
Visit www.computer.org/oj to learn more.

IEEE COMPUTER SOCIETY | IEEE

SUBMIT TODAY

IEEE TRANSACTIONS ON

SUSTAINABLE COMPUTING

SCOPE

The *IEEE Transactions on Sustainable Computing* (*T-SUSC*) is a peer-reviewed journal devoted to publishing high-quality papers that explore the different aspects of sustainable computing. The notion of sustainability is one of the core areas in computing today and can cover a wide range of problem domains and technologies ranging from software to hardware designs to application domains. Sustainability (e.g., energy efficiency, natural resources preservation, using multiple energy sources) is needed in computing devices and infrastructure and has grown to be a major limitation to usability and performance.

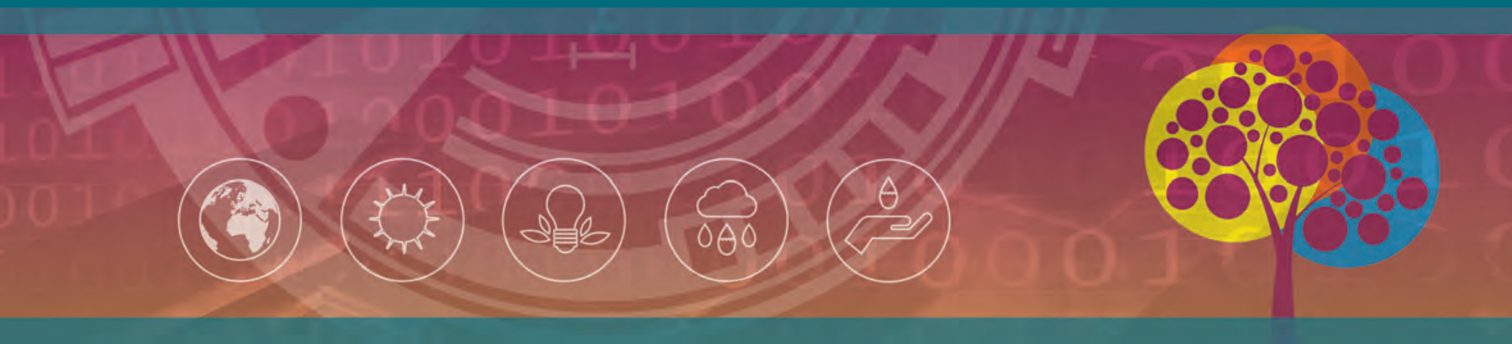
Contributions to *T-SUSC* must address sustainability problems in different computing and information processing environments and technologies, and at different levels of the computational process. These problems can be related to information processing, integration, utilization, aggregation, and generation. Solutions for these problems can call upon a wide range of algorithmic and computational frameworks, such as optimization, machine learning, dynamical systems, prediction and control, decision support systems, meta-heuristics, and game-theory to name a few.

T-SUSC covers pure research and applications within novel scope related to sustainable computing, such as computational devices, storage organization, data transfer, software and information processing, and efficient algorithmic information distribution/processing. Articles dealing with hardware/software implementations, new architectures, modeling and simulation, mathematical models and designs that target sustainable computing problems are encouraged.

SUBSCRIBE AND SUBMIT

For more information on paper submission, featured articles, calls for papers, and subscription links visit:

www.computer.org/tsusc



IEEE COMPUTER SOCIETY

IEEE ComSoc IEEE Communications Society

CEDA IEEE Council on Electronic Design Automation

IEEE

IEEE

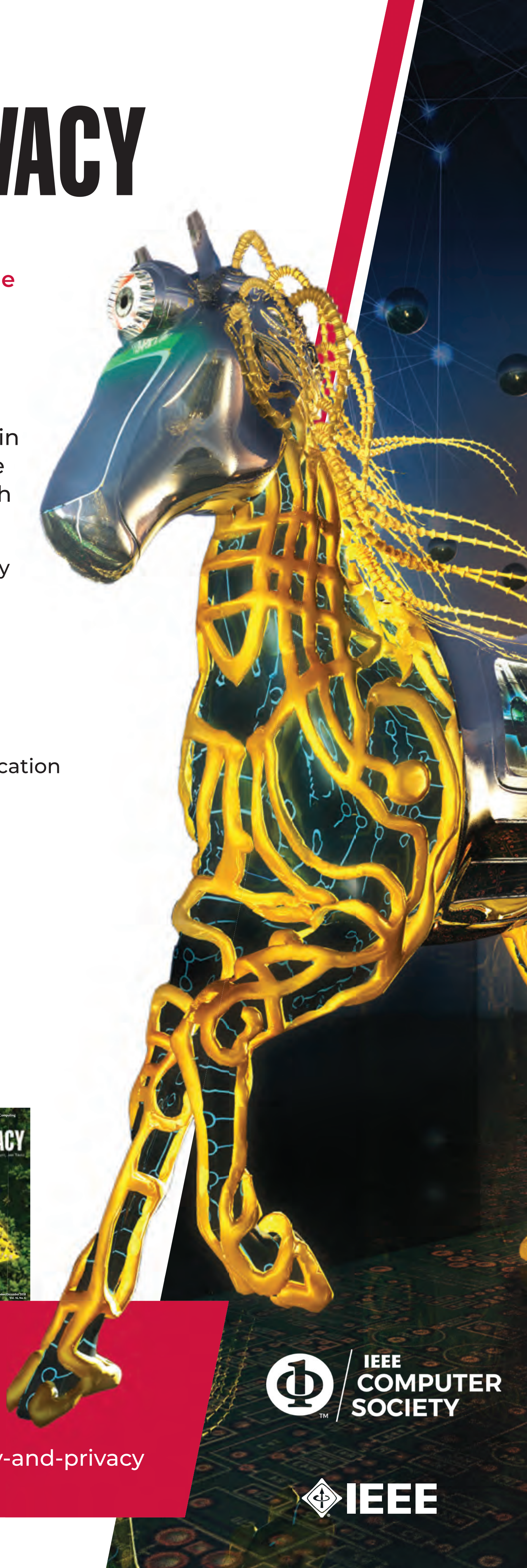
SECURITY & PRIVACY

***IEEE Security & Privacy* is a bimonthly magazine communicating advances in security, privacy, and dependability** in a way that is useful to a broad section of the professional community.

The magazine provides articles with both a practical and research bent by the top thinkers in the field of security and privacy, along with case studies, surveys, tutorials, columns, and in-depth interviews. Topics include:

- Internet, software, hardware, and systems security
- Legal and ethical issues and privacy concerns
- Privacy-enhancing technologies
- Data analytics for security and privacy
- Usable security
- Integrated security design methods
- Security of critical infrastructures
- Pedagogical and curricular issues in security education
- Security issues in wireless and mobile networks
- Real-world cryptography
- Emerging technologies, operational resilience, and edge computing
- Cybercrime and forensics, and much more

www.computer.org/security



Join the IEEE Computer Society for subscription discounts today!

www.computer.org/product/magazines/security-and-privacy

IEEE COMPUTER SOCIETY

IEEE

IEEE

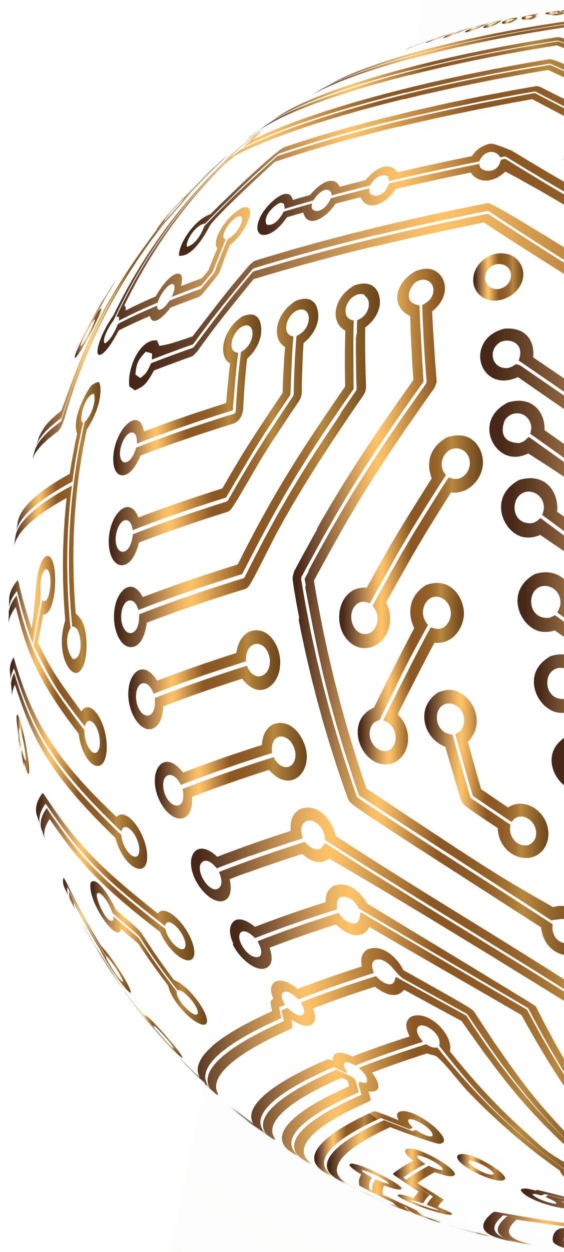
COMPUTER ARCHITECTURE LETTERS

***IEEE Computer Architecture Letters* is a forum for fast publication of new, high-quality ideas in the form of short, critically refereed technical papers.** Submissions are accepted on a continuing basis and letters will be published shortly after acceptance in IEEE Xplore and in the Computer Society Digital Library.

Submissions are welcomed on any topic in computer architecture, especially:

- Microprocessor and multiprocessor systems
- Microarchitecture and ILP processors
- Workload characterization
- Performance evaluation and simulation techniques
- Interactions with compilers and operating systems
- Interconnection network architectures
- Memory and cache systems
- Power and thermal issues at the architectural level
- I/O architectures and techniques
- Independent validation of previously published results
- Analysis of unsuccessful techniques
- Domain-specific processor architecture (embedded, graphics, network)
- High-availability architectures
- Reconfigurable computer architectures

www.computer.org/cal



Join the IEEE Computer Society for subscription discounts today!

www.computer.org/product/journals/cal

IEEE COMPUTER SOCIETY

IEEE

IEEE TRANSACTIONS ON

BIG DATA

IEEE Transactions on Big Data **is a quarterly journal that publishes peer-reviewed articles with big data as the main focus.**

The articles provide cross-disciplinary, innovative research ideas and applications results for big data including novel theory, algorithms, and applications. Research areas include:

- Big data
    - Analytics
    - Curation and management
    - Infrastructure
    - Performance analyses
    - Semantics
    - Standards
    - Visualization
- Intelligence and scientific discovery from big data
- Security, privacy, and legal issues specific to big data

Applications of big data in the fields of endeavor where massive data is generated are of particular interest.

www.computer.org/tbd

Join the IEEE Computer Society for subscription discounts today!

www.computer.org/product/journals/tbd

IEEE COMPUTER SOCIETY

IEEE

Conference Calendar

IEEE Computer Society conferences are valuable forums for learning on broad and dynamically shifting topics from within the computing profession. With over 200 conferences featuring leading experts and thought leaders, we have an event that is right for you. Questions? Contact conferences@computer.org.

MARCH

2 March

- CGO (IEEE/ACM Int'l Symposium on Code Generation and Optimization), Edinburgh, UK
- HPCA (IEEE Int'l Symposium on High-Performance Computer Architecture), Edinburgh, UK

11 March

- PerCom (IEEE Int'l Conf. on Pervasive Computing and Communications), Biarritz, France

12 March

- SANER (IEEE Int'l Conf. on Software Analysis, Evolution and Reengineering), Rovaniemi, Finland

16 March

- VR (IEEE Conf. on Virtual Reality and 3D User Interfaces), Orlando, USA

17 March

- SSIAI (IEEE Southwest Symposium on Image Analysis and Interpretation), Santa Fe, New Mexico, USA

APRIL

9 April

- SaTML (IEEE Conf. on Secure and Trustworthy Machine Learning), Toronto, Canada

17 April

- COOL CHIPS (IEEE Symposium in Low-Power and High-Speed Chips), Tokyo, Japan

22 April

- VTS (IEEE VLSI Test Symposium), Tempe, Arizona, USA

23 April

- PacificVIS (IEEE Pacific Visualization Symposium), Tokyo, Japan

29 April

- DCOSS-IoT (Int'l Conf. on Distributed Computing in Smart Systems and the Internet of Things), Abu Dhabi, United Arab Emirates

MAY

1 May

- MOST (IEEE Int'l Conf. on Mobility, Operations, Services and Technologies), Dallas, USA

5 May

- ISPASS (IEEE Int'l Symposium on Performance Analysis of Systems and Software), Indianapolis, USA

6 May

- FCCM (IEEE Int'l Symposium on Field-Programmable Custom Computing Machines), Orlando, USA
- HOST (IEEE Int'l Symposium on Hardware Oriented Security and Trust), Tysons Corner, Virginia, USA
- ICFEC (IEEE Int'l Conf. on Fog and Edge Computing), Philadelphia, USA

13 May

- CCGrid (IEEE Int'l Symposium on Cluster, Cloud and Internet Computing), Philadelphia, USA
- ICCPS (ACM/IEEE Int'l Conf. on Cyber-Physical Systems), Hong Kong
- ICDE (IEEE Int'l Conf. on Data Eng.), Utrecht, The Netherlands
- RTAS (IEEE Real-Time and Embedded Technology and Applications Symposium), Hong Kong

20 May

- SP (IEEE Symposium on Security and Privacy), San Francisco, USA

21 May

- ISORC (IEEE Int'l Symposium on Real-Time Distributed Computing), Tunis, Tunisia

27 May

- FG (IEEE Int'l Conf. on Automatic Face and Gesture Recognition), Istanbul, Turkey
- ICST (IEEE Conf. on Software Testing, Verification and Validation), Toronto, Canada
- IPDPS (IEEE Int'l Parallel and Distributed Processing Symposium), San Francisco, USA

28 May

- ISMVL (IEEE Int'l Symposium

 2469-7087/24 © 2024 IEEE

on Multiple-Valued Logic), Brno, Czech Republic

JUNE

3 June

- ICHI (IEEE Int'l Conf. on Healthcare Informatics), Orlando, USA

4 June

- ICSA (IEEE Int'l Conf. on Software Architecture), Hyderabad, India
- WoWMoM (IEEE Int'l Symposium on a World of Wireless, Mobile and Multimedia Networks), Perth, Australia

10 June

- ARITH (IEEE Symposium on Computer Arithmetic), Malaga, Spain

16 June

- CVPR (IEEE/CVF Conf. on Computer Vision and Pattern Recognition), Seattle, USA

17 June

- SVCC (Silicon Valley Cybersecurity Conf.), Seoul, South Korea

19 June

- CHASE (IEEE/ACM Conf. on Connected Health: Applications, Systems and Eng. Technologies), Wilmington, USA

24 June

- DSN (IEEE/IFIP Int'l Conf. on Dependable Systems and Networks), Brisbane, Australia
- MDM (IEEE Int'l Conf. on Mobile Data Management), Brussels, Belgium
- RE (IEEE Int'l Requirements Eng. Conf.), Reykjavik, Iceland

25 June

- CAI (IEEE Conf. on Artificial Intelligence), Singapore

26 June

- CBMS (IEEE Int'l Symposium on Computer-Based Medical Systems), Guadalajara, Mexico

27 June

- CS (IEEE Cloud Summit), Washington, DC, USA (Hybrid)

29 June

- ISCA (ACM/IEEE Annual Int'l Symposium on Computer Architecture), Buenos Aires, Argentina

JULY

1 July

- COMPSAC (IEEE Annual Computers, Software, and Applications Conf.), Osaka, Japan
- ICALT (IEEE Int'l Conf. on Advanced Learning Technologies), Nicosia, Cyprus

3 July

- IOLTS (IEEE Int'l Symposium on On-Line Testing and Robust System Design), Rennes, France

7 July

- SERVICES (IEEE World Congress on Services), Shenzhen, China

8 July

- CSF (IEEE Computer Security Foundations Symposium), Enschede, Netherlands
- EuroS&P (IEEE European Symposium on Security and Privacy), Vienna, Austria

15 July

- ICME (IEEE Int'l Conf. on Multimedia and Expo), Niagara Falls, Canada
- SCC (IEEE Space Computing Conf.), Mountain View, USA
- SMC-IT (IEEE Int'l Conf. on Space Mission Challenges for Information Technology), Mountain View, USA

16 July

- ICDCS (IEEE Int'l Conf. on Distributed Computing Systems), Jersey City, USA

22 July

- ICCP (IEEE Int'l Conf. on Computational Photography), Lausanne, Switzerland

24 July

- ASAP (IEEE Int'l Conf. on Application-specific Systems, Architectures and Processors), Hong Kong

AUGUST

25 Aug

- HCS (IEEE Hot Chips Symposium), Stanford, USA

27 Aug

- SustainTech (IEEE SustainTech Expo: Technology Solutions for a Sustainable Future), San Diego, USA

IEEE COMPUTER SOCIETY CAREER CENTER

Career Accelerating Opportunities

Explore new options—upload your resume today

careers.computer.org

Changes in the marketplace shift demands for vital skills and talent. The IEEE Computer Society Career Center is a valuable resource tool to keep job seekers up to date on the dynamic career opportunities offered by employers.

Take advantage of these special resources for job seekers:

- JOB ALERTS
- TEMPLATES
- WEBINARS
- CAREER ADVICE
- RESUMES VIEWED BY TOP EMPLOYERS

No matter what your career level, the IEEE Computer Society Career Center keeps you connected to workplace trends and exciting career prospects.

IEEE COMPUTER SOCIETY

IEEE