

Computing Curricula 2020

CC2020

Paradigms for Future Computing Curricula

encompassing undergraduate programs in

Computer Engineering
Computer Science
Cybersecurity
Data Science
Information Systems
Information Technology
Software Engineering

A Computing Curricula Series Report

Association for Computing Machinery (ACM)
IEEE Computer Society (IEEE-CS)

Version 36 | 2020 April 29

30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68

Copyright © 2020 by ACM and IEEE

ALL RIGHTS RESERVED

Copyright and Reprint Permissions: Permission granted to use these curriculum guidelines for the development of educational materials and programs. Other use requires specific permission. Permission requests should be addressed to: ACM Permissions Dept. at permissions@acm.org.

ISBN: 000-0-0000-0000-0

DOI: 10.0000/0000000

Web link: <https://dl.acm.org/citation.cfm?id=0000000>

When available, you may order additional copies from:
ACM Order Department, P.O. Box 30777, New York, NY 10087-0777

Sponsoring Societies

This report was possible by financial support from:

Association for Computing Machinery (ACM)

IEEE Computer Society (IEEE-CS)

and the

ACM Europe

Association for Information Systems (AIS)

Education SIG of the Association for Information Technology Professionals (ED-SIG)

GRIN (Italian Association of Computer Scientists)

Informatics Europe

Special Interest Group on Computer Human Interaction of (SIGCHI)

The following organizations have officially endorsed this report:

The Latin American Center on Computing (CLEI)

69
70
71
72
73
74
75
76
77
78

CC2020 Task Force

Co-Chairs

Alison Clear
Eastern Institute of Technology
New Zealand

Allen Parrish
University of Alabama
United States of America

79
80
81
82
83
84

Steering Committee

Paolo Ciancarini
University of Bologna
Italy

Ernesto Cuadros-Vargas
Universidad de Ingeniería y Tecnología
Peru

Stephen Frezza
Gannon University
United States of America

Judith Gal-Ezer
Open University
Israel

John Impagliazzo
Hofstra University
United States of America

Arnold Pears
KTH Royal Institute of Technology
Sweden

Shingo Takada
Keio University
Japan

Heikki Topi
Bentley University
United States of America

Gerrit van der Veer
Vrije Universiteit
Netherlands

Abhijat Vichare
ACM India
India

Les Waguespack
Bentley University
United States of America

Pearl Wang
George Mason University
United States of America

Ming Zhang
Peking University
China

85
86
87
88
89
90
91
92
93

Other Task Force Members

(Next Page)

94
95
96
97

CC2020 Task Force (continued)

Hala Alrumaih
Al Imam Mohammad Ibn Saud Islamic Univ
Saudi Arabia

Renata Araujo
Brazilian Computer Society
Brazil

Jeffrey Babb
West Texas A&M University
United States of America

Olga Bogoyavlenskaya
Petrozavodsk State University
Russia

Héctor Cancela
Universidad de la República
Uruguay

Juan Chen
National University of Defense Technology
China

Adrienne Decker
University of Buffalo
United States of America

Eric Durant
Milwaukee School of Engineering
United States of America

Marisa Exter
Purdue University
United States of America

Beatriz Florián Gaviria
Universidad del Valle
Colombia

Itana Maria Gimenes
Brazilian Computer Society
Brazil

Steven Gordon
Ohio State University
United States of America

Eiji Hayashiguchi
Waseda University, CTO VJP Co. Ltd.
Japan

Amey Karkare
India Institute of Technology – Kanpur
India

Richard Le Blanc
Seattle University
United States of America

Paul Leidig
Grand Valley State University
United States of America

David Lopez
Universitat Politècnica de Catalunya
Spain

Barry Lunt
Brigham Young University
United States of America

Linda Marshall
University of Pretoria
South Africa

Bruce McMillin
Missouri University of Science and
Technology
United States of America

Tania McVeety
IBM
United States of America

Nancy Mead
Carnegie Mellon University
United States of America

Teresa Pereira
Instituto Politécnico de Viana Castelo
Portugal

Melinda Reno
Deloitte Consulting
United States of America

Ariel Sabiguero
Universidad de la República
Uruguay

Fermin Sanchez
Universitat Politècnica de Catalunya
Spain

Yann Secq
University of Lille
France

Nello Scarabottolo
Università di Milano
Italy

Simon
University of Newcastle
Australia

Ye Tian
ByteDance
China

Paul Tymann
Rochester Institute of Technology
United States of America

Barbara Viola
Viotech Solutions
United States of America

Xi Wu
Chengdu Univ. of Information Technology
China

Jane (Xiaochun) Yang
AchieveFun Info Technology
China

Stuart Zweben
Ohio State University
United States of America

98

99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146

Contents

CC2020 TASK FORCE	3
CONTENTS	5
EXECUTIVE SUMMARY	11
CHAPTER 1: INTRODUCING CC2020	14
1.1: CC2020 EXPECTATIONS	14
<i>1.1.1: Project Purpose, Vision and Mission</i>	<i>14</i>
<i>1.1.2: Project Strategies</i>	<i>15</i>
<i>1.1.3: Project Diversity</i>	<i>15</i>
1.2: PROJECT STAKEHOLDERS	16
<i>1.2.1: Prospective Students and their Guardians</i>	<i>16</i>
<i>1.2.2: Current Students</i>	<i>16</i>
<i>1.2.3: Industry</i>	<i>16</i>
<i>1.2.4: Computing Educators</i>	<i>17</i>
<i>1.2.5: Educational Organizations and Authorities</i>	<i>17</i>
1.3: PROJECT BACKGROUND	17
<i>1.3.1: Brief History</i>	<i>17</i>
<i>1.3.2: Project Organization and Structure</i>	<i>18</i>
1.4: OVERALL SCOPE OF COMPUTING	18
<i>1.4.1: Current Discipline Structure</i>	<i>18</i>
<i>1.4.2: Timeline of Curricular Guidelines</i>	<i>19</i>
1.5: GUIDING PRINCIPLES	20
<i>1.5.1: Four Principles</i>	<i>20</i>
<i>1.5.2: Constituent and Public Outreach</i>	<i>21</i>
1.6: CC2020 REPORT STRUCTURE	22
1.7: DIGEST OF CHAPTER 1	22
CHAPTER 2: EVOLUTION OF COMPUTING EDUCATION	23
2.1: WHAT IS COMPUTING?	23
<i>2.1.1: Early Meanings</i>	<i>23</i>
<i>2.1.2: Recent Undertakings</i>	<i>23</i>
2.2: LANDSCAPE OF COMPUTING DISCIPLINES	24
<i>2.2.1: Early Developments</i>	<i>24</i>
<i>2.2.2: Contemporary Advances</i>	<i>25</i>
2.3: STATUS OF COMPUTING DISCIPLINE REPORTS	25
<i>2.3.1: Computer Engineering</i>	<i>26</i>
<i>2.3.2: Computer Science</i>	<i>26</i>
<i>2.3.3: Cybersecurity</i>	<i>27</i>
<i>2.3.4: Data Science</i>	<i>27</i>
<i>2.3.5: Information Systems</i>	<i>27</i>
<i>2.3.6: Information Technology</i>	<i>28</i>
<i>2.3.7: Software Engineering</i>	<i>28</i>
2.4: EXTENSIONS OF COMPUTING DISCIPLINES	29
<i>2.4.1: Computing Interrelationships</i>	<i>29</i>
<i>2.4.1: Emerging Curricula</i>	<i>30</i>

147	2.4.2: <i>Computing + X</i>	30
148	2.4.3: <i>X + Computing</i>	31
149	2.4.4: <i>Other Tertiary Computing Models</i>	31
150	2.4.5: <i>Computing in Primary and Secondary Education</i>	32
151	2.4.6: <i>Computing Specialization</i>	33
152	2.5: DIGEST OF CHAPTER 2.....	33
153	CHAPTER 3: KNOWLEDGE-BASED COMPUTING EDUCATION.....	34
154	3.1: KNOWLEDGE-BASED LEARNING	34
155	3.1.1: <i>Learning and Knowledge</i>	34
156	3.1.2: <i>Learning from Knowledge Contexts</i>	34
157	3.1.3: <i>Knowledge and Computing Education</i>	35
158	3.2: COMPUTING CURRICULA 2005	35
159	3.2.1: <i>Intent of CC2005</i>	35
160	3.2.2: <i>Content of CC2005</i>	36
161	3.2.3: <i>Comparison Tables</i>	36
162	3.2.4: <i>Curricular Visuals</i>	37
163	3.2.5: <i>Global and Other Considerations</i>	40
164	3.3: DYNAMICS AND COMPUTING DEGREE PROGRAMS.....	40
165	3.3.1: <i>Recent Workplace Trends</i>	40
166	3.3.2: <i>Forces Shaping Employment Markets</i>	41
167	3.5: DIGEST OF CHAPTER 3	42
168	CHAPTER 4: COMPETENCY-BASED COMPUTING EDUCATION	43
169	4.1: LIMITATIONS OF KNOWLEDGE-BASED CURRICULUM SPECIFICATIONS	43
170	4.1.1: <i>Articulating a Need</i>	43
171	4.1.2: <i>The Skills Gap</i>	43
172	4.1.3: <i>Moving Toward Competency</i>	44
173	4.2: COMPETENCY AND COMPETENCY-BASED LEARNING	44
174	4.2.1: <i>Competency and its Meaning</i>	44
175	4.2.2: <i>Previous Work on Computing Competency</i>	45
176	4.2.3: <i>Initial and Developing CC2020 Explorations of Competencies</i>	46
177	4.3: A COMPETENCY MODEL TO SUPPORT AUTOMATED VISUALIZATION AND ANALYSIS.....	47
178	4.3.1: <i>The CC2020 Competency Model</i>	47
179	4.3.2: <i>Component Definitions</i>	47
180	4.3.3: <i>Competency Statements</i>	48
181	4.3.4: <i>Component Elements</i>	49
182	4.4: FROM COMPETENCIES TO CURRICULA.....	50
183	4.4.1: <i>Identifying and authoring competencies</i>	50
184	4.4.2: <i>Competency Specifications and Curricular Specifications</i>	51
185	4.5: PRACTICAL APPLICATIONS OF COMPETENCY	52
186	4.5.1: <i>Competency Targets</i>	52
187	4.5.2: <i>Target Proximity</i>	52
188	4.5.3: <i>Consequences of Target Proximity</i>	53
189	4.6: DIGEST OF CHAPTER 4.....	53
190	CHAPTER 5: VISUALIZATION AND COMPETENCY SPECIFICATION	55
191	5.1: ON VISUALIZATION.....	55
192	5.1.1: <i>Some Basic Functions</i>	55
193	5.1.2: <i>Stakeholders and Visualization</i>	55
194	5.2: COMPETENCY AS A BASIS FOR VISUALIZATION.....	56
195	5.2.1: <i>Data and Use</i>	56
196	5.2.2: <i>Visualization and Competency Specification</i>	56

197	5.3: VISUALIZATION EXAMPLES OF STAKEHOLDER QUERIES	57
198	5.3.1: Students	57
199	5.3.2: Industry	57
200	5.3.3: Computing Educator	58
201	5.3.4: Educational Authority	59
202	5.4: OTHER RELATED VISUALIZATIONS	59
203	5.4.1: Visualization of a full curriculum	59
204	5.4.2: Visualization of the Knowledge table	61
205	5.4.3: Visualization of Competency	61
206	5.5: CHALLENGES CONCERNING VISUALIZATION	62
207	5.5.1: Consistent Vocabulary	62
208	5.5.2: Entity Comparison	63
209	5.5.3: Visualization types	63
210	5.6: DIGEST OF CHAPTER 5	63
211	CHAPTER 6: GLOBAL CONSIDERATIONS	64
212	6.1: GLOBAL CONTEXT AND COMPUTING PROGRAMS	64
213	6.1.1: Computing ‘Tower of Babel’	64
214	6.1.2: Computing Nomenclature	65
215	6.1.3: Cultural Sensitivity and Diversity	65
216	6.2: GLOBAL ECONOMICS AND COMPUTING EDUCATION	66
217	6.2.1: Emerging and New Technologies	66
218	6.2.2: Forces Shaping Academic Programs	67
219	6.2.3: Degree Names and the Workplace	67
220	6.2.4: Innovation in Computing	67
221	6.2.5: Entrepreneurship in Computing	68
222	6.3: WORLDWIDE COMPUTING DEGREE STRUCTURES	68
223	6.3.1: Computing Education in Africa	68
224	6.3.2: Computing Education in Australasia	69
225	6.3.3: Computing Education in China	69
226	6.3.4: Computing Education in Europe	69
227	6.3.5: Computing Education in India	70
228	6.3.6: Computing Education in Japan	70
229	6.3.7: Computing Education in the Middle East	71
230	6.3.8: Computing Education in North Americas	71
231	6.3.9: Computing Education in South Americas	72
232	6.4: PROFESSIONALISM AND ETHICS	72
233	6.4.1: Ethics in the Curriculum	72
234	6.4.2: Professional and Ethical Work	73
235	6.5: DIGEST OF CHAPTER 6	73
236	CHAPTER 7: CURRICULAR DESIGN – CHALLENGES AND OPPORTUNITIES	74
237	7.1: TRANSFORMING TO COMPETENCIES	74
238	7.1.1: Distinguishing Competency from Knowledge	74
239	7.1.2: Curricular Dynamics	74
240	7.1.3: Transferring Computing Competencies	75
241	7.1.4: Knowledge Transfer	75
242	7.1.5: Skill Transfer	75
243	7.1.6: Disposition Transfer	75
244	7.1.7: Need for Local Adaptation	76
245	7.2: INDUSTRY ENGAGEMENT FOR WORKPLACE COMPETENCIES	76
246	7.2.1: Professional Advisory Boards	77
247	7.2.2: Work-Study and Cooperative Programs	77

248	7.2.3: <i>Internship Programs</i>	78
249	7.3: INSTITUTIONAL ADAPTATIONS	78
250	7.3.1: <i>Attracting and Retaining Academic Educators</i>	78
251	7.3.2: <i>Need for Adequate Laboratory Resources</i>	78
252	7.3.3: <i>Transfer and Educational Pathways in the United States</i>	79
253	7.4: PROGRAM QUALITY ASSURANCE AND ACCREDITATION	80
254	7.4.1: <i>Accreditation Overview</i>	80
255	7.4.2: <i>Benefits of Program-Specific Accreditation</i>	80
256	7.4.3: <i>Quality Assurance</i>	81
257	7.4.4: <i>Global Recognition</i>	81
258	7.5: DEGREE NAMES AND JOB TITLES	82
259	7.5.1: <i>Job Positions and Titles</i>	82
260	7.5.2: <i>Use of the Word “Engineer”</i>	82
261	7.6: TECHNOLOGY TRENDS FOR CC2020	83
262	7.6.1: <i>Current and Emerging Technologies</i>	83
263	7.6.2: <i>Existing Computing Areas with No Formal Curriculum</i>	83
264	7.6.3: <i>Emerging Computing Areas</i>	84
265	7.7: DIGEST OF CHAPTER 7	85
266	CHAPTER 8: BEYOND CC2020	86
267	8.1: PUBLIC ENGAGEMENT AND THE CC2020 PROJECT	86
268	8.1.1: <i>CC2020 Project Website</i>	86
269	8.1.2: <i>Relating Curricula and Competencies</i>	86
270	8.1.3: <i>Project Dissemination</i>	86
271	8.2: COMPETENCY IN FUTURE CURRICULAR GUIDELINES	87
272	8.2.1: <i>Recent Curricular Development</i>	87
273	8.2.2: <i>Future Curricular Development</i>	87
274	8.3: COMPETENCY ADVOCACY	87
275	8.4: FUTURE ACTIVITIES	88
276	8.5: DIGEST OF CHAPTER 8	88
277	APPENDIX A: POSTER EXPLAINING CC2005 CURRICULAR VISUALS	89
278	APPENDIX B: COMPUTING SKILLS FRAMEWORKS	90
279	B.1: SKILLS FRAMEWORK FOR THE INFORMATION AGE	90
280	B.2: SKILLS AND THE EUROPEAN COMPETENCY FRAMEWORK	91
281	B.3: SKILLS AND THE I COMPETENCY DICTIONARY	93
282	B.3.1: <i>Task Dictionary</i>	94
283	B.3.2: <i>Task Dictionary Chart</i>	94
284	B.3.3: <i>Examples of Task Evaluation Diagnostic Level and Criteria</i>	95
285	B.3.4: <i>Skill Dictionary</i>	96
286	B.3.5: <i>Skill Dictionary Chart</i>	96
287	B.3.6: <i>Skill Proficiency Level</i>	97
288	B.4: SKILLS VIA ENTERPRISE INFORMATION TECHNOLOGY	98
289	B.5: REFERENCES FOR APPENDIX B	98
290	APPENDIX C: EXAMPLES OF COMPETENCIES	99
291	C.1: INITIAL CC2020 EXPLORATIONS OF COMPETENCIES	99
292	C.1.1: <i>Computer Engineering Draft Competencies</i>	100
293	C.1.2: <i>Computer Science Draft Competencies</i>	106
294	C.1.3: <i>Information Systems Draft Competencies</i>	111
295	C.1.4: <i>Information Technology Competencies</i>	115
296	C.1.5: <i>Software Engineering Draft Competencies</i>	118

297	<i>C.1.6: Master’s in Information Systems Draft Competencies</i>	122
298	C.2: REFERENCES FOR APPENDIX C	128
299	APPENDIX D: COMPETENCY-BASED COMPUTING CURRICULA	129
300	D.1: COMPETENCY IN COMPUTING BACCALAUREATE EDUCATION	129
301	D.2: THE CC2020 DEFINITION OF COMPETENCY	130
302	D.3: THE ANATOMY OF COMPETENCY SPECIFICATION	131
303	<i>D.3.1: The Competency Statement’s Role in a Competency Specification</i>	133
304	<i>D.3.2: Knowledge, “Knowing What,” as a Component of Competency</i>	133
305	<i>D.3.3: Skills, “Knowing How,” as Components of Competency</i>	138
306	<i>D.3.4: Dispositions, “Knowing Why,” as a Component of Competency</i>	139
307	D.4: STRUCTURING COMPETENCY STATEMENTS FOR COMPETENCY SPECIFICATION	140
308	<i>D.4.1: Developing Competency Statements and Specifications</i>	140
309	<i>D.4.2: Elaborating Competency Statements</i>	141
310	D.5: COMPETENCY IN COMPUTING EDUCATION	145
311	D.6: COMPETENCY IN FUTURE CURRICULAR GUIDELINES	146
312	D.7: SUMMARY	147
313	D.8: REFERENCES FOR APPENDIX D	147
314	APPENDIX E: FROM COMPETENCIES TO CURRICULA	149
315	E.1: COMPETENCY IN FUTURE CURRICULAR GUIDELINES	149
316	<i>E.1.1: Stakeholders</i>	149
317	<i>E.1.2: Competency targets</i>	150
318	<i>E.1.3: Outcome Expectations and Learning Specifications</i>	150
319	E.2: IDENTIFYING AND AUTHORIZING COMPETENCIES	151
320	<i>E.2.1: Free-form Narratives vs. Semi-formal Specifications</i>	152
321	<i>E.2.2: Eliciting competencies</i>	152
322	<i>E.2.3: Hierarchical Structure of Competencies</i>	153
323	<i>E.2.4: Deriving Semi-formal Specifications from Free-form Narratives</i>	154
324	<i>E.2.5: Authoring Free-form Narratives from Competency Components</i>	154
325	E.3: USING COMPETENCY SPECIFICATIONS AS A FOUNDATION FOR CURRICULUM SPECIFICATIONS	154
326	<i>E.3.1: Existing Models</i>	155
327	<i>E.3.2: Building Curricular Guidelines by Based on Competency Specifications</i>	156
328	<i>E.3.3: Building University-level Curricula Based on Competency Specifications</i>	157
329	<i>E.3.4: Specifying Program Outcomes as Competencies from Pedagogical Requirements</i>	157
330	E.4: COMPETENCIES AND STAKEHOLDER VALUE	158
331	E.5: ASSESSING COMPETENCIES	159
332	E.6: SUMMARY	159
333	E.7: REFERENCES FOR APPENDIX E	159
334	APPENDIX F: REPOSITORY DEVELOPMENT	161
335	F.1: REPOSITORY DEVELOPMENT	161
336	F.2: REFERENCES FOR APPENDIX F	162
337	APPENDIX G: ADDITIONAL VISUALIZATIONS AND ANALYSES	163
338	G.1: USE CASE-BASED ANALYSIS	163
339	<i>G.1.1: Case 1: Question from Prospective Student</i>	163
340	<i>G.1.2: Case 2: Question from Industry</i>	166
341	<i>G.1.3: Case 3: Question from Teacher</i>	168
342	<i>G.1.4: Case 4: Question from Educational Authority</i>	170
343	G.2: COMPARISON OF COMPETENCY SPECIFICATIONS	174
344	G.3: VARIOUS VISUALIZATIONS OF KNOWLEDGE	174
345	G.4: VISUALIZING FULL CURRICULA	178

346	G.5: REFERENCES FOR APPENDIX G	181
347	APPENDIX H: GLOSSARY AND NOMENCLATURE	182
348	H.1: CC2020 REPORT DEFINITIONS.....	182
349	H.2: DEFINITIONS/NOMENCLATURE ON A GLOBAL SCALE.....	184
350	APPENDIX I: SUSTAINABLE COMPUTING AND ENGINEERING COMPETENCE IN CHINA	194
351	I.1: ADAPTABLE AND SUSTAINABLE COMPETENCIES	194
352	I.2: AGILE EDUCATION FOR SUSTAINABLE COMPETENCIES	195
353	I.3: FACTORS AFFECTING AGILE COMPUTING AND ENGINEERING EDUCATION	196
354	I.4: OPEN EDUCATION ECOSYSTEMS FOR AGILE EDUCATION.....	197
355	I.5: SERVICE-ORIENTED COMPUTING EDUCATION.....	198
356	I.5: REFERENCES FOR APPENDIX I.....	198
357	APPENDIX Z: CONTRIBUTORS AND REVIEWERS.....	199
358	ACKNOWLEDGEMENTS.....	200
359	REFERENCES.....	201
360	R1: REFERENCES FOR REPORT	201
361	R2: EXTRA REFERENCES.....	206
362		
363		

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

Executive Summary

The field of computing has dramatically influenced science, engineering, business, education, philanthropy, and many other areas of human endeavor. In today's world, nearly everyone uses computers as part of everyday life. From smartphones and televisions to guidance systems, computing continues to be present in human life. This computing landscape offers students many challenging career opportunities. For those who are working in industry and government, computing is and will continue to be an essential component in shaping the future for humanity.

The computing disciplines must attract quality students from a broad and diverse cross-section of the public and prepare them to be capable and responsible professionals. Over decades, professional and scientific computing societies have taken leading roles in providing support for higher education in various ways, particularly in the formulation of curricular guidelines. The widely acclaimed Computing Curricula 2005 (CC2005), also known as the Overview Report, consolidated undergraduate computing curricula as they existed in 2005. The report contrasted published computing curricular guidelines for computer engineering, computer science, information systems, information technology, and software engineering. This landmark report provided illustrations to show the focus of these five curricula; it also provided tables to highlight the topic intensity and graduate profiles. CC2005 became a positive contribution to computing education.

Since 2005, much has changed in the computing field. Likewise, much has changed in the computing education world. The computer engineering reports progressed from CE2004 to CE2016; computer science reports progressed from CS2001 to CS2008 to CS2013. Information systems progressed from IS2002 to IS2010 with a new report that is pending. The initial information technology report was only in draft form in 2005, which eventually became IT2008, and since then evolved into IT2017. The software engineering report SE2004 became SE2014. Additionally, the computing field saw an emergence of cybersecurity which led to the CSEC2017 report. Furthermore, a new report is under development for the area of data science. To provide a better foundation for the continuous evolution of computing, it was apparent that a need existed to create a contemporary new report called Computing Curricula 2020, known also as CC2020.

To create a CC2020 report, a task force of fifty people from twenty countries has examined curriculum guidelines for undergraduate education and has referred to the computing professions and other supporting information as necessary. This report does not address graduate computing education or pre-baccalaureate education, although it occasionally mentions these areas.

This CC2020 report encompasses most of the themes contained in its predecessor. However, because of the changing dynamics of the computing disciplines, research in computing education and the workplace, this report contains many new "add-ons" and features that did not appear in the earlier report. Some of these changes include the following.

- Focusing on competency.
- Transitioning from knowledge-based learning to competency-based learning.
- Expanding curricular disciplines to include cybersecurity as well as data science.
- Expanding curricular diagrams and visuals.
- Creating a toolbox of visualizations for curricula and competencies.
- Developing an interactive website that will bring CC2020 results to public use.
- Charting a framework for future computing curricular activities.

The CC2020 report covers undergraduate programs in computer engineering, computer science, cybersecurity, data sciences, information systems, information technology, and software engineering. It provides a brief history of the evolution of previous curricular reports. Four important guidelines were followed.

1. The report must preserve and support the notion of computing in current and future decades throughout the world.
2. The report must capture future trends and visions from industry, from research, and "grass-roots" developments.
3. The report must be expansive and support existing, emerging, and future computing programs for its constituents.

419 4. The report must be flexible to achieve global enduring acceptance and be adaptable within multiple
420 educational contexts.

421 The stakeholders or constituents of this report are prospective students and their parents, current students, industry
422 and governmental officials, computing educators, and educational organizations and authorities.

423
424 Although computing as a discipline has been around for more than eighty years, many population groups are still not
425 clear about the subject area or what it means. This report contains descriptions of the computing disciplines, including
426 a brief description of their developments. The philosophy underpinning the CC2020 report is to treat computing as a
427 meta-discipline – a collection of disciplines having a central focus of computing.

428
429 One basic aspect of this report compared with the previous one is the move from knowledge-based learning to
430 competency-based learning. In general terms, one can think of competencies as the qualities an individual must possess
431 to be effective in a job, role, function, task, or duty. Competency requires demonstration of human behavior with
432 knowledge and skills.

433
434 There is a general agreement in educational circles that success in career readiness requires three things:

- 435 ○ Knowledge – "know-what" – designates a proficiency in core concepts and content, and application of
436 learning to new situations
- 437 ○ Skills – "know-how" – the ability to carry out tasks with determined results
- 438 ○ Dispositions – "know-why" – intellectual, social or moral tendencies.

439 Hence, any definition of competency must connect the three dimensions within a context or task represented as:

440
$$\text{Competency} = \text{Knowledge} + \text{Skills} + \text{Dispositions}$$

441 Therefore, instead of centering study guidelines on a body of knowledge, this report centers on competencies and
442 develops a competency framework.

443
444 An important feature of this report are the visual tools available to a stakeholder. For example, a prospective student
445 may wish to know how one computing discipline (e.g., computer engineering) differs from another discipline (e.g.,
446 information systems). By addressing the competency elements of knowledge, skills, and dispositions, it is possible to
447 generate a visual diagram that shows the commonalities and the differences between the two computing disciplines.
448 These competency-based visuals as well as other visuals provide a rich set of perspectives on computing disciplines.

449
450 The report also provides pathways for the future. It emphasizes the need for industry engagement to formulate
451 workplace competencies and the need for professional advisory boards to become involved with the development of
452 meaningful computing programs, e.g. through internship programs. Study cases also provide some suggestions for
453 ways that academics could transfer knowledge, skills and disposition to their students so that upon graduation, they
454 are competent graduates.

455
456 It is not the intent of this report to completely solve the nomenclature problem surrounding the computing field. For
457 example, "information technology" as used in this report refers to a subset of the computing field; some areas of the
458 world use this term to represent the entire computing field. This "Tower of Babel" challenge may never achieve a
459 solution. However, stakeholders must be aware of the nuances and differences in the meaning of different
460 terminologies used in different parts of the world.

461
462 Universal acceptance of global diversity and cultural sensitivity are essential in all fields, especially in the field of
463 computing which is very diverse itself. Degree structures are different in different countries and sometimes even in
464 the same country. Generally, there exist two, three, and four-year programs at the undergraduate level.

465
466 This CC2020 report does not provide specific curricula for each computing discipline. However, it does include some
467 examples. It points out important issues, some of which were not addressed until several years ago. Such subjects can
468 be included in a dedicated course but can also be included in a variety of existing or newly developed courses. The
469 report suggests and provides many opportunities. These include refreshing the paradigm of teaching and educating,
470 moving from knowledge or outcomes to proficiencies, and engaging graduates to exploit the benefits of workplace
471 competencies. These are described in the closing chapters of the report.

472
473 Finally, this CC2020 report is the result of an unprecedented cooperative effort among several computing
474 organizations spanning twenty countries. As one of the volumes of the "Computing Curricula Series," it provides

475 introspection and analysis of seven computing disciplines. The curricular guidelines that already exist today are the
476 product of many years of experimentation and refinement by industry leaders, computing educators, and faculty
477 colleagues in other disciplines. The academia-employer partnerships that will follow in the wake of this report will
478 help build even stronger computing programs for undergraduates worldwide.

479
480 This report is more than an overview of curricular guidelines. The overriding goal is to provide a useful and dynamic
481 pathway toward the 2030s. The report also provides a perspective on the major computing disciplines as they currently
482 exist and how they might exist in the future. It will help guide students, industry, and academia in the preparation of
483 capable computing graduates for the future. CC2020 and its tools will help shape the future of computing education.

484
485 — CC2020 Task Force
486
487

DRAFT

488

489 **Chapter 1: Introducing CC2020**

490

491 The Computing Curriculum 2020 (CC2020) project is an initiative launched jointly by several professional computing
492 societies to summarize and synthesize the current state of curricular guidelines for academic programs that grant
493 baccalaureate-level degrees in computing. This project aims not only to reflect the state-of-the-art in computing
494 education and practice, but also to provide insights into the future of the field of computing education for the 2020s
495 and beyond. The participating societies engaged a task force of individuals representing organizations from academia,
496 industry, and government. The principal organizations involved are the Association for Computing Machinery (ACM)
497 and the IEEE Computer Society (IEEE-CS). Other organizations include the Association for Information Systems
498 (AIS) and Association for Information Technology Professionals, Education Special Interest Group (AITP/EDSIG),
499 and the ACM Special Interest Group for Computer Human Interaction (SIGCHI). Project collaborators include:
500 Information Processing Society of Japan (IPSJ), the Chinese Computing Federation (CCF), the Latin American Center
501 on Computing (CLEI), ACM India, ACM Europe, and Informatics Europe. The results from this initiative provide a
502 durable portfolio of resources useful to students, industry, government agencies, educational institutions, and the
503 public on a global scale. This report is one key element of this portfolio.

504

505

506

507 **1.1: CC2020 Expectations**

508

509 The Computing Curricula 2005 Overview report CC2005 [Acm02] was an inaugural effort of several computing
510 organizations to provide a perspective on several computing disciplines for which baccalaureate curricula existed.
511 Much in the computing world has changed over fifteen years. Geography and varied conceptions of computing as
512 disciplines, professions, and cultures have influenced the context of degree-granting computing programs. The
513 CC2020 project considers regions of the world by involving organizational representatives from a variety of countries
514 to be part of the project. While currently published curricular guidelines (i.e., computer engineering, computer science,
515 cybersecurity, data science, information systems, information technology, software engineering) and the currently
516 emerging curricular models comprise CC2020's central domain of interest, the CC2020 deliverables are intended to
517 inform the process of rethinking existing or shaping new computing degree programs and disciplines.

518

519

520 **1.1.1: Project Purpose, Vision and Mission**

521

522 The following statement reflects the purpose of this CC2020 project.

523 *The purpose of the CC2020 project, as a modern extension of the CC2005 report, is to*
524 *provide global guidance in an evolving computing environment as it affects computing*
525 *baccalaureate degree programs worldwide.*

526

527 This report offers an up-to-date comparison and contrast of the curricular guidelines to situate and contextualize them
528 in the landscape of computing education globally. Additionally, it provides a characterization of computing that
529 facilitates designing and evaluating curricula and content, including interactive tools for academia and industry to
530 prototype models of proficiency development useful for exploring curricular opportunities.

531

532 The results of the project should inform students and their guardians, industry, government agencies, and academia
533 on the status and future of computing programs. This report should be helpful to current and prospective students and
534 their parents and guardians to make informed decisions in selecting and entering computer degree programs. It also
535 assists industry and government to understand profiles and expectations of graduates of computing degree programs.
536 Additionally, it helps computing programs to prepare students and resulting graduates, both academically and
537 professionally, to meet the challenges of the next decade.

538

539 The CC2020 task force vision is that:

540 *The CC2020 report shall become a sought-after and durable set of guidelines for use by*
541 *students, industry, governments and educational institutions worldwide to assist them to*

542 *gain insight on the expectations of computing baccalaureate-degree graduates for the next*
543 *decade.*

544
545 Likewise, knowledge alone is not sufficient for an individual to be productive in the changing world of computing.
546 Graduates of computing programs will require technical skills and (human) disposition that are integrated with
547 knowledge to achieve the professional expectations of a modern workplace. Therefore:

548 *The mission of the CC2020 project and this report is to produce a globally accepted*
549 *framework for specifying and comparing computing baccalaureate degree program that*
550 *meets the growing demands of a changing technological world and is useful for students,*
551 *industry and academia.*

552
553

554 **1.1.2: Project Strategies**

555 The CC2020 task force has established a set of goals to achieve the project’s vision and mission. These steps form a
556 pathway toward completion of the CC2020 project.

- 557 1. Develop a project plan with achievable milestones to complete ongoing projects on time.
- 558 2. Develop a robust report that reflects the project’s vision and mission.
- 559 3. Garner feedback from constituents for the development of this report.
- 560 4. Disseminate the CC2020 report worldwide.
- 561 5. Evaluate the efficacy of the CC2020 report.

562

563
564 Underlying these steps is the effort to extend the earlier overview report so it incorporates the developments of the
565 past fifteen years as well as the advancements forecast in the next decade. Computing technologies have developed
566 and continue to develop rapidly over time in ways that have had a profound effect on graduate expectations, curriculum
567 design and learning.

568

569 The CC2020 report proposes a performance-centered framework expressing what graduates of baccalaureate
570 computing programs should be able to learn and deliver with what they know. This report articulates computing
571 capabilities to enable faculty members to implement baccalaureate degree programs that focus on what students should
572 be able to accomplish rather than what instructors should teach. The report draws on learning sciences and educational
573 research and practices to advance the case for learning and curriculum development.

574

575

576 **1.1.3: Project Diversity**

577

578 The CC2020 project promotes sound principles regarding the ways that computing permeates society on a global
579 scale. Notwithstanding, it is not possible to cover all modes of thinking and all ways of learning. For example, the use
580 of experiential learning is beyond the scope of this report. Individual institutions and their faculties should use
581 innovative strategies to engage students in the learning process.

582

583 There are many pedagogical challenges and opportunities involving the computing field. Although this report
584 addresses the need for accessibility for all people, it does not discuss how to deal with the situation. The task force
585 believes such attention should take place at the institution level as well as through ongoing research by scholars and
586 practitioners.

587

588 One underlying theme of the report is the development of computing talent from all sectors and groups in our society.
589 A lack of diversity limits creativity and productivity. It excludes many potentially qualified individuals and it can be
590 a significant concern for many employers. For example, the underrepresentation of women in computing in some
591 countries has received much attention [Reg1]. This report recognizes the importance of diversity and recommends
592 that academic computing departments promote best practices to attract and retain greater student diversity.

593

594 The CC2020 project has placed inclusion at the core of its activities from the very first step of forming its membership.
595 The task force has a diverse composition by gender, type of work, affiliation, geography, and global professional
596 presence. Some statistics for the task force are as follows.

597

Number of task force members: 50
Number of continents: 6
Number of countries: 20
Number of women: 21; men: 29

Number of international professional societies: 11
Number of industry-government: 7, academic: 43
Steering committee: 5 women, 10 men
Task force co-chairs: 1 woman; 1 man

598

599

600

601

602

603

604

605

1.2: Project Stakeholders

606

607

608

609

610

611

612

613

1.2.1: Prospective Students and their Guardians

614

615

616

617

618

619

620

621

622

623

624

625

626

1.2.2: Current Students

627

628

629

630

631

632

633

634

635

636

1.2.3: Industry

637

638

639

640

641

642

643

644

645

646

647

648

649

The task force is aware that it cannot satisfy the desires of all people. It has made every effort to position this report within the broader computing landscape. As a global document, this report provides guidelines from diverse communities and is not prescriptive in its recommendations.

The CC2020 project has identified five groups of stakeholders the members of which may benefit from the competency-based approach to specifying outcome expectations. The project stakeholders include: (a) prospective students and their guardians, (b) current students, (c) industry, (d) educators, and (e) educational organizations and authorities. All stakeholders should benefit from the fruits of this CC2020 project.

When prospective students, supported by their parents or guardians, are considering studying computing at a university, they need to understand differences in computing programs when making their choice. This report could enable them to compare programs allow them to evaluate the extent to which a program aligns with their expectations of a job or a career path. Students may understand that they want to study computing, but very few will likely understand that there are many disciplines and the differences between them. A typical question posed by a prospective student might be:

I am considering a computing curriculum that fits my preferences. Among the candidate schools, there are several computing programs available. Are graduates of these programs expected to work primarily as individuals (e.g., doing coding) or also work with other people?

Current students enrolled at an institution of higher learning could face a choice of courses from their own institution or another institution. This stakeholder category could also apply to students in another discipline who are considering a hybrid curriculum that includes computing components such as a minor in a computing discipline. A typical question posed by a current student might be:

Which courses in the information systems curriculum of my university have greater emphasis (more detailed coverage or longer duration) compared to the information systems curriculum guidelines?

Industry refers to entities that (1) are hiring graduates, (2) are collaborating with universities to choose or specialize a curriculum or need a special course, or (3) are collaborating in a curriculum by providing internships. For representatives of industry (such as hiring managers), the most important question relates to the preparation of the graduates of a program compared to the expectations of a specific employer. More importantly, industry employers and recruiters need to understand what incoming employees have learned. For example, employers who are looking for software developers would likely desire people that have deeply studied courses usually found in software engineering or computer science programs. On the other hand, if the employers desire people who have studied organizational issues in addition to a solid foundation in computing, then they would prefer graduates from an information systems curriculum. A relevant question that industry stakeholders may like answered could be:

Our industry requires our employees to have specific knowledge at relevant knowledge levels and several key dispositions. Are there outcomes of a course in curriculum XYZ that are appropriate for the continued professional education for our employees?

650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705

1.2.4: Computing Educators

Computing educators are faculty members or teachers of a computing academic unit within a school or university, who are responsible for designing and implementing educational experiences related to a computing discipline. These computing educators should understand how their current curriculum, or a prospective curriculum, fits within accepted curricular recommendations. It would be useful if they were able to compare their curriculum to professional curricula guidelines to help them understand what may be missing. They might ask a question as follows.

What knowledge areas are applicable for my course? Could I adopt an existing course from elsewhere to fill a gap or provide an alternative in my curriculum?

1.2.5: Educational Organizations and Authorities

Educational organizations or authorities are entities who have some authority over university education. These entities could include professional organizations or societies, national or regional ministries of education that govern and finance universities, and national or international (e.g., European) bodies that rate, assess, or accredit university education, or define qualifications for certification. The following shows a typical question that educational authorities may like answered.

Could we accept students from a specified curriculum X to complete curriculum Y?

1.3: Project Background

Computing curriculum guidelines have been of interest to colleges and universities and their faculty members for more than six decades. The following is a brief summary.

1.3.1: Brief History

In the 1980s, the ACM and the Computer Society of the Institute for Electrical and Electronics Engineers (IEEE-CS) established a joint committee to update Curriculum'78. The committee's goal was to develop more modern computing curricula (CC) guidelines for baccalaureate, undergraduate degree programs in computing. The committee's effort created Computing Curricula 1991 [Tuc1], also called CC1991 or CC'91. This report, which many educators interpreted as computer science, received only moderate acceptance because by the early 1990s different computing disciplines were maturing (e.g., information systems) or emerging (e.g., computer engineering, information technology, software engineering). Over the years, however, the efforts of the CC'91 committee resulted in a series of comprehensive reports that reflected not only maturing and emerging computing disciplines. Many of these documents are available on the ACM website [Acm01]. Additionally, Europe has also formulated computing definitions through the European Higher Education Area (EHEA) [Eur2].

In the late 1990s, ACM and IEEE-CS cooperated to generate the CC2001 report [Acm01] that represented some major advances. This report called for the creation of an overview document; it also called for each of the major computing disciplines recognized at the time to develop its own curricular report. The major areas at the time included computer engineering, computer science, information systems, information technology, and software engineering, although information systems had published its own discipline reports already for two decades. CC2001 recognized the growing and dynamic nature of computing. The number of computing-related disciplines was increasing; hence, work in curricular development was to embrace new computing disciplines as they emerge. The tenets established within the CC2001 report eventually produced the broadly influential CC2005 overview report, co-sponsored by the ACM, the Association for Information Systems (AIS), and IEEE-CS.

One of the documents created in the series was the 2005 publication titled Computing Curricula 2005, also known as CC2005 [Acm02]. The report has received worldwide acclaim by contrasting the differences and the commonalities of diverse computing discipline areas. It was an inaugural effort of several computing organizations to provide

706 perspective on several computing disciplines for which baccalaureate curricula existed. It also described “how the
707 respective computing undergraduate degree programs compare and complement each other” [Acm02 p1]. Chapter 3
708 will review CC2005 in greater depth.

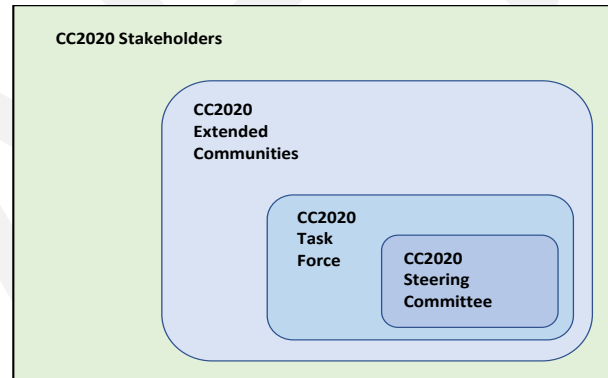
709
710 Since the publication of CC2005, much has changed. Each of the curricula described in 2005 has had an update, in
711 some cases multiple times. New areas of the computing field have gained prominence to warrant production of their
712 own curriculum guidelines. The global and interdisciplinary nature of computing has become even more evident today
713 [Sim1]. The 2005 document was by its own admission “North-American-centric”, and it mentioned the need for future
714 such documents to be more international in scope. The CC2020 project fulfills that promise.

715
716

717 **1.3.2: Project Organization and Structure**

718
719 In 2015, the Association for Computing Machinery (ACM) began to explore avenues through which to update the
720 overview report. In 2016, the ACM decided to proceed with CC2020. It established an exploratory committee to
721 ascertain the need for a new report. Initially, ACM and IEEE-CS became the principal sponsors of the CC2020 project.
722 Other professional organizations joined in the effort with additional sponsorship. These included the Association for
723 Information Systems (AIS) and the Association for Information Technology Professionals, Education Special Interest
724 Group (AITP/EDSIG), and the Special Interest Group on Computer Human Interaction (SIGCHI). Project
725 collaborators included the Information Processing Society of Japan (IPSJ), the Chinese Computing
726 Federation (CCF), the Latin American Center for
727 Computer Studies (CLEI), Informatics 4 All (I4All),
728 and ACM India.

729
730 The ACM and IEEE-Computer Society respectively
731 appointed Steering Committee co-chairs who in
732 turn, recruited other members of the Steering
733 Committee in 2017. These societies expanded the
734 Steering Committee into a Task Force of fifty
735 volunteers who joined the effort to work on the
736 project and produce this report. Figure 1.1 illustrates
737 the current structure of the CC2020 project.
738



739
740 Figure 1.1. CC2020 project structure

741
742

743
744 **1.4: Overall Scope of Computing**

745
746 There are many types of computing degree programs. Reliable information about the number of different kinds of
747 computing degree programs is difficult to obtain. However, in the past fifteen years, the number and type of computing
748 degree programs available to students has dramatically increased. This report suggests ways that future computing
749 curricula guidelines might develop. To this end, the CC2020 report aims not only to describe the computing sub-
750 disciplines currently identified by curricular documents, but also to acknowledge that the boundaries of computing
751 disciplines has expanded and will continue to expand greatly.

752
753

754 **1.4.1: Current Discipline Structure**

755
756 The baccalaureate disciplines for which computing curricula exist or are in the development process at the time of this
757 writing are as follows.

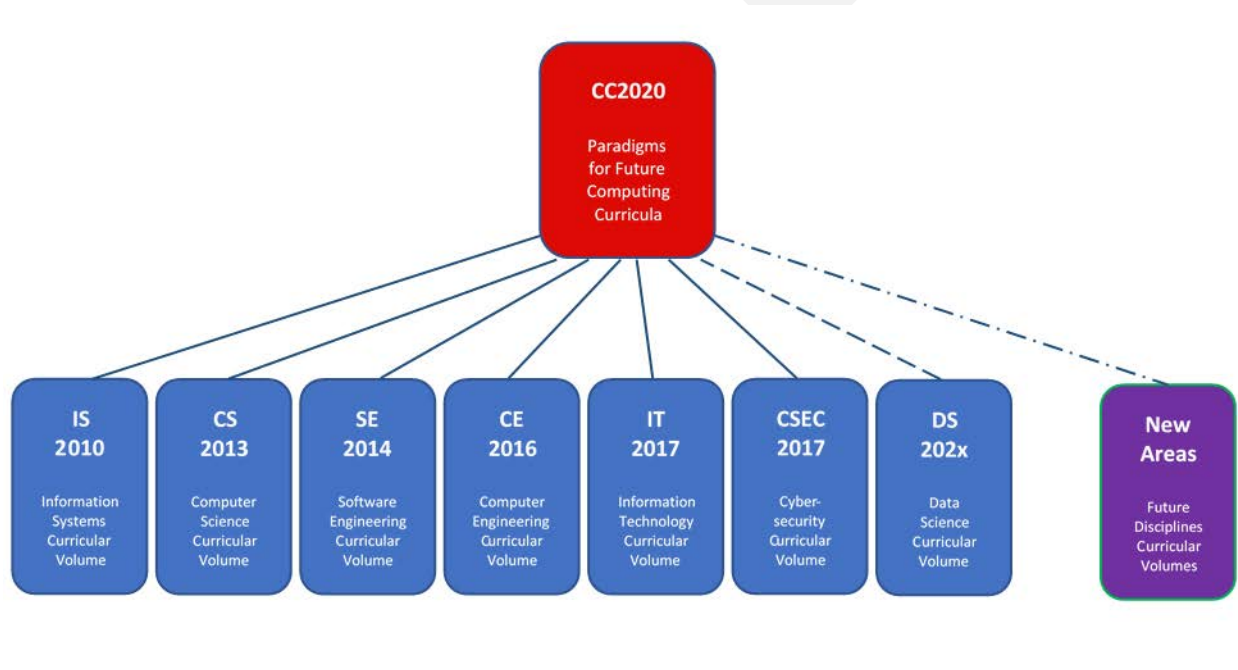
Computer engineering (CE) Information systems (IS)
 Computer science (CS) Information technology (IT)
 Cybersecurity (CSEC) Software engineering (SE)
 Data science (DS)

758 Each of these disciplines has a recent volume or will soon complete a volume for undergraduate curriculum guidelines
 759 that one or more international professional and scientific societies has endorsed and published. These disciplines have
 760 affected a large majority of undergraduate students worldwide who are majoring in computing.
 761

762 One would expect that groups from other disciplines in computing might undertake the effort to create and maintain
 763 international undergraduate curricular guidelines. In such cases, those guidelines will become part of future editions
 764 of this report.
 765

767 **1.4.2: Timeline of Curricular Guidelines**
 768

769 The foundation for this report is the set of curriculum standards that currently exist for undergraduate degree programs
 770 in major computing-related fields. The diagram in Figure 1.2 illustrates what has become the “computing curricula
 771 series” and the top-level overview block, CC2020, represents this report. For the six-existing discipline-specific
 772 curricula volumes, each one represents the best judgment of the volunteers representing relevant professional,
 773 scientific, and educational associations. Each report serves as a definition of what these degree programs should be
 774 and accomplish.
 775



776
 777 Figure 1.2 Structure of the Computing Curricula Series
 778
 779

780 This report encompasses recent and ongoing curricula efforts including the following:

- 781 ▪ Curriculum Guidelines for Undergraduate Degree Programs in Information Systems 2010 (IS 2010)
- 782 ▪ Computer Science Curricula 2013 (CS 2013)
- 783 ▪ Software Engineering Curricula 2014 (SE 2014)
- 784 ▪ Computer Engineering Curricula 2016 (CE 2016)
- 785 ▪ Information Technology Curricula 2017 (IT 2017)
- 786 ▪ Cybersecurity Curricula 2017 (CSEC 2017)
- 787 ▪ Data Science Curricula 202x (DS202x)
- 788 ▪ Other emerging disciplines

789
790 The data science curricular guideline report addressing the computing components useful for data engineering, big
791 data, and data analytics is currently under development. Other recent publications that have had some influence in this
792 area include the EDISON Data Science Framework [Edi1] and the document titled “Envisioning the Data Science
793 Discipline: The Undergraduate Perspective: Interim Report” by the National Academies Press [Nas1].
794

795 Professional organizations should view the computing curricular reports as minimal guidelines to avoid being
796 prescriptive. Curriculum developers have had and still have the freedom to act independently for their constituencies.
797 The anticipation is that undergraduate baccalaureate degree programs will greatly exceed the minimal
798 recommendations suggested in these and subsequent curricular reports.
799

800 While some of the mentioned reports are undergoing revision at the time of this writing (e.g., IS2010), the task force
801 has made no effort to update their contents as that endeavor is beyond the mission and authority of this project. Rather,
802 the CC2020 task force has taken current curricula volumes, compared their contents, and synthesized what it believes
803 to be essential descriptive and comparative information. The current curricular volumes contain much detailed
804 information not included here. Readers who want detailed information about any of the disciplines covered in this
805 report should consult the original sources found at the ACM website [Acm01].
806

807 In addition to using these reports, the task force has referred to the computing professions and other supporting
808 information as necessary in preparing this report. The report has not focused on other types of undergraduate
809 computing degree programs, on graduate education in computing, or on the identities of the computing research
810 communities. Additionally, it has not included any information or comment about non-traditional computing
811 education such as provided in conjunction with vendor- and government-specific certification programs and massive
812 open online courses (MOOCs). Those areas continue to be deserving of evaluation, but such work is beyond the scope
813 of this project.
814

815

816 **1.5: Guiding Principles**

817

818 The intent of the CC2020 report is to provide a conceptual framework within which to conceptualize computing
819 education as well as the relationships between abilities, bodies of knowledge, professional profiles, educational
820 contexts, and degree programs. The CC2020 Task Force acknowledges that the terminology in use has never had
821 universal agreement and it is subject to a great deal of variation. Terms such as “computer engineering,” “computer
822 science,” and “information technology” mean radically different things to computing educators and practitioners in
823 different parts of the world and indeed, within national localities. Section 6.1 discusses nomenclature and terminology
824 in the computing field.
825

826

827

827 **1.5.1: Four Principles**

828

829

829 The CC2020 task group followed these four principles in developing this CC2020 report.

830

831 *1. It must preserve and support the notion of computing, both now and in the future, throughout the world.*

832 The term *computing* identifies the broad field involving computers and reflects the reality that this word has become
833 globally ubiquitous. The task force is aware that some regions of the world use different terminologies such as
834 “informatics” or “information and communication technology (ICT)” to have a similar meaning as the word *computing*
835 to represent a field. This report assimilates these similarities and differences.
836

837

837 *2. It must capture future trends and visions from industry, from research, and across the entire spectrum of society.*

838 This principle has many facets to it. The CC2020 project must remain responsive to general educational needs, changes
839 in technology, as well as existing and emerging technologies. The intended audience for this report includes students,
840 industry/employers, policymakers, professional societies, accreditation agencies, and academic institutions.
841

842

842 *3. It must be expansive and support existing, emerging, and future computing programs for its constituents.*

843 It is important to contrast different types of computing programs. This principle is important for the CC2020 project
844 and its constituents (e.g., students and prospective students) to aid in their awareness of how one computing discipline

845 (e.g., computer engineering) differs from another computing discipline (e.g., information technology). This principle
 846 is also important for creators of educational programs who will use the pending resources to guide the development
 847 and enhancement of robust degree programs. The task force also seeks to establish a universal understanding of the
 848 terminology used to describe anticipated capabilities of computing graduates in different areas of specialization.
 849

850 *4. It must be flexible to achieve global enduring acceptance and be adaptable within multiple educational contexts.*
 851 Authors in the computing education field must use language that is globally neutral (not specific to an educational
 852 system or context) to reflect documents such as the Bologna Declaration [Bol1]. To some degree, users of this report
 853 should be able to use different parts of this report to suit their own needs.
 854
 855

856 **1.5.2: Constituent and Public Outreach**

857
 858 As a follow-up to this report, the task force is preparing and will publish an interactive “Guide to Undergraduate
 859 Programs in Computing” website. This website will include an interactive toolbox of procedures to assist the
 860 constituents of this report and the public. The website will include much more information than this CC2020 report
 861 can convey. It will provide information for prospective students and for those who advise them to help them make
 862 well-informed choices. This interactive website will also provide the means for comparing computing programs across
 863 national contexts as well as provide support for finding information for assessing and developing modern and future
 864 computing curricula. Figure 1.3 shows the landing page of the CC2020 website [Ccw1].
 865



Figure 1.3. Landing page (partial) of the CC2020 website at <https://www.cc2020.net/>

866
 867
 868 Computing itself will continue to evolve. In addition, new computing-related fields are likely to emerge. As updates
 869 of existing discipline-specific reports develop and as additional reports for new fields of computing emerge, updated
 870 versions of this report will likely be produced. A visit to the ACM website [Acm00] or to the IEEE-CS website [Cos1]
 871 will allow users to access the most current version of this or other curricular reports.
 872
 873

874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915

1.6: CC2020 Report Structure

This CC2020 report addresses baccalaureate degree programs in computing. The main body of the report consists of seven chapters in addition to this Chapter 1 titled *Introducing CC2020*. Chapter 2, titled *Evolution of Computing Education*, addressing computing and its disciplines in general and the computing landscape. Chapter 3, titled *Knowledge-based Computing Education*, discusses computing curricula guidelines and the knowledge derived from CC2005. Chapter 4, titled *Competency-based Computing Education*, addresses market forces, the importance of skills and disposition, and the canonical definition of competency. Chapter 5, titled *Visualization and Competency Specification*, provides a catalyst for unifying computing education through graphs for the project’s stakeholders with modern visualizations. Chapter 6, titled *Global Considerations*, addresses worldwide nomenclature and degree structures as well as global examples of computing programs. Chapter 7, titled *Curricular Design – Challenges and Opportunities* examines various aspects of computing education useful for a successful implementation, and Chapter 8, titled *Beyond CC2020* concludes the report.

Appendix A shows an example of a poster that displayed the use of the CC2005 report for a broader audience. Appendix B presents several computing skill frameworks. Appendix C illustrates examples of competencies for various computing curricula. Appendix D explores the nature of competency-based computing curricula. Appendix E addresses the use of competencies for specification of degree programs in computing. Appendix F addresses a strategy for the development of a visualization repository. Appendix G contains a large set of visualization examples. Appendix H provides a glossary of terms as well as a crosscutting nomenclature as used in different parts of the world. Appendix I summarizes the Chinese “Blue Book” project surrounding agile competencies. Appendix Z recognizes contributors to the CC2020 project as well as reviewer contributions.

The CC2020 task force is hopeful that this report will help students in deciding on a computing path of study, industry representatives to understand better the profiles of graduating students, and departments to create effective curricula or help them improve the curricula they already have. This CC2020 report, with its curricula recommendations and illustrations, should be a guiding light for computing education worldwide. The intent is to have students develop computing aptitudes, so that they can achieve professional success in their future careers.

1.7: Digest of Chapter 1

This chapter reviews the vision, mission, purpose, and development of the CC2020 project. It describes the project strategies and project stakeholders and how they will benefit from this report. It also reviews the background associated with the CC2020 project and explains the guiding principles in detail. The structure of this report is previewed.

916

917 **Chapter 2: Evolution of Computing Education**

918

919

920

921

922

923

924

925

926

927

927 **2.1: What is Computing?**

928

929

930

931

932

933

934

935

936

937

938

938 **2.1.1: Early Meanings**

939

940

941

942

943

944

945

946

947

947 Early on, computing had a somewhat singular meaning. In its short history, various shades of interpretation have evolved with varying specializations. For example, a person with a background in information systems will view computing somewhat differently from a computer engineer. The emergence of new information technology industries, the increased reliance on computation in all parts of society, and the shifts in the demand for computing throughout a worldwide economy reflect changes in the field and its broad applications [Nrc1]. Because society needs people to do computing well, it is important to understand that computing is not only a profession but also a collection of disciplines [Acm02].

948

949

950

951

952

953

954

955

948 Computing is not just a single area of study, but rather a family of study areas. During the 1990s, important changes in computing, communications technology, and their societal effects led to important changes in this family of disciplines. Those changes included:

- 951 ▪ Computer engineering emerging from electrical engineering
- 952 ▪ Computer science evolving into a more mature academic discipline
- 953 ▪ Information systems expanding as computers became part of organizational work environments
- 954 ▪ Information technology emerging as a new discipline that fosters practical application and maintenance
- 955 ▪ Software engineering emerging as a discipline based on computer science and computer engineering.

956

957

958

959

960

956 After the 1990s, computing programs around the world saw a maturation. They continued to evolve, thereby creating a greater range of study opportunities for students and educational institutions [Acm02]. Additionally, there were many jobs available focused on software use rather than design and development that accelerated that maturation.

961

961 **2.1.2: Recent Undertakings**

962

963

964

965

966

967

968

969

963 Advances in worldwide curricula development have expanded the scope of traditional computing areas of study (computer engineering, computer science, information systems, information technology, and software engineering). New curricular efforts have expanded educational frontiers with significant development in areas of cybersecurity, data science, and other emerging areas of study. While these efforts have general agreement as being within the frontiers of computing education, what lies at the core of computing and how this core supports future expansions in computing education is less clear.

970 Section 2.3 of this report describes recent updates to curricular development in the traditional computing areas of
 971 study as described above. Additionally, it addresses a recent curricular report in cybersecurity published in 2017. It
 972 also addresses an emerging document in the area of data science. Although not part of this report, a study area of
 973 artificial intelligence is an area of renewed interest.
 974

975 In 2018, the National Academies of Sciences, Engineering and Medicine in the United States described the changing
 976 landscape of computing as follows [Nas2]:

977 *Two areas have been central in the last decade: the continued and increased need for information security, and data as a*
 978 *resource and driver for decision making. The protection of digital information and data; the protection of software and*
 979 *hardware systems and networks from unauthorized access, change, and destruction; and the education of users to follow best*
 980 *security practices are crucial to every organization. We rely upon a connected, networked, and complex cyberspace with*
 981 *vulnerabilities that is almost continuously under attack. ...*

982 *During the last decade, computing has taken a new, more empirically driven path with the maturing of machine learning,*
 983 *the emergence of data science, and the “big data” revolution. Data science combines computing and statistical methods to*
 984 *identify trends in existing data and generate new knowledge, with significant applications throughout all sectors of the*
 985 *economy, including marketing, retail, finance, business, health care and medicine, agriculture, smart cities, and more. ...*

986 *Software tools and systems for animation, visualization, virtual reality, and conceptualization have emerged as a medium*
 987 *for the arts (digital media and multimedia practices) and are driving advances in the entertainment industry (computer-*
 988 *generated graphics in films and video games, and digital methods in music recording), as well as training and education using*
 989 *virtual environments.*

990 *Computing has become more pervasive among a host of academic disciplines, beyond just the practical use of ubiquitous*
 991 *software tools. New algorithmic approaches and discoveries are helping to drive advances across a range of fields, leading*
 992 *to new collaborations and an increased demand for deeper knowledge of computing among academics and researchers,*
 993 *challenging conventional disciplinary boundaries.*

994 This National Academies report has the expectation of generating a profound influence on the development of data
 995 engineering and data science worldwide, as well as computer security.
 996
 997
 998

999 **2.2: Landscape of Computing Disciplines**

1000 This section of the report provides both historical and contemporary perspectives on the evolution of computing. The
 1001 section places computing in context as viewed by professionals in computing.
 1002
 1003

1004 **2.2.1: Early Developments**

1005 At the earliest stages of the development of computing, education and training for computing jobs were closely
 1006 associated with research and development of computing technologies as manifested by the manufacturers of the
 1007 artifacts that industry produced. Relatively soon, however, universities started to offer courses associated with
 1008 computing. By the end of 1950s, about 150 universities and colleges in the United States offered courses in computing
 1009 in a broad range of topics ranging from “logical design of computers” through “programming of digital computers”
 1010 as well as from “information storage and retrieval” to “business and industrial analysis” [Fei1; Ted1]. Fein also
 1011 provides an insightful discussion that explores the concept of a “computer sciences” discipline and suggests that one
 1012 such area of study is likely to emerge. Continuing:
 1013

1014 *Most aspects of computers, data processing and the related fields discussed in this study now meet (the specifications of a*
 1015 *discipline articulated in the paper) or may be meeting them in the next ten years [Fei1].*
 1016
 1017

1018 The report also clearly defined computing as a field of study that consists of multiple disciplines, proposing five
 1019 different departments: computer, operations research, information and communication, systems, and philosophy of
 1020 organization. A modern interpretation would roughly correspond to current disciplines such as computer
 1021 science/computer engineering, operations research/management science, information science, information systems,
 1022 and computing ethics. It is interesting to see how the breadth of the field links computing as an academic discipline to
 1023 the practical applications and contexts [Fei1].
 1024

1025 In the 1960s, three major streams of academic computing program types emerged: computer science, computer
 1026 engineering, and information systems. These three had clearly different perspectives: computer science was a highly

1027 theoretical study of “information structures and processes and how those structures and processes can be implemented
1028 on a digital computer” [Ted1 p45]; computer engineering was an offshoot of electrical engineering that focused on
1029 applying established engineering practices and processes to the design and construction of computing hardware; and
1030 (management) information systems focused on the practical use of computing in organizations (mostly businesses).
1031 Both computer science and information systems had ACM-sponsored curriculum recommendation projects, leading
1032 ultimately to *Curriculum 68* [Acm13] for computer science and IS curricula for graduate (1972) [Acm14] and
1033 undergraduate (1973) [Acm15] programs.

1034
1035 In 1989, a *Task Force on the Core of Computer Science* characterized the discipline of computing as a combination
1036 of three separate but tightly intertwined aspects: theory, abstraction (modeling), and design [Den1]. Those aspects
1037 relied on three different intellectual traditions (the task force called them paradigms): the mathematical (or analytical,
1038 theoretical, or formalist) tradition, the scientific (or empirical) tradition, and the engineering (or technological)
1039 tradition [Ted2, p153].

1040
1041

1042 **2.2.2: Contemporary Advances**

1043

1044 In the 1970s, 1980s, and 1990s, relatively little changed structurally in computing education: computer engineering,
1045 computer science, and information systems all evolved but continued to have separate identities that made it relatively
1046 easy for prospective students to choose between different options. However, in early 2000s the landscape of computing
1047 education started to change significantly. Software engineering emerged as its own discipline with a curriculum
1048 recommendation after decades of organizational practice and research. Programs in information technology started to
1049 fill the need for graduates with an applied focus on developing and maintaining computing infrastructure and
1050 supporting users. At the same time, the five established computing disciplines (CE, CS, IS, IT, and SE) strengthened
1051 their collaboration that allowed computing to gain a stronger integrated identity. One of the achievements of CC2005
1052 was the formation of an integrated computing discipline,
1053 which was the result of the analysis, documentation, and
1054 clarification of the relationships between the five
1055 subdisciplines. The document illustrated the general
1056 characteristics of computing education with Figure 2.1,
1057 which summarizes the development of the field during the
1058 transformation that took place starting in 1990s.

1059

1060 In the 2010s, two new areas emerged as new disciplines in
1061 the broader computing space: cybersecurity and data science.
1062 In 2017, a curriculum recommendation and accreditation
1063 criteria for cybersecurity emerged. Data science, however,
1064 often has different instantiations and possible directions
1065 depending on the disciplinary background of those engaging
1066 in a discussion [Cas1].

1067

1068 As Figure 2.1 suggests (based on academic curricular reports), hardware and software occur in different forms.
1069 Computing hardware is primarily the domain of computer engineering, often with close links to electrical engineering.
1070 The disciplines with the strongest focus on software development are computer science and software engineering;
1071 computer science is the foundational discipline with an emphasis on discovery related to programming, algorithms,
1072 and data structures, whereas software engineering addresses more applied concerns regarding the processes and
1073 actions needed for designing reliable, secure, and high-quality software systems.

1074

1075

1076

1077 **2.3: Status of Computing Discipline Reports**

1078

1079 This section briefly characterizes seven computing disciplines for which the ACM and IEEE-CS together with AIS
1080 have developed an undergraduate curriculum recommendation over the past decade. The seven areas include computer

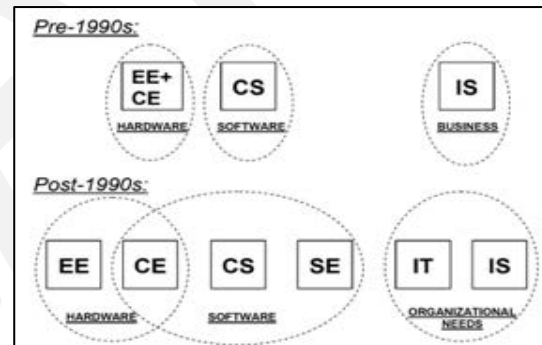


Figure 2.1 Computing disciplines compared, from CC2005

1081 engineering, computer science, cybersecurity, data science, information systems, information technology, and
1082 software engineering. This section describes the disciplines with a focus on their educational programs.
1083
1084

1085 **2.3.1: Computer Engineering**

1086

1087 Computer engineering (CE) brings together computing and electrical engineering in a way that embodies the science
1088 and technology of design, construction, implementation, and maintenance of software and hardware components of
1089 modern computing systems, computer-controlled equipment, and networks of intelligent devices. CE is the computing
1090 discipline that explicitly focuses on the development of hardware and software interface as a hardware embedded
1091 element of a computing system. The *Computer Engineering Curricula 2016 Report*, known also as CE2016, represents
1092 curriculum guidelines for undergraduate degree programs in computer engineering [Acm06]. The goals of the effort
1093 include incorporating past and future development needs, supporting professionals responsible for teaching a range of
1094 degree programs in computer engineering worldwide.
1095

1096 The capabilities of CE graduates integrate aptitudes of electrical engineering, software engineering, and computer
1097 science with a heavy emphasis on mathematics required as a foundation. CE2016 is very clear about the fact that
1098 graduates from CE programs should have the ability to design computers, design computer-based systems, and design
1099 networks with additional specifications that design needs exceed simple configuration and assembly. CE is specifically
1100 an engineering discipline where graduates must have a breadth of knowledge in mathematics and engineering sciences
1101 with a preparation for professional practice or graduate work in engineering. Many countries provide CE graduates
1102 the opportunity to become licensed professional engineers according to local governmental rules.
1103

1104 The computer engineering discipline enables graduates to analyze and design circuits, manage the design of computer
1105 hardware components, and develop networking hardware solutions. For students interested in gaining experiences in
1106 integrating computing capabilities directly with computing hardware, computer engineering could be an appropriate
1107 degree program choice. Computer engineering also provides an excellent preparation for the design and development
1108 of modern technologies that closely integrate the physical world with the world of the artificial.
1109
1110

1111 **2.3.2: Computer Science**

1112

1113 The *Computer Science Curricula 2013* project had two directives in developing its subsequent report, known as
1114 CS2013 [Acm04]. They included (1) a review of Computing Curriculum 2001 and CS2008, and (2) to seek input from
1115 diverse audiences to broaden participation in computer science (CS). CS2013 also had several high-level themes that
1116 provided overarching guidance for the development of its report. These include embracing an outward-looking view
1117 of the discipline, size management of the curriculum, providing actual exemplars to identify and describe existing
1118 successful courses and curricula, and being responsive to institutional needs, goals, and resource constraints.
1119

1120 Because of its theoretical foundations, people sometimes view computer science as a fundamental discipline. It is,
1121 however, at times erroneously equated with all of computing. This misconception is understandable given that the
1122 theoretical roots of computer science have emerged separately from the engineering tradition of computing's earliest
1123 days [Ted1 sec3.2]. While the physical sciences are fundamental and offer theoretical basis to engineering fields, none
1124 subsumes the other and each has a well understood distinct identity. Similarly, this document and its predecessors
1125 have successfully established independent identities relative to computer science.
1126

1127 CS continues to have a more theoretical focus among the other computing disciplines and its connection with abstract
1128 mathematics is still strong. A CS degree alone typically does not provide expertise regarding a specific context
1129 applicable to computing. Instead, CS programs emphasize abstract computational capabilities. CS2013 identifies
1130 abstraction, complexity, and evolutionary change as recurring themes in computer science, while sharing a common
1131 resource, security, and concurrency as general principles. These principles are closely linked to proficiency in
1132 programming and software development which are very important in most CS programs. CS2013 allocates about 40%
1133 of its core hours to algorithms and complexity, programming languages, software development fundamentals, and
1134 software engineering.
1135
1136

1137
1138
1139
1140

2.3.3: Cybersecurity

1141 Cybersecurity is a highly interdisciplinary field of study. Specific degree programs are often associated conceptually
1142 and practically with one of the established disciplines in a way that has a significant effect on the fundamental identity
1143 of the program. The *Cybersecurity Curricula 2017* report [Acm08], known also as CSEC2017, became public in 2017.
1144 The report recommends security in eight areas to include data, software, component, connection, system, human,
1145 organizational, and societal. The CSEC2017 mission was to develop comprehensive and flexible curricular guidance
1146 in cybersecurity education that would support future program development and to produce a curricular volume that
1147 structures the cybersecurity discipline and provides guidance to institutions seeking to develop or modify a broad
1148 range of programs.

1149
1150 The report explicitly states that there is a broad spectrum of cybersecurity jobs from technical (such as cryptography
1151 and network defense) to managerial (such as policy and regulatory compliance) positions. At the same time, it also
1152 recognizes that every graduate of a cybersecurity program requires both technical skills and business acumen,
1153 essentially a managerial understanding of the organizational actions needed to ensure system-level security. A degree
1154 in cybersecurity prepares graduates for a broad range of application areas, including public policy, procurement,
1155 management, research, software development, IT security operations, and enterprise architecture.

1156
1157 The need for specialized abilities that cybersecurity graduates should have occurs almost daily. Continuous challenges
1158 of various types face organizations around the world who must secure data regarding their customers. Solutions that
1159 secure organizational data are multidimensional ranging from highly technical to organizational policies and societal
1160 legal and regulatory responses, creating a significant need for professionals with a broad range of specialized security
1161 expertise combined with the generic individual foundational abilities (such as problem solving, critical thinking, oral
1162 and written communication, teamwork, negotiation) that all computing professionals need.

1163
1164

2.3.4: Data Science

1165
1166
1167 Data science (DS) is a new area of computing that is closely related to the fields of data analytics and data engineering.
1168 One definition of data science is “a set of fundamental principles that guide the extraction of knowledge from data ...
1169 [and] involves principles, processes, and techniques for understanding phenomena via the (automated) analysis of
1170 data” [Pro1].

1171
1172 Several DS projects have emerged in recent years. These include the *EDISON Data Science Framework (2017)* project
1173 [Edi1], the *National Academies Report on Data Science for Undergraduates (2018)* [Nas1], the *Park City Report*
1174 *(2017)* [Par1], the *Business Higher Education Framework (BHEF) Data Science and Analytics (DSA) Competency*
1175 *Map (2016)* [Bhe1], and the *Business Analytics Curriculum for Undergraduate Majors (2015)* [Ban1]. ACM
1176 conducted initial DS workshops in 2015; a report described the discussions, reflected the diversity of opinions, and
1177 proposed a list of knowledge areas useful for the field [Cas1]. In August 2017, the ACM Education Council created a
1178 task force to articulate the role of computing in the DS field. The task force produced an initial draft report tentatively
1179 tagged as (DS202x) in February of 2019 [Dat1] followed by a second draft report in December of 2019 [Dat2].

1180
1181 The second draft describes a “competency framework” that addresses knowledge areas representing a body of material
1182 for data science degree programs that capture high-level competencies, skills, and dispositions. The knowledge areas
1183 include (a) computing fundamentals, (b) data acquisition and governance, (c) data management, storage, and
1184 retrieval, (d) data privacy, security, and integrity, (e) machine learning, (f) data mining, (g) big data, (h) analysis and
1185 presentation, and (i) professionalism. For a full curriculum, these areas need augmentation with courses covering
1186 calculus, discrete structures, probability theory, elementary statistics, advanced topics in statistics, and linear algebra.

1187
1188

2.3.5: Information Systems

1189
1190
1191 As the name suggests, the discipline of information systems (IS) focuses on information (i.e. data in a specific context)
1192 together with information capturing, storage, processing and analysis/interpretation in ways that supports decision

1193 making. The IS field deals with building information processing into organizational procedures and systems that
1194 enable processes as permanent, ongoing capabilities. The discipline emphasizes the importance of building systems
1195 solutions, preferably so that they are continuously improved. At the same time, IS recognizes that in terms of many of
1196 the technical computing knowledge areas and skills, it is heavily dependent on the other computing disciplines.
1197

1198 The *Curriculum Guidelines for Undergraduate Degree Programs in Information Systems 2010* report is also known
1199 as IS2010 [Acm03]. This report emphasizes that information systems as a discipline can make significant contributions
1200 to several domains including business and that its core areas of expertise are highly valuable or essential for the best
1201 practices. The IS discipline focuses on the ability of computing to enable transformative change within domains of
1202 human activity, sometimes called IS environments. That is, IS addresses the ongoing and innovative use of computing
1203 technologies to enable human activities to achieve their goals in ways that are better, faster, cheaper, less painful,
1204 cleaner, or more effective.
1205

1206 Degree programs in information systems always include coursework and other educational experiences in computing
1207 and information technology usually within business environments. IS fosters foundational professional abilities that
1208 are important for all computing disciplines. Given the role of information systems as a bridge builder and integrator,
1209 communication and leadership skills have even more weight than in the context of the other computing disciplines.
1210 Information systems as a discipline has an interest in serving organizational information needs. In the context of
1211 analytics, IS focuses primarily on the integration of analytics into organizational systems.
1212

1213

1214 **2.3.6: Information Technology**

1215

1216 Information technology emphasizes the central role of user needs. The *Information Technology Curricula 2017* report,
1217 known also as IT2017, is globally relevant and informed by educational research [Acm07]. Its task group sought to
1218 balance perspectives from educators, practitioners, and information technology (IT) professionals. The IT2017 report
1219 took a futuristic approach to curricular recommendation and proposed a learner-center framework for programs that
1220 prepare successful IT graduates for professional careers or further their academic study. It eliminated all notions of
1221 topics and learning outcomes, often represented by long lists of knowledge activities. Instead, the task group developed
1222 the use of competencies defined as a combination of knowledge, technical skills, and (human) dispositions. The IT
1223 task group followed pedagogical research and practice similar to what takes place in medical schools.
1224

1225 Degree programs in information technology started to appear in the 1990s. They were a precursor to the discipline
1226 that emerged in the 2000s through the development of the IT2008 curriculum recommendation and accreditation
1227 criteria. IT is a response to the need for professionals with the capability to develop, acquire, maintain and support the
1228 increasingly complex computing technology requirements of modern organizations. Information technology is “the
1229 study of systemic approaches to select, develop, apply, integrate, and administer secure computing technologies to
1230 enable users to accomplish their personal, organizational, and societal goals” [Acm07 p18]. For IT, the primary focus
1231 is on technology, closely aligned with user goals.
1232

1233 In the IT graduate profile specification, the focus is on analysis of problems and user needs, specification of computing
1234 requirements, and design of computing-based solutions. As general professional capabilities, communication, the
1235 ability to make ethically informed judgments, and function effectively as a team member augment this set. Of the
1236 currently identified computing disciplines, IT deals most directly with specific, concrete technology components in
1237 an organizational context.
1238

1239

1240 **2.3.7: Software Engineering**

1241

1242 Software engineering is an engineering discipline that focuses on the development and use of rigorous methods for
1243 designing and constructing software artifacts that will reliably perform specified tasks. The term “software engineer,”
1244 denoting a professional, is much more broadly used than “software engineering” as an academic discipline or a degree
1245 program. There are many more individuals with a job title or professional identity of a “software engineer” than those
1246 who have graduated from software engineering programs. Adding to the confusion, software engineering or software
1247 development is often a part of computer engineering and computer science programs.
1248

1249 The purpose of the *Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in*
1250 *Software Engineering* report, known also as SE2014, is to provide guidance to academic institutions and accreditation
1251 agencies about what should constitute undergraduate software engineering (SE) education [Acm05]. The SE2014
1252 report identified a set of student outcomes describing the qualities of a SE graduate. These include professional
1253 knowledge, technical knowledge, teamwork, end-user awareness, design solutions in context, performance trade-offs,
1254 and continuing professional development. Similarly, the report presented a list of principles “that embraces both
1255 general computing principles as well as those that reflect the special nature of software engineering and that
1256 differentiate it from other computing disciplines.”

1257
1258 Even though SE focuses on creating software-based solutions, it is much more than programming. SE emphasizes the
1259 use of appropriate software development practices and the integration of engineering rigor with the ability to apply
1260 advanced algorithms and data structures developed in computer science. Software engineering has strong focus on the
1261 design of reliable, trustworthy, secure, and usable software systems. The capabilities of trained software engineers
1262 often apply to large-scale systems with high reliability and security requirements such as complex manufacturing
1263 systems, industrial applications, medical devices, autonomous transportation systems, and military solutions.

1264
1265
1266

1267 **2.4: Extensions of Computing Disciplines**

1268
1269
1270
1271
1272
1273

1274 **2.4.1: Computing Interrelationships**

1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288

The discussion in Section 2.3 demonstrates that clear differences exist between the computing disciplines and that they all have distinguishing characteristics that are essential for their identity: CE is the only discipline that focuses on integration of hardware, software and signal processing that are essential in areas such as cyber-physical systems, data communication, or medical imaging. CS has a strong and specific focus on developing strong conceptual foundations and computational capabilities. CSEC explores questions of safety, security, and continuity across the entire computing landscape. DS addresses large-scale data management, storage, and retrieval founded in mathematics and statistics. IS focuses on discovering and implementing positive organizational change using computing capabilities with a special emphasis on value generated by information. IT emphasizes building and maintaining organizational computing infrastructure capabilities and user support. SE addresses large-scale software development processes, particularly in safety and security critical areas.

1289 Information technology platforms and
 1290 infrastructure capture the integration of
 1291 hardware and software into technology
 1292 solutions that enable computing-based
 1293 solutions having capabilities associated
 1294 with data storage, processing, artificial
 1295 intelligence, and visualization.
 1296 Computer engineering, computer
 1297 science, and software engineering
 1298 provide the components required for
 1299 these computing technology
 1300 capabilities to exist. Information
 1301 technology focuses on making and
 1302 keeping them available for individual
 1303 and organizational users. The area of
 1304 digital intelligence and transformation
 1305 covers the capture, management, and
 1306 analysis of data enabling individuals,
 1307 organizations, and societies to conduct
 1308 their activities in a way that helps them
 1309 achieve their goals better. The fields of
 1310 information systems and data science
 1311 enable digital intelligence and
 1312 transformation; these are the processes
 1313 through which organizations change using
 1314 computing capabilities. Figure 2.2 illustrates three computing levels
 1315 (foundations, technology, domain activity) of computing, as well as the way in which cybersecurity permeates
 1316 the entire space of computing.

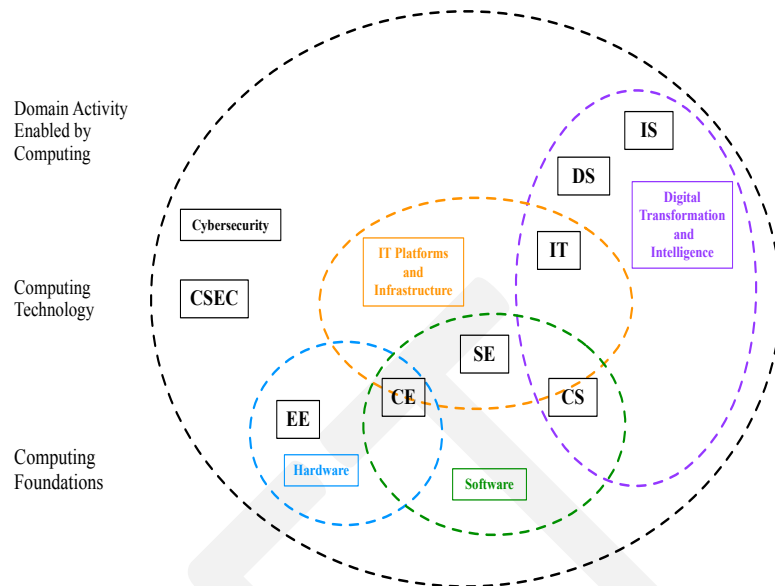


Figure 2.2. A contemporary view of the landscape of computing education
 Legend: CE=computer engineering; CS=computer science; CSEC=cybersecurity;
 DS=data science; EE=electrical engineering; IS=information systems; IT=information
 technology; SE=software engineering

2.4.1: Emerging Curricula

1320 Computing curricula in different forms offer a rich variety of fields that continue to expand rapidly. Consequently, the
 1321 number of educational fields that focus on the intersection of a specific scientific or business domain continues to
 1322 grow. One of the more interesting but also the most complex of the emerging new computing-related disciplines is
 1323 artificial and augmented intelligence (AI). The roots of AI go back to the 1950s, and these areas of computing have
 1324 blossomed during the last ten years. AI and its allied field of robotics have become highly popular fields of study in
 1325 computing. Although at the time of this writing no formal professionally endorsed AI curriculum exists, a curricular
 1326 recommendation in these areas has the potential to emerge in the next few years.

1328 Current curricular areas that have emerged in recent times include cloud computing, smart cities, sustainability,
 1329 parallel computing, internet of things, and edge computing. Additionally, the predicted top-ten emerging computing
 1330 trends are (a) deep learning (DL) and machine learning (ML), (b) digital currencies, (c) blockchain, (d) industrial IoT,
 1331 (e) robotics, (f) assisted transportation, (g) assisted/augmented reality and virtual reality (AR/VR), (h) ethics, laws,
 1332 and policies for privacy, security, and liability, (i) accelerators and 3D, and (j) cybersecurity and AI [Iee1]. All these
 1333 areas have some coverage within existing curricula guidelines, and some areas (e.g., cybersecurity) even have its own
 1334 formal guidelines. Other areas include 3D graphics and accelerators. One can only guess whether these top-ten trends
 1335 will still be viable over the next dozen years.

2.4.2: Computing + X

1340 A growing interest in recent times is the development of computing programs (e.g., software engineering or
 1341 information systems) that have an extension to a non-computing discipline (e.g., avionics or finance). We call this
 1342 “Computing + X” where ‘Computing’ represents one of the computing disciplines and ‘X’ is a non-computing
 1343 discipline. This mode of learning has the goal of integrating a non-computing area of study as an extension of a
 1344 computing area. For example, if X is linguistics, then CE + X represents a linguistic extension of a computer

1345 engineering program. Such programs allow students to pursue their computing interests in other academic fields.
1346 It also allows computing students to pursue flexible programs of study that incorporate a strong grounding in a
1347 computing discipline with technical or professional exposure in other fields.
1348

1349 The relatively recent initiatives surrounding Computing + X is nothing new. For decades, computing programs had
1350 offered tracks, concentrations, or minors in a variety of subject areas to expand the knowledge base of students for
1351 computing programs. These programs continue today. However, the level of interest in the longtime practice has
1352 increased. So, the Computing + X phenomenon continues where X could be in areas such as astronomy, chemistry,
1353 economics, languages, linguistics, music, and other computing extensions. Computing + X allows students to discover
1354 transformational relationships between computing and non-computing fields. Degrees in this category often have the
1355 term ‘informatics’ included in them such as medical informatics, health informatics, legal informatics, bioinformatics,
1356 or chemical informatics. In many ways, information systems was the original “Computing + X” discipline, integrating
1357 computing primarily with business to transform the way businesses and other enterprises operate.
1358
1359

1360 **2.4.3: X + Computing**

1361

1362 Computing is ubiquitous with application areas in almost every field imaginable. Therefore, the study of
1363 computing in other disciplines arises naturally. That is, computing becomes an extension to an established
1364 discipline of study. This representation is “X + Computing” where ‘X’ is the established non-computing domain
1365 usually in science, business, or humanities. For example, a program in computational biology would have its
1366 roots in some aspect of biology augmented by a study in computing related to it. Computational finance is
1367 another example where computing becomes an extension of finance.
1368

1369 As before, for decades, non-computing programs had offered tracks, concentrations, or minors in a variety of subject
1370 areas to expand the knowledge base of students from non-computing programs. The “X + Computing” practice
1371 continues today where ‘X’ could be principal interest areas such as accounting, biology, art, or other computing
1372 extensions. “X + Computing” allows non-computing students to discover transformational relationships between their
1373 principal area of study and computing. Hence, “X + Computing” is different from “Computing + X” because in the
1374 former, the base area is a non-computing discipline (e.g., chemistry) while in the latter, the base area is a computing
1375 discipline (e.g., computer engineering).
1376

1377 Whether it is X + Computing or Computing + X, both designations reflect the impact computing has had across a
1378 broad range of other, non-computing domains. Not surprisingly, in the German-speaking world, the term business
1379 informatics (Wirtschaftsinformatik) is used for degree programs that are globally aligned with those in information
1380 systems [Hel1]. In all extended degree programs, the fundamental question is the same: how do computing and
1381 computational thinking transform the way in which those working within non-computing domain X achieve their
1382 goals? In practice, this mode of thinking requires three types of abilities: domain, computing, and integrative related
1383 to the transformative opportunities offered by the computing field. Beyond graduation, graduates may benefit from
1384 interdisciplinary studies and lifelong learning.
1385

1386 There are many similar examples to represent other fields with computing. For example, high-performance computing
1387 (HPC) field is a representative field that is interdisciplinary. The students working for the HPC field not only need
1388 computing field knowledge, but also domain-specific knowledge as well as supercomputer system maintenance
1389 knowledge.
1390

1392 **2.4.4: Other Tertiary Computing Models**

1393

1394 In addition to the traditional three- or four-year baccalaureate program models, there are thousands of academic
1395 institutions around the world offering non-traditional educational programs in computing such as the two-year colleges
1396 in the United States. According to ACM’s Educational Policy Committee’s “Lighting the Path” report of 2018, 41%
1397 of all U.S. post-secondary students attend community colleges and the numbers are somewhat higher for most minority
1398 populations [Acm16]. Currently, ACM offers curriculum recommendations for two-year programs in information
1399 technology [Acm09] and computer science [Acm10]. A recommendation for two-year cybersecurity programs is
1400 currently under development.

1401
1402 Education in computing has been a shared interest of traditional universities and other academic institutions, other
1403 education providers, and employers. Even though specific research-based data are difficult to find, the number of
1404 people employed in computing-related jobs is much higher than those with an actual degree in computing [Nas2, p7].
1405 A recent national academies report [Nas2, App C] refers to transfers from other fields of study and immigration as
1406 sources of employees to fill the gap in the context of computer science. The report does not, however, discuss the
1407 other broadly defined computing-related job roles, but anecdotally it is common knowledge that many individuals
1408 with an educational background in other fields (or those without a completed higher education degree) serve
1409 successfully in computing-related positions (e.g., IBM’s “New Collar” jobs).

1410
1411 There are at least four commonly used pathways where a person with an academic background in a different field
1412 gains the proficiencies needed to perform well in a computing-related field. These are as follows [Dab1, Per3, Wag1].

- 1413 ▪ Self-study without any formal educational sources.
- 1414 ▪ Self-study using extensive computing education resources available online for free or at a very low cost,
1415 including providers such as various open universities, Udemy, EdX, Coursera, Khan Academy, and
1416 SkillShare.
- 1417 ▪ Coding bootcamps that are typically 8-12 weeks in length, intensive, and solely focused on providing their
1418 participants with software development and related skills that provide students with immediate
1419 employability.
- 1420 ▪ Specialized schools in coding, software engineering, or other related schools. Examples of this category are
1421 École 42 in Paris, 42 Silicon Valley, and the recently launched Hive Helsinki. Typical descriptions of these
1422 institutions are that they operate very closely with industry partners and declare themselves to be operating
1423 without teachers and without courses and classes.

1424 There are many ways to transition into the computing field and to become successful in doing so. Vendor certifications
1425 such as Cisco, CompTIA, and Microsoft offer certifications to achieve this goal. Additionally, some universities offer
1426 tandem pathways to graduate school and governments offer security certifications for professional enhancement.

1427
1428

1429 **2.4.5: Computing in Primary and Secondary Education**

1430

1431 The computing education community around the world has done extensive work to improve the availability and quality
1432 of computing-related courses in primary and secondary education, with a specific focus on improving the diversity of
1433 students selecting computing as their career. Some examples of influential actors in this area include the following
1434 activities.

- 1435 ▪ *Computer Science Principles* is a year-long course offered in United States high schools that introduces
1436 students to the foundational concepts of computer science and challenges them to explore how computing
1437 and technology can impact the world. It is a rigorous, engaging, and approachable course that explores many
1438 of the foundational ideas of computing, so all students understand how these concepts are transforming the
1439 world in which we live. It is an Advanced Placement (AP) course designed and administered by the College
1440 Board. Code.org is an approved AP CS Principles provider.
- 1441 ▪ *Code.org* is a not-for-profit organization founded by Hadi Partovi that focuses on “expanding access to
1442 computer science in schools and increasing participation by women and underrepresented minorities.”
1443 Among other activities, code.org organizes an annual “Hour of Code” event with millions of participants
1444 annually, offers a library of computer science courses for primary and secondary education, and advocates
1445 computer science education with policy makers, primarily in the United States.
- 1446 ▪ *CSforAll* is a hub for the national “Computer Science for All” movement, which works to enable all students
1447 in grades kindergarten through twelfth year (K-12) to achieve computer science literacy as an integral part
1448 of their educational experience. It has currently 355 member organizations, including content providers,
1449 education associations, and both companies and non-profits as funders.
- 1450 ▪ *Computer Science Teachers Association (CSTA)* is a membership organization for primary and secondary
1451 education teachers in computer science with more than 25,000 members in 145 countries. CSTA’s mission
1452 is to “empower, engage and advocate for K-12 CS teachers worldwide.” ACM established it in 2004.

1453

1454 In addition, for example, CSpathshala is an ACM India education initiative to bring a modern computing curriculum
1455 to Indian schools [Csp1]. Launched in 2016, CSpathshala has developed an unplugged curriculum to teach
1456 computational thinking (CT) without the use of computers along with sample teaching aids for the first eight years of

1457 school in India. More than 300,000 students largely from rural government schools in India are learning computational
1458 thinking using the CSpathshala curriculum. The draft “National Education Policy 2019,” recently released by the
1459 Indian government, also recognizes CT as a fundamental skill and recommends teaching CT from age six using well-
1460 designed worksheets. Similar initiatives include those in Finland, New Zealand, Sweden, and the United Kingdom.
1461 Computing in the Middle East is described in section 6.3.7.

1462
1463

1464 **2.4.6: Computing Specialization**

1465

1466 Many specializations exist in computing. One area that goes back to the 1940s is scientific computing—considered to
1467 consists of algorithms and the associated methods for computing discrete approximations used to solving problems
1468 involving continuous mathematics. Numerical methods and computational science are other names used for this area
1469 which deals with mathematical models to solve problems, methods for system optimization, and computing
1470 infrastructure in support of engineering and science problems.

1471

1472 Another such area is digital game design and development (DGDD). Another is media development. In the United
1473 States alone, more than five hundred DGDD programs currently exist [Are1], and many more exist worldwide.
1474 Curricular efforts in game and media programs are ongoing. The game and media industries develop specialized
1475 hardware and software that are becoming apparent in higher education. The industry stands at about \$43.4B in the
1476 United States alone [The1, Deal] thereby making this emerging area a worldwide phenomenon.

1477

1478 In the future, the world should expect to see increasing specialization on the development of core computational
1479 capabilities within computer science and software engineering (software) and especially in computer engineering
1480 (integration of hardware and software). The number of types of computing degree programs should also increase
1481 dramatically that focus on the transformation of computing programs into specific domains of human activity (e.g.,
1482 information systems and Computing + X) as well as a greater emersion of computing in existing domains or other
1483 disciplines (e.g., X + Computing). The world should also witness more degree program types for specialized pervasive
1484 themes with a broad ranging effect across multiple domains (e.g., cybersecurity, data analytics, artificial intelligence),
1485 as well as continued contributions of degree programs that prepare professionals for roles focused on organizational
1486 computing infrastructure (e.g., information technology).

1487

1488

1489

1490 **2.5: Digest of Chapter 2**

1491

1492 This chapter examines the continuing evolution of computing education. In the context of undergraduate programs,
1493 computing can refer to a family of study areas corresponding to the discipline reports for computer engineering,
1494 computer science, cybersecurity, data science, information systems, information technology, and software
1495 engineering degrees that have been developed by the ACM and IEEE-CS with AIS in recent decades. On the other
1496 hand, the changing landscape of computing has now led to the recognition of the importance of information security
1497 and data as a resource for decision making. Within the computing education landscape, there exist a rich variety of
1498 fields that continue to broaden, including opportunities for Computing+X and X+Computing degrees, and tertiary
1499 models for computing programs. Around the world, computing education has also expanded into primary and
1500 secondary schools. At the same time, specialization areas such as scientific computing or digital game design have
1501 led to new degree programs, a trend that should continue.

1502

1503

1504

1505 **Chapter 3: Knowledge-based Computing Education**

1506

1507 The philosophical underpinnings of the CC2020 project treats computing as a meta-discipline – a collection of
1508 disciplines having a central focus of computing. This chapter explains the concept of knowledge-based learning and
1509 how it has encompassed computing education over decades. It reviews the CC2005 report which is primarily a
1510 knowledge-based document. Additionally, it addresses how workplace and employment dynamics affect knowledge-
1511 based learning and related issues.

1512

1513

1514

1515 **3.1: Knowledge-Based Learning**

1516

1517 This section addresses some of the underpinnings of knowledge-based learning (KBL). It explores the definitions of
1518 learning and knowledge, the attributes of KBL, and the relationship between KBL and computing curricula.

1519

1520

1521 **3.1.1: Learning and Knowledge**

1522

1523 Before engaging in knowledge-based learning, it would be useful to first understand the contextual meaning of
1524 learning and knowledge. The word *learning* refers to the endeavor of “knowledge or skill acquired by instruction or
1525 study” [Mer3] often in an environment conducive to the activity. In turn, the word *knowledge* refers to the
1526 “acquaintance with or understanding of a science, art, or technique” [Mer4] or more encompassing, “the sum of what
1527 is known : the body of truth, information, and principles acquired by humankind.”

1528

1529 There is an inextricable connection between the two words knowledge and learning. The former refers to content while
1530 the latter refers to activity. Thus, people acquire content and skills through the process of learning. Humans acquire
1531 (learn) content (knowledge) continuously, almost from the time of birth. For the purposes of this report, content
1532 acquisition refers to learning in formal settings or structures such as in classrooms or online environments.

1533

1534 Recently, the term *content knowledge* has come into use, which refers to the body of knowledge and information that
1535 teachers teach and that students should learn in a subject or content area. Content knowledge generally refers to the
1536 facts, concepts, theories, and principles taught and learned in specific academic courses [Edg1]. This form of
1537 knowledge occurs in core courses of study, curriculum, or learning standards.

1538

1539

1540 **3.1.2: Learning from Knowledge Contexts**

1541

1542 In general, learning occurs by building on knowledge a person already has. That is, a person, namely a student
1543 scaffolds new knowledge based on a student’s existing knowledge. *Knowledge-based learning* (KBL) depicts this
1544 form of learning activity. More formally, “knowledge-based learning is learning that revolves around both the
1545 knowledge that the student already has, and the understanding that they are going to achieve by doing work” [Tes1].

1546

1547 Students, teachers, and the public have all experienced knowledge-based learning. Basic schooling allows
1548 advancement from one grade level to the next based on the verified knowledge acquired in one grade before
1549 advancement takes place. Often, verification takes place by evaluating students’ knowledge content through tests, oral
1550 or written examinations, interviews, and other tools useful in assessing whether a student has achieved the expected
1551 knowledge base for a given level. At universities, the use of course prerequisites attempt to ensure that a student has
1552 to necessary knowledge base to advance to the next course level.

1553

1554 Knowledge-based learning has existed for millennia. Whether formally or informally, KBL has used approaches to
1555 elevate human knowledge on a global scale. Teachers deliver information to learners and then check their level of
1556 attainment. Reflective learners can assess themselves on the acquired new knowledge. Teachers direct learners on
1557 what they need to know and check whether they learned it. Using this approach and providing reliable comment,
1558 teachers can help students see where they have learned or where they have erred.

1559
1560
1561
1562
1563
1564
1565
1566

Knowledge-based learning enjoys many benefits. It builds on learners' existing knowledge; it helps learners see how they are progressing, and it helps them highlight gaps in their knowledge. With clear learning objectives, learners can see how their existing knowledge will help them to complete the task [Dso1]. Practicing knowledge-based techniques can identify where learners require more emphasis. By building on the knowledge a person already has, KBL lifts learners' confidence by showing them that they have the knowledge they need to finish a task [Icd1].

1567 **3.1.3: Knowledge and Computing Education**

1568
1569
1570
1571
1572
1573
1574

For computing and other disciplines, *knowledge* has always been the focus of the study area. Computing curricular reports often describe a discipline through knowledge areas (KAs), knowledge units (KUs), and learning outcomes (LOs). Sometimes, people refer to this phenomenon as the KA-KU-LO or Ka-Ku-Lo syndrome, with lists of topics associated with each knowledge unit. These curricular reports generally do not provide guidance related to skills or guidance related to human behavior particularly reflected by performance in the workplace.

1575
1576
1577
1578
1579
1580

These documents reflected the concept of knowledge-based learning, viewed traditionally as a form of learning that involves knowledge students learned and already have, together with the understanding that they are going to achieve through work [Cla1]. That is, teachers transfer knowledge to students through experience, notes, textbooks or other means; having received the knowledge, students have an expectation of achievement because of it and work toward demonstrating that achievement. Almost all universities worldwide produce graduates through knowledge-based learning.

1581
1582
1583
1584
1585
1586
1587
1588
1589

However, the traditional knowledge-based learning paradigm may no longer be appropriate. Technology now influences new ways of learning. Students use many non-traditional learning formats, thereby challenging traditional methods. Furthermore, universities produce computing graduates who may be intellectually smart, but have difficulties in workplace settings. While knowledge-based learning is commendable, it is less effective and may not be useful when technical skills and human behaviors occur in a changing computing and engineering world.

1590 **3.2: Computing Curricula 2005**

1591
1592
1593
1594
1595
1596

The CC2005 report provided readers with an overview of five undergraduate computing degree programs that were available in the early 2000s. At that time, five computing curricula reports were in existence, which included computer engineering (CE2004), computer science (CS2001), information systems (IS2002), information technology (a work in progress published in IT2008), and software engineering (SE2004). These computing fields were related but quite different from each other.

1597
1598
1599
1600
1601
1602
1603
1604

The CC2005 report explained the character of the various undergraduate degree programs in computing and assisted people to determine which programs best suited their goals and circumstances. Beneficiaries of the report included recruiters from industry and government, students and potential students, university faculty members and administrators who were developing plans and curricula for computing-related programs at their institutions. In addition, beneficiaries include those interested in accreditation of computing programs, and responsible parties in public education including boards of education, government officials, elected representatives, and others who seek to represent the public interest.

1605
1606
1607

1607 **3.2.1: Intent of CC2005**

1608
1609
1610
1611
1612
1613

Reliable information about the number of different types of computing degree programs was difficult to ascertain in the early 2000s. Hence, the focus on just five prominent computing programs (CE, CS, IS, IT, SE) satisfied the committee's criterion for proper inclusion to distinguish undergraduate curricular guidelines. These five computing areas represented the majority all undergraduates specializing or majoring in computing. Notwithstanding, at that time, the committee expected additional computing disciplines to generate curricular expansion as extensions to the report.

1614 Candidate programs for future editions could include new fields that did not yet exist or established fields that did not
1615 have generally accepted curricular guidelines. In the end, each one of the five discipline-specific curricular volumes
1616 reflected in the CC2005 report represented the best judgment of the relevant professional, scientific, and educational
1617 associations and served as a definition of what these five computing degree programs should be.
1618

1619 The CC2005 committee made no effort to update the contents of existing curricular reports as that effort was beyond
1620 its mission and authority. Rather, the committee had reviewed the five curricular volumes, compared their contents to
1621 one another, and synthesized what they believed to be essential descriptive and comparative information. In addition
1622 to using the five curricular reports as the basis for the CC2005 report, the committee had referred to the computing
1623 professional organizations and other supporting information as necessary. The committee did not focus on other types
1624 of undergraduate computing degree programs (e.g., associate degree or similar programs), graduate education in
1625 computing, computing research communities, or nontraditional computing education such as vendor-specific
1626 certification programs. Additionally, the CC2005 committee realized that computing itself will continue to evolve and
1627 new computing-related fields would likely emerge.
1628

1629

1630 **3.2.2: Content of CC2005**

1631

1632 The most significant part of the report is the definitive articulation of the five computing disciplines. As mentioned
1633 earlier, these disciplines included computer engineering, computer science, information systems, information
1634 technology and software engineering. In addition to addressing the landscape of computing in the early 2000s, the
1635 report defined the meaning of “computing” and provided a brief history of the evolution of computing before, during,
1636 and after the 1990s as shown earlier in Figure 2.1 It then described (and defined) each of the five computing disciplines
1637 followed by graphical visuals for these five disciplines. Discussion of the graphical visuals occurs later in this chapter.
1638

1639 One of the useful aspects of CC2005 was the discussion on the expectations of graduates for the degree programs.
1640 The discussion revolved on two themes. One theme dealt with curricular summaries as a comparison of degree
1641 programs with an interpretation of the tabular representation and suggestions on its use. The other theme focused on
1642 expected degree outcomes with an expected comparison of degree graduates. Both these tabular representations are
1643 useful elements to contrast the outcomes of the five computing degrees. Since the CC2005 report was largely U.S.
1644 centric, it was important to acknowledge global differences in the content of the report.
1645

1646 The CC2005 report also acknowledged the rapid pace of change in academia and how computing might affect the
1647 offered degrees, specifically in the five focus degree areas of computer engineering, computer science, information
1648 systems, information technology, and software engineering. The pace of change particularly reflects the changes in
1649 the workplace where change is continuous. Because of this change, computing degree programs should be responsive
1650 to such variations.
1651

1652 Additionally, the CC2005 report addressed institutional considerations such as the evolution of degree programs and
1653 strategies to monitor course portfolios. It addressed diversity challenges, faculty development, adaptation, as well as
1654 organizational and curricular structures. Coupled with curricular response to market forces and academic integrity, the
1655 report discussed aspects of quality assurance and program accreditation as it exists in the United States and the United
1656 Kingdom. Regardless of the metrics used, all would agree that program quality should be paramount for all computing
1657 disciplines.
1658

1659

1660

1661 **3.2.3: Comparison Tables**

1662

1663 CC2005 provided a comparative view of the emphasis on computing topics among the five types of computing degree
1664 programs. A comparison table provided a summary of the topics studied at the undergraduate level in one or more of
1665 the computing disciplines, presented in its first column. The remaining columns showed the numerical values per topic
1666 for each of the five types of computing degree programs. These values range between 0 (lowest) and 5 (highest) and
1667 they represent the expected relative emphasis each type of computing degree program might place on each given topic.
1668 In the CC2005 report, its Table 3.1 showed this comparison. An update of this table appears in Chapter 4 of this report.

1669 In addition to this comparison table, the CC2005 report provided a table showing the relative performance capabilities
 1670 of computing graduates by discipline [Acm02 p28 Tab3.3]. This table focused on outputs, summarizing the relative
 1671 capability expectations of computing graduates. Table 3.1 shows a copy of that table.
 1672
 1673

Table 3.2. Computing Graduate Profiles

Area	Performance Capability	CE	CS	IS	IT	SE
Algorithms	Prove theoretical results	3	5	1	0	3
	Develop solutions to programming problems	3	5	1	1	3
	Develop proof-of-concept programs	3	5	3	1	3
	Determine if faster solutions possible	3	5	1	1	3
Application programs	Design a word processor program	3	4	1	0	4
	Use word processor features well	3	3	5	5	3
	Train and support word processor users	2	2	4	5	2
	Design a spreadsheet program (e.g., Excel)	3	4	1	0	4
	Use spreadsheet features well	2	2	5	5	3
Computer programming	Train and support spreadsheet users	2	2	4	5	2
	Do small-scale programming	5	5	3	3	5
	Do large-scale programming	3	4	2	2	5
	Do systems programming	4	4	1	1	4
	Develop new software systems	3	4	3	1	5
	Create safety-critical systems	4	3	0	0	5
Hardware and devices	Manage safety-critical projects	3	2	0	0	5
	Design embedded systems	5	1	0	0	1
	Implement embedded systems	5	2	1	1	3
	Design computer peripherals	5	1	0	0	1
	Design complex sensor systems	5	1	0	0	1
	Design a chip	5	1	0	0	1
	Program a chip	5	1	0	0	1
Human-computer interface	Design a computer	5	1	0	0	1
	Create a software user interface	3	4	4	5	4
	Produce graphics or game software	2	5	0	0	5
	Design a human-friendly device	4	2	0	1	3
Information systems	Define information system requirements	2	2	5	3	4
	Design information systems	2	3	5	3	3
	Implement information systems	3	3	4	3	5
	Train users to use information systems	1	1	4	5	1
	Maintain and modify information systems	3	3	5	4	3
Information management (Database)	Design a database mgt system (e.g., Oracle)	2	5	1	0	4
	Model and design a database	2	2	5	5	2
	Implement information retrieval software	1	5	3	3	4
	Select database products	1	3	5	5	3
	Configure database products	1	2	5	5	2
	Manage databases	1	2	5	5	2
IT resource planning	Train and support database users	2	2	5	5	2
	Develop corporate information plan	0	0	5	3	0
	Develop computer resource plan	2	2	5	5	2
	Schedule/budget resource upgrades	2	2	5	5	2
	Install/upgrade computers	4	3	3	5	3
Intelligent systems	Install/upgrade computer software	3	3	3	5	3
	Design auto-reasoning systems	2	4	0	0	2
	Implement intelligent systems	2	4	0	0	4
Networking and communications	Design network configuration	3	3	3	4	2
	Select network components	2	2	4	5	2
	Install computer network	2	1	3	5	2
	Manage computer networks	3	3	3	5	3
	Implement communication software	5	4	1	1	4
	Manage communication resources	1	0	3	5	0
	Implement mobile computing system	5	3	0	1	3
Systems Development Through Integration	Manage mobile computing resources	3	2	2	4	2
	Manage an organization's web presence	2	2	4	5	2
	Configure & integrate e-commerce software	2	3	4	5	4
	Develop multimedia solutions	2	3	4	5	3
	Configure & integrate e-learning systems	1	2	5	5	3
	Develop business solutions	1	2	5	3	2
	Evaluate new forms of search engine	2	4	4	4	4

1674
 1675
 1676 **3.2.4: Curricular Visuals**

1677
 1678 One highlight of the CC2005 report is the two-dimensional visual graphics that depicted the five computing
 1679 disciplines. These graphics illustrated the commonalities and differences among computing disciplines. Their
 1680 dimensions highlighted the relative degree to which a computing discipline focused on theory versus practice; it also
 1681 highlighted the degree to which a computing discipline focused on hardware versus humans. They suggested how

1682 each discipline occupies the problem space framework of computing as shown in Figure 3.1. The focus is on what
 1683 students in each of the disciplines typically do after graduation, not on all topics a student might study. Some
 1684 individuals will have career roles that go beyond the scenarios described by these snapshots.
 1685

1686 The horizontal range runs from theory, principles, and innovation on the left, to application, deployment, configuration
 1687 on the right. Thus, someone who likes the idea of inventing new things or enjoys a university setting to develop new
 1688 principles will want to work in a discipline that occupies the space to the left. Conversely, someone who wants to help
 1689 people choose and use appropriate technology or who wants to integrate off-the-shelf products to solve organizational
 1690 problems will want an area that occupies space to the right. Because there are many kinds of job tasks that fall between
 1691 the extremes, one should not just look only at the far left and far right but consider possibilities between the extremes.
 1692

1693 The vertical axis runs from computer hardware and architecture at the bottom
 1694 to organizational issues and information systems at the top. As we go up this axis,
 1695 the focus changes from wires, hardware,
 1696 chips, and circuits at the bottom to
 1697 people, information, and organizational
 1698 issues at the top. Thus, someone who
 1699 likes designing circuits or is curious
 1700 about a computer’s inner workings will
 1701 care about the lower portion of the
 1702 diagram; someone who wants to see how
 1703 technology can work for people or who is
 1704 curious about how technology affects
 1705 organizations will care about the upper
 1706 portions. We can consider the horizontal
 1707 and vertical dimensions together.
 1708 Someone who cares about making things
 1709 work for people and is more interested in
 1710 devices than organizations will be
 1711 interested in the lower right, while
 1712 someone who wants to develop new
 1713 theories about how information affects
 1714 organizations will be interested in the
 1715 upper-left area of the diagram.
 1716
 1717
 1718

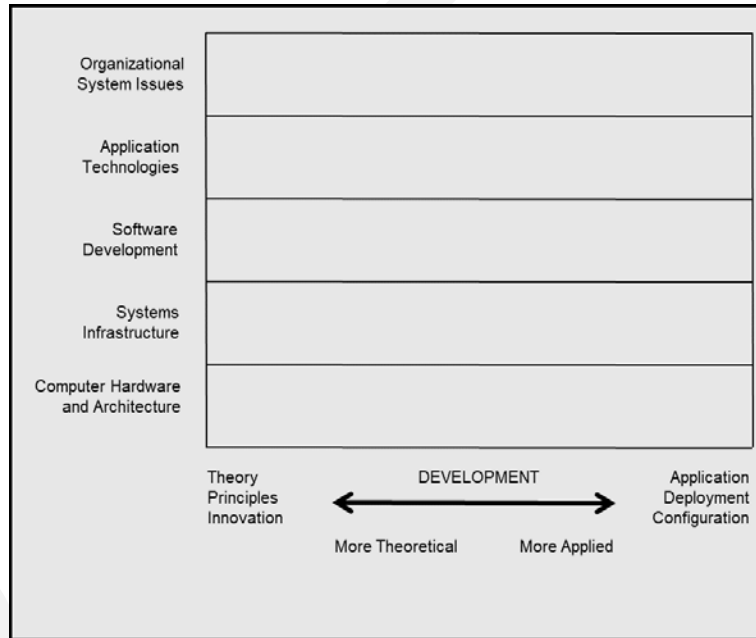


Figure 3.1. Problem Space Framework (CC2005)

1719 Figure 3.2 provides curricular illustrations that sketches the conceptual territory occupied by each of the five
 1720 computing disciplines. These are informal illustrations used to communicate the CC2005 task force’s subjective
 1721 interpretation of the various disciplines. They are not based on any precise quantitative foundation. Furthermore, they
 1722 show only computing interests of themes. Computer engineering occupies a broad area across the bottom because
 1723 computer engineering covers the range from theory and principles to the practical application of designing and
 1724 implementing products using hardware and software. Computer science covers most of the vertical space between the
 1725 extreme top and extreme bottom because computer science generally deals with theory and software development
 1726 such as operating systems and web browsers. Information systems occupies the shaded area across most of the top-
 1727 most level because it concerns the relationship between computing systems and the organizations they serve and often
 1728 tailor application technologies to the needs of the enterprise. Information technology covers the shaded area along the
 1729 top right edge because it focuses on the application, deployment, and configuration needs of organizations and people
 1730 over a wide spectrum. Software engineering spans the entire horizontal dimension at middle of the diagram because
 1731 the subject covers a wide range of large-scale software applications with respect to the systematic development of
 1732 software. The images from Figure 3.2 have received worldwide acclaim in computing educational circles. They have
 1733 appeared in many contexts such as the poster shown in Appendix A.
 1734
 1735
 1736
 1737

1738
1739

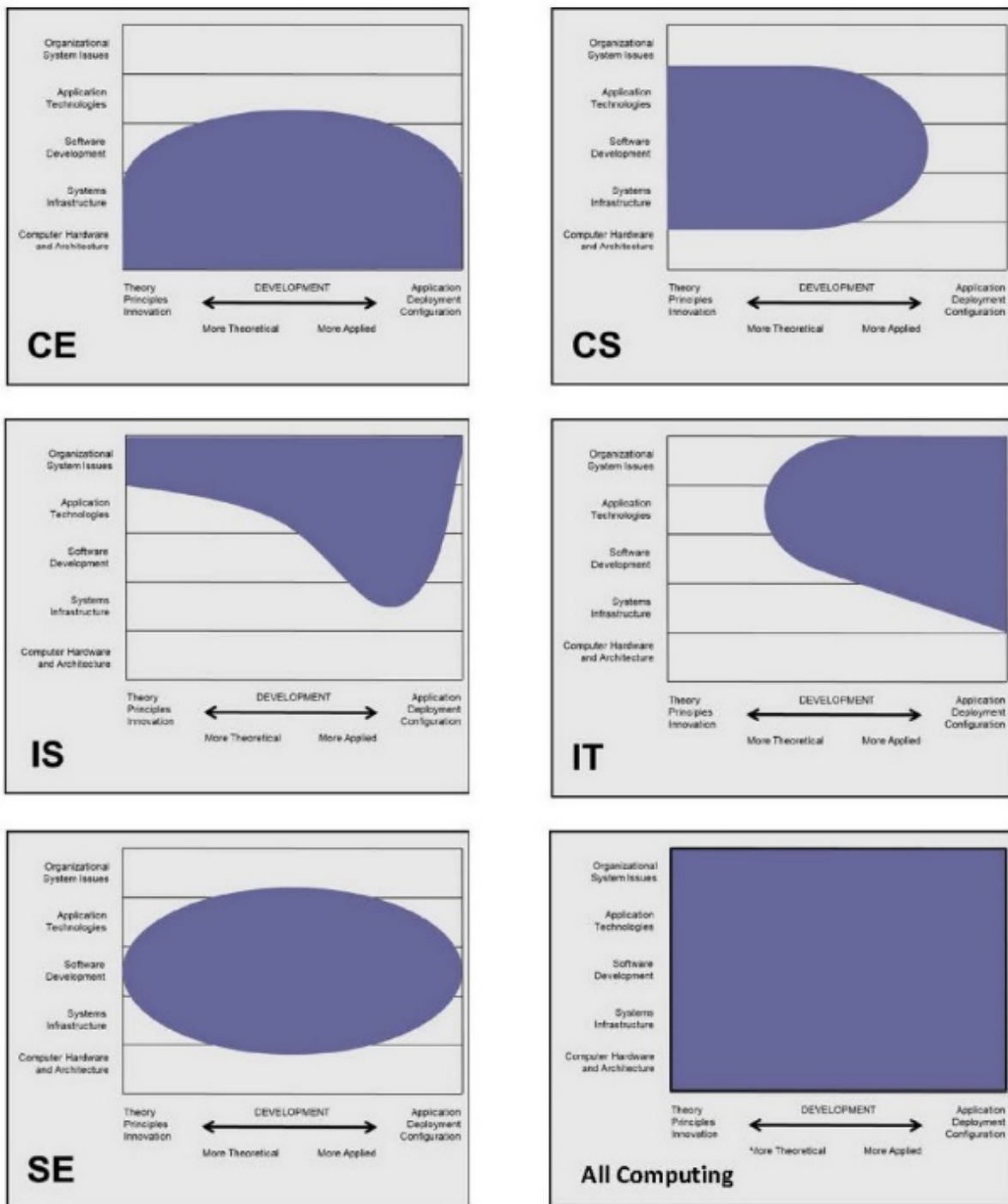


Figure 3.2. Visuals from CC2005

1740
1741
1742
1743

1744 **3.2.5: Global and Other Considerations**

1745
1746 The CC2005 report and the associated five volumes of the Computing Curricula Series benefited to some degree from
1747 international input. Notwithstanding, the CC2005 task force was very conscious of the void of encompassing greater
1748 global contributions to its work. The task force recognized that future efforts must feature significantly expanded
1749 international participation. Some differences include the structure of the academic year, the emphasis given to the
1750 study of computing within a degree program, and quality control mechanisms such as different expectations and
1751 practices regarding accreditation. In addition, there were different approaches to defining the focus of degree
1752 programs.

1753
1754 The CC2005 report addressed other issues useful for a global computing community. The CC2005 task force
1755 recognized that the computing field is evolving and as a result, it provided some suggestions for academia to keep in
1756 step with the pace of change. It also recognized that the pace of change that exists in the workplace whereby graduates
1757 of computing-degree programs should have the ability to fulfill their own career opportunities. In addition, the CC2005
1758 report addressed institutional considerations and urged institutions to be mindful of the evolution of computing degree
1759 programs, to initiate strategic approaches to manage change, and to approach diversity through faculty adaptation and
1760 development as well as organizational and curricular structures.

1761
1762 The CC2005 report received universal acceptance as a document to differentiate computing degree programs at the
1763 time. Educators, students, and industry professionals are familiar with the illustrations shown in Figure 3.2 and the
1764 poster in Appendix A. Overall, the CC2005 project was a positive contribution to students, to industry, and to the
1765 computing academic communities.

1766
1767
1768

1769 **3.3: Dynamics and Computing Degree Programs**

1770
1771 CC2005 reflected a knowledge-based view of computing education. This view resulted in the ability to compare
1772 specializations with respect to the types of knowledge that they contain, in models already shown in Figure 3.2. Such
1773 a view has been helpful in establishing the course structure of curricula within the various specializations, and it
1774 reflects the traditional model of education in that regard. In such a view, curricula reflect topics taught within a
1775 conglomeration of courses, but the skills learned within those courses depends heavily on the design of the individual
1776 course. Two wildly different curricula in terms of skills learned could both meet the same knowledge-based curricular
1777 requirements.

1778
1779

1780 **3.3.1: Recent Workplace Trends**

1781
1782 The variability stated above is a ubiquitous consequence of classical education in most disciplines. However, for jobs
1783 that require certain skills, it means that recent computing graduates may not have those skills even though they
1784 graduate from a curriculum that meets prescribed knowledge-based requirements. This has classically put the onus on
1785 industry to do training to meet the requirements of their workforce. However, with the fast-paced dynamics of change
1786 in the computing fields, industry can no longer afford to train recent graduates of computing programs. Some industries
1787 and companies expect to have performance (and profit) almost immediately after hiring. As the computing field
1788 evolves in its importance to the world economy, computing professionals must apply their skills in a broad range of
1789 diverse career sectors.

1790
1791 People seeking computing careers will have a strong potential for success if they possess relevant skills and appropriate
1792 temperament. A recent study by the United States Bureau of Labor Statistics (BLS) estimates that by 2024, computing
1793 employment in the United States will increase by 12% [Bls1], with information security leading by 36.5% [Bls2].
1794 Employment growth for information security analysts projected for 2014-2024 is 18%. Other computing occupations
1795 have larger projected growth: application software developers (19%), computer systems analysts (21%), and web
1796 developers (27%). Figure 3.3 presents these data.

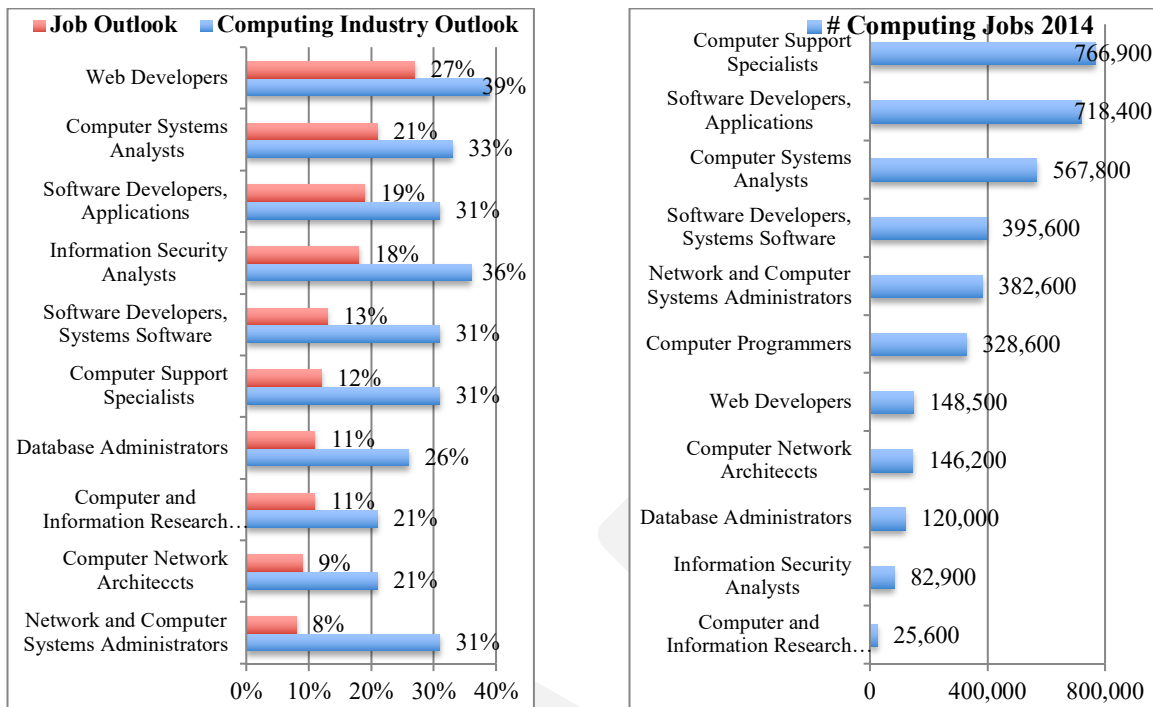


Figure 3.3. Left: Computing occupations projected growth 2014-2024 across all sectors (job outlook) and in the computing sector. Right: Computing jobs in 2014. (Courtesy of Bureau of Labor Statistics)

1797 Unfortunately, although jobs are and will be available, employers find it increasingly difficult to find qualified people
 1798 to fill them. Students graduating from computing programs often do not have the attributes to fill the needs of industry.
 1799 Perhaps they have knowledge or some technical skills acquired from their studies. However, they often lack other
 1800 skills (e.g., communication, teamwork, time management) needed “to fit” within an industry or government
 1801 environment. In other cases, individuals may possess typical technical skills, but do not have a grasp over more
 1802 complex analytical and troubleshooting skills.
 1803
 1804

1805 3.3.2: Forces Shaping Employment Markets

1806 To address the skill challenge, industry market forces have created robust alternatives to compete with computing
 1807 degrees. Some of these forces include vendor and non-vendor certifications, industry in-service training, specialized
 1808 commercial schools, and similar marketable activities. Commercial certifications have had some influence on
 1809 academic institutions. Some of these certifications have even become part of some university computing curricula
 1810 such as acquiring network certification in an academic networking class. Individuals who complete certification exams
 1811 can use these credentials to supplement the value of their academic education to potential employers. Employees can
 1812 use certifications to demonstrate their job readiness and pursuit of extra-curricular activities to demonstrate IT skills
 1813 to potential employers. Table 3.3 lists some leading computing certifications for 2017 compiled by the CRN media
 1814 outlet [Nov1]. The CC2020 public website will show a more inclusive collection of certifications on a global scale.
 1815
 1816

Table 3.3. Common Computing Certifications

Entry-level networking and security (CompTIA, Cisco)	Project management (Project Management Institute, Axelos)
Professional networking and routing and switching (Cisco, Citrix)	Security (ISC2)
Virtualization and networking (Citrix VMWare)	Security management (ISC2)
Windows servers and infrastructure (Microsoft)	Cloud computing (Amazon)
IT service management (Axelos)	Risk management (ISACA)
	IT auditing (ISACA)

1817
1818 Some employers offer in-service training within their own facilities. These training courses or seminars are often
1819 highly focused on some specialty indigenous to company's work. Such focused training activities have little influence
1820 on academic computing degree programs because of the narrow and specialized concentration. Additionally, it is quite
1821 common to find commercial schools, especially in cities and populated areas, that teach computing courses and
1822 computing certificate programs that compete with local colleges and universities. While all these experiences and
1823 external dynamics have value, they have had only a limited effect on university programs.
1824

1825
1826

1827 **3.5: Digest of Chapter 3**

1828
1829 This chapter discusses the concepts associated with knowledge-based learning which has been the traditional focus
1830 of computing education. The CC2005 report was primarily an articulation of the five computing discipline reports
1831 that existed at that time. It provided a comparative view and visual representation of those disciplines and reflected
1832 a model of education that primarily aided in establishing the course structure of respective curricula. The skills that
1833 students learned within those courses then relied on the design of the individual courses. As noted in recent job
1834 reports, however, the traditional model of computing education can lead to computing graduates not having the
1835 skills and attributes needed to pursue computing careers.

1836

1837 **Chapter 4: Competency-based Computing Education**

1838

1839

1840

1841

1842

1843

1844

1845

1846

1847

1848

1849

1850

1851

1852

1853

1854

1855

1856

1857 **4.1: Limitations of Knowledge-Based Curriculum Specifications**

1857

1858

1859

1860

1861

1862

1863

1864

1865

1866

1867

1868

1869

1870 **4.1.1: Articulating a Need**

1870

1871

1872

1873

1874

1875

1876

1877

1878

1879

1880

1881

1882

1883

1884

1885

1886

1887

1888

1889

1890 **4.1.2: The Skills Gap**

1890

- 1891 There is some evidence that there is a gap between the skills of today’s college graduates and the skills expected by
 1892 employers. This gap is frequently referred to as a *skills gap*. A recent survey by PSI Services found the following
 1893 situation in the United States [Psi1].
- 1894 ○ 80% of Americans (U.S.) agree there is a skills gap, and 35% say it affects them personally.
 - 1895 ○ 42.5% of recent graduates were underemployed as of March 2018, according to the Federal Reserve Bank of
 1896 New York.
 - 1897 ○ \$160 billion is the annual cost that researchers from the Centre for Economic Research calculated to be the
 1898 total cost of the skills gap to US companies.
 - 1899 ○ 60% of U.S. employers have job openings that stay vacant for 12 weeks or longer. The average cost from job
 1900 vacancies is at least \$800,000 annually.
 - 1901 ○ 81% of employers indicated that prospective employees lack critical thinking and analytical reasoning skills.
 1902 75% think graduates lack adequate innovation and diversity skills.

1903
 1904 Students who graduate from a university computing program might assume that the baccalaureate degree is a basic
 1905 qualification to attain a position, and that those who have baccalaureate computing degrees will be easily employable
 1906 in the computing field. The current high demand for computing professionals reinforces this idea. Yet the skills gap
 1907 that exists for college graduates in general is arguably equally true for computing graduates. And with the supply of
 1908 computing graduates significantly lagging the demand for computing graduates, this skills gap simply exacerbates
 1909 what is already a dire shortage.

1910
 1911 Industry has been attempting to articulate its needs for a number of years. Industry-sponsored certifications capture
 1912 both knowledge as well as what the individual is able to do. Skills frameworks, which are largely industry-driven and
 1913 produced, are similarly broad-based. Appendix B provides several summaries that address computing skills such as
 1914 SFIA, EU iCD frameworks. What needs to happen is that industry and academia need to begin to speak the same
 1915 language to the extent possible. The use of competencies by academic programs to specify curricula and curricular
 1916 standards is a way to move toward a common language for academia and industry.

1917 1918 1919 **4.1.3: Moving Toward Competency**

1920
 1921 The arrival of the IT2017 report [Acm07] heralded a shift away from the Ka-Ku-Lo mindset and redirected emphasis
 1922 toward performance. The IT2017 project was the first of the ACM/IEEE baccalaureate curriculum projects to embrace
 1923 the concept of competency as the primary characteristic of curriculum definition. The report stated that “competence
 1924 refers to the performance standards associated with a profession or membership to a licensing organization” and
 1925 that “assessing some level of performance is frequently used as a competence measure, which means measuring
 1926 aspects of the job at which a person is competent.” Independently of IT2017, the MSIS2016 report [Acm11]
 1927 introduced competencies at the master’s level, indicating that “competencies represent a dynamic combination of
 1928 cognitive and meta-cognitive skills, demonstration of knowledge and understanding, interpersonal, intellectual and
 1929 practical skills, and ethical values.” The *Software Engineering Competency Model* [Iee2] defined competency as
 1930 the “demonstrated ability to perform work activities at a stated competency level.” These three publications suggest
 1931 that competency is some combination of knowledge, technical skills, and human behavior within a computing context.

1932 1933 1934 **4.2: Competency and Competency-Based Learning**

1935
 1936 Competency is not a novel idea. The concept goes back centuries and millennia. The construction of the Giza Pyramids
 1937 or the Roman Colosseum are examples of structures designed and engineered by competent professionals of the time.
 1938 A general dictionary defines competence as “the quality or state of having sufficient knowledge, judgment, skill, or
 1939 strength” [Mer1]. It is important to note that the use of this word always occurs in a context: being competent in law
 1940 does not mean someone is competent in medicine.

1941 1942 1943 **4.2.1: Competency and its Meaning**

1944

1945 The Harvard University Competency Dictionary [Har2] describes a useful overview of competency through the
1946 following definition and explanation.

1947 *Competencies, in the most general terms, are “things” that an individual must demonstrate to be effective in a job, role,*
1948 *function, task, or duty. These “things” include job-relevant behavior (what a person says or does that results in good or poor*
1949 *performance), motivation (how a person feels about a job, organization, or geographic location), and technical*
1950 *knowledge/skills (what a person knows/demonstrates regarding facts, technologies, a profession, procedures, a job, an*
1951 *organization, etc.). Competencies are identified through the study of jobs and roles.*

1952 Thus, competency identifies closely with job-related behavior and performance. It is a person-centered concept that
1953 requires demonstration of human behavior together with technical skills and knowledge.

1954
1955 The CC2020 project has embraced competency as an underlying theme of its activities and as a principal component
1956 of this report. Its task force believes that every career path in computing, whether industrial, academic, governmental,
1957 or any other career, demands some level of competency. The project observes that knowledge is only one component
1958 of the idea of competency. While the working definition of computing competency may evolve, adopting the idea of
1959 competency as the foundational idea on which to base academic program design serves as an effective bridge between
1960 the deliverables produced by the academia and their consumption by society at large. Thus, it is logical that this report
1961 should foster competency-based learning instead of knowledge-based learning. This approach ensures that all
1962 graduates of computing programs have the preparation to be effective for every career path.

1963
1964

1965 **4.2.2: Previous Work on Computing Competency**

1966

1967 As noted earlier, three competency viewpoints have emerged in recent information technology and information
1968 systems curricula reports as well as for software engineering practice. Although these viewpoints evolved
1969 independently, their conclusions were very similar.

1970

1971 **Information Technology**

1972

1973 The information technology IT2017 report embraced competency-based learning, as opposed to the Ka-Ku-Lo model,
1974 mostly because almost all graduates from information technology degree programs enter industry and the workplace.
1975 The report adopted the term competence as related to performance in the workplace, i.e. what a graduate should bring
1976 to a job.

1977

1978 In education, success in career readiness requires that students in degree programs develop a range of qualities
1979 typically organized along three dimensions: knowledge, skills, and dispositions, so *competency* must connect these
1980 three elements or dimensions. The IT2017 report described this concept simply as:

1981

1982
$$\text{COMPETENCY} = \text{KNOWLEDGE} + \text{SKILLS} + \text{DISPOSITIONS} \dots \textit{in Context}$$

1983

1984 The interrelated dimensions had the following meanings. *Knowledge* designates an awareness and understanding of
1985 core concepts and content. This dimension receives initial attention from teachers when they design their syllabi, from
1986 departments when they develop program curriculum, and from accreditation organizations when they articulate
1987 accreditation criteria. This is the “know-what” dimension. *Skills* refer to capabilities and strategies that develop over
1988 time through deliberate practice and through interactions with others. Skills also require engagement in higher-order
1989 cognitive activities such as programming. This is the “know-how” dimension. *Dispositions* encompass socio-
1990 emotional skills, behaviors, and attitudes that characterize the inclination to carry out tasks and the sensitivity to know
1991 when and how to engage in those tasks [Per1]. This “know-why” dimension is the most challenging for academics
1992 because computing teachers routinely ignore disposition in educational settings.

1993

1994 There has been general agreement in education that success in career readiness requires that students in degree
1995 programs develop a range of qualities typically organized along these three dimensions. The IT2017 report also
1996 addressed approaches to learning. It rejected the content-driven mode of framing curricular guidelines using a
1997 disciplinary body of knowledge decomposes into areas, units, and topics to track recent developments in rapidly
1998 changing computing field. Instead, it proposed the use the “Understanding by Design” approach to transform content-
1999 based curricular models into a competency-based curricular framework. Here, learning transfer is multi-faceted with
2000 the transfer blended with skills and dispositions. Dispositions relate to metacognitive awareness, e.g. being

2001 responsible, adaptable, flexible, self-directed, and self-motivated, and having self-confidence, integrity, and self-
2002 control. They also include how to work with others to achieve common goal or solution.

2003

2004 **Information Systems**

2005

2006 Instead of specifying a body of knowledge or a set of courses as developed in the previous MSIS2006 report, the
2007 MSIS2016 curricular model identified a set of graduate competencies. Here, the term “competency” referred to
2008 graduate level ability to use knowledge, skills, and attitudes to perform specified tasks successfully. The report used
2009 a more formal definition for competency, as mentioned in the previous section, as follows:

2010 *Competencies represent a dynamic combination of cognitive and metacognitive skills, demonstration of knowledge and*
2011 *understanding, interpersonal, intellectual and practical skills, and ethical values [Loc1 p21].*

2012 In this context, competency is an integrative concept that brings together graduate level knowledge, skills, and
2013 attitudes.

2014

2015 The report also specified four different levels of category attainment: awareness, novice, supporting (role), and
2016 independent (contributor). The awareness level implies that a graduate student knows that the competency category
2017 exists and is aware of the reasons it is important for the domain of practice. The novice level specifies that a graduate
2018 can effectively communicate regarding matters related to the competency, perform component activities under
2019 supervision, and develop on-the-job experience related to the competency. The supporting (role) level indicates that a
2020 graduate has achieved a level of knowledge and skill that allows him/her to collaborate effectively in a supporting role
2021 with colleagues who have achieved a higher level of the competency to produce the desired outcomes. Finally, the
2022 independent (contributor) level refers to a graduate who has achieved a level of knowledge and skills that allows the
2023 graduate to perform without continuous support/supervision the tasks required to produce the desired outcomes.
2024 Higher levels of competencies do exist as an expert level.

2025

2026 The MSIS2016 curricular model suggested that all programs should not expect to prepare students to attain
2027 competencies at the same level in all competency categories. Different professional profiles have different needs and
2028 the professional profiles that a program desires its graduates to achieve can vary. That is, programs should determine
2029 the level at which its graduates should attain each of the competency categories.

2030

2031 **Software Engineering Competency Model**

2032

2033 The software engineering competency model (SWECOM) [Iee2] described capabilities for software engineers who
2034 participate in the development of and modifications to software-intensive systems. The model specifies skill areas,
2035 skills within skill areas, and work activities for each skill. Activities occur at five levels of increasing proficiency.

2036

2037 The SWECOM suggests that competency is a combination of knowledge, skill, and ability. A competent person has
2038 the knowledge and ability to perform work activities (i.e., skills) at a given competency level. The competency model
2039 includes cognitive attributes, behavioral attitudes, and technical skills. Some cognitive skills include reasoning,
2040 analytics, problem-solving, and innovation skills. Behavioral attributes include aptitude, enthusiasm, trustworthiness,
2041 cultural sensitivity, as well as communication, teamwork, and leadership skills. The model also specifies lifecycle
2042 skill areas, cross-cutting skills (e.g., quality, safety, security), and related activities. It also defined competency levels
2043 to be that of a technician (able to follow instructions), an entry-level practitioner (can assist in performing activities
2044 with some supervision), a practitioner (able to perform activities with little or no supervision), a technical leader
2045 (capable to lead and direct participants), and a senior software engineer (capable to create new processes and modify
2046 existing processes).

2047

2048 The SWECOM is very similar to the IT2017 and the MSIS2016 philosophies of competency. Knowledge and technical
2049 (computing) skills are integrated with behavioral attributes that correspond to either disposition or ability. Competency
2050 is central to the model and provides a modern view to generate excellence in computing education.

2051

2052

2053 **4.2.3: Initial and Developing CC2020 Explorations of Competencies**

2054

2055 The CC2020 task force initially explored the creation of competency statements by organizing subgroups of experts
2056 from different computing disciplines. Their initial work produced competency statements compatible with the

2057 canonical definition that *Competency = Knowledge + Skills + Dispositions, in context*. Appendix C presents a set of
 2058 preliminary computing competencies for computer engineering, computer science, information systems, information
 2059 technology, and software engineering competencies generated from this early work of the CC2020 task force. These
 2060 initial and developing explorations have inspired a more detailed expression of competency, presented in the next
 2061 section.

2062
2063

2064 **4.3: A Competency Model to Support Automated Visualization and Analysis**

2065

2066 The CC2020 project has developed a definition of competency and a template for specifying the subject matter of
 2067 baccalaureate computing education. This definition evolved from those developed and applied in the different
 2068 educational frameworks reported in the IT2017 report, the MSIS2016 report, and the SWECOM. The CC2020
 2069 representation supports a consistent, scalable model for writing curricular specifications.

2070
2071

2072 **4.3.1: The CC2020 Competency Model**

2073

2074 CC2020 articulates a notion of competency as a practical educational goal [Wag5, Fre5, Tak5, Top5] that refines the
 2075 Knowledge-Skill-Disposition (K-S-D) framework popularized in the IT2017 curriculum report. While the knowledge
 2076 dimensions of computing have been extensively explored in the various computing curricula, what is meant by skill
 2077 and disposition have had significantly less focus. Extending previous work, we specify competency as composed of
 2078 K-S-D dimensions observed within the performance of a task, T.

2079

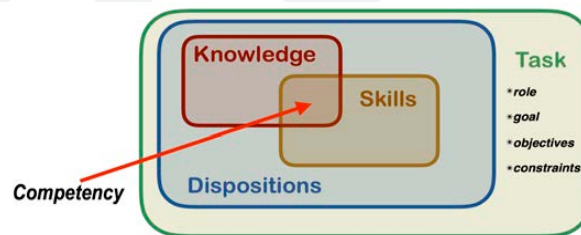
$$2080 \text{ COMPETENCY} = [\text{KNOWLEDGE} + \text{SKILLS} + \text{DISPOSITIONS}] \text{ IN TASK}$$

2081

2082 A competency specification enumerates knowledge, skills, and dispositions that are observable in the accomplishment
 2083 of a task, a task that prescribes purpose within a work context [Wag5]. Figure 4.1 illustrates the conceptual structure
 2084 of competency.

2085

2086



2087
2088 Figure 4.1. Conceptual Structure of the CC2020 Competency Model

2089

2090

2091

2092

2093 **4.3.2: Component Definitions**

2094

2095 The four components (knowledge, skills, dispositions, and task) that structure the competency specification have the
 2096 following meanings.

2097

2098 **Knowledge**

2099

2100 *Knowledge* is the “know-what” dimension of competency as a factual understanding. An element of knowledge
 2101 designates a core concept essential to a competency. This dimension reflects the enumerated subject matter that
 2102 teachers catalog as topics in their syllabi, departments distribute and balance among the courses they develop in an
 2103 academic program, accreditation organizations stipulate in their accreditation criteria, and employers identify in job
 2104 descriptions of their workers. Traditionally, curriculum guidelines for computing education have been predominated
 2105 by the designation of knowledge elements composed of facts based upon scientific derivation or proof.

2106
2107
2108

Skills

2109 *Skills* refer to the capability and strategy for applying knowledge to actively accomplish a task. Hence, skills express
2110 the application of knowledge and is the “know-how” dimension of competency. Skills often develop over time and
2111 with practice. Consequently, skill development often requires engagement in a progressive hierarchy of higher-order
2112 cognitive process. CC2020’s definition of competency has adopted Bloom’s levels of cognitive process [Acm22] to
2113 specify the degree of skill expected in successful task accomplishment.

2114
2115 One often assesses the skills dimension of competency indirectly through observation of the process or quality of work
2116 produced. The activation of “know-what” animated by “know-how” fuses the knowledge and skills dimensions. For
2117 that reason, the usefulness of any element of knowledge in a competency specification is only understandable when
2118 applied at a level of skillfulness; that is, specified or observed as a level of Bloom’s cognitive process. Therefore, an
2119 element of knowledge and the applied level of skill necessarily and naturally pair in the specification of a competency.
2120 In this way, the CC2020 competency model realizes a performance-based philosophy that animates an element of
2121 knowledge in achieving a task.

2122
2123 **Dispositions**

2124
2125 *Dispositions* frame the “know-why” dimension of competency and prescribe a requisite character or quality in task
2126 performance. Dispositions shape the discernment of skillful engagement of “know-what” and “know-how.” Specific
2127 to the task at hand, dispositions exert a moderating or controlling influence on a practitioner’s choices by proposing
2128 or projecting a desirable quality onto the outcome. How dispositions moderate knowledge and skill could be thought
2129 of as the extent that it accounts for the relation between the predictor and the criterion [Bar1] in that dispositions
2130 connect the ‘better’ or ‘correct’ application of knowledge and skill to the context in which they are applied. For
2131 example, dispositions moderate a practitioner’s capabilities to discern a task as “*professionally accomplished*” rather
2132 than simply “*completed*.” In this sense, dispositions are able to reflect the professional values behind a competency.

2133
2134 Dispositions characterize socio-emotional tendencies, predilections and attitudes that characterize the inclination to
2135 carry out tasks and the sensitivity to know when and how to engage in those tasks [Per1]. Hence, dispositions denote
2136 the values and motivation that guide applying knowledge while designating the quality of knowing commensurate
2137 with a standard of professional performance. “Know-why” exhibits as enacted values and because of the difficulty of
2138 assessing values and intention, people typically assess disposition indirectly through the observance of patterns of
2139 behavior or reflective practice.

2140
2141 **Task**

2142
2143 *Task* is the construct that frames the skilled application of knowledge and makes dispositions concrete. Task expressed
2144 as a colloquial prose statement provides the setting to manifest dispositions, where individuals moderate their choices,
2145 actions, and effort necessary to pursue and succeed in an efficient and effective manner. In this sense, task enfolds the
2146 purposeful context of competency, exposing the integral nature of knowledge, skills, and dispositions. To this end, a
2147 task definition stipulates pragmatic engagement that reflects professional practice relevant to the particular vision for
2148 the program graduates. For this reason, task descriptions provide an explicit context for the program to develop
2149 pedagogy that enables graduates to demonstrate competency as a computing professional.

2150
2151
2152 **4.3.3: Competency Statements**

2153
2154 An effective specification of competency is a synthesis of 1) a colloquial, prose competency statement that sets out a
2155 task, and 2) the component structure of constituent K, S, and D elements necessary to succeed in that task. As a whole,
2156 a competency specification expresses a model of knowledge that is skillfully and professionally applied in some task
2157 execution.

2158
2159 The competency statement corresponding to a competency specification is a free-form colloquial expression that
2160 succinctly conveys the pertinent ability goals attained through a course of study or the capabilities relevant to
2161 successfully performing a task in the workplace. The competency statement expresses the competency in terms that

2162 are familiar and comprehensible to a wide audience, typically using a vocabulary familiar to, and that resonates with,
2163 the stakeholder audience. The competency statement is then structurally augmented and amplified in the enumeration
2164 of knowledge, skills, and dispositions that complete the competency specification.
2165

2166 While the natural language of the competency statement favors a public audience, the competency component
2167 structure is more formal as it enumerates the components, e.g., knowledge elements demonstrated at a skill level and
2168 moderating dispositions determined necessary to demonstrate the competency in task. This structural enumeration of
2169 components is essential for automating comparative analyses and visualization of curricula. Having both the free-form
2170 of the competency statement alongside the more formal component-specific enumeration corroborates that the two
2171 perspectives align. Any divergence perceived in these perspectives would suggest the need for a closer reflection upon
2172 the usefulness of one or both representations.
2173

2174
2175 **4.3.4: Component Elements**
2176

2177 It is possible to identify specific elements for each component of the knowledge, skills, and dispositions dimensions
2178 of competency. Tables 4.1, 4.2, and 4.3 present suggested elements of these components. Appendix D recounts these
2179 tables in greater detail.
2180

2181 Table 4.1 illustrates thirty-six abbreviated knowledge areas partitioned into six categories. While the table is
2182 incomplete, it does provide an example high-level vocabulary for computing knowledge rooted in the collective
2183 wisdom of different computing communities. This summary of computing knowledge areas represents a well-
2184 understood and consistent vocabulary from which computing competency statements can evolve.
2185
2186

Table 4.1. Elements of Knowledge

Humans and Organizations	Systems Modeling	Software Systems Architecture	Software Development	Software Fundamentals	Hardware
Social Issues User Experience Security Policy IS Management Enterprise Architecture Project Management	Security Issues Systems Analysis Requirements Analysis Data Management	Virtual Systems Embedded Systems Integrated Systems Intelligent Systems Internet of Things Computer Networks Platform Technologies Parallel Computing Security Technology	Software Quality Software Verification Software Process Software Design Software Modeling Platform Development Software Development	Graphics and Visualization Operating Systems Algorithms Programming Languages Software Development Systems Fundamentals	Architecture and Organization Digital Design Circuits/Electronics Signal Processing

2187
2188
2189 Table 4.2 summarizes six cognitive skills together with abbreviated definitions. This illustration correlates with
2190 Bloom’s taxonomy and permits the adoption of a commonly agreed vocabulary as described in the 2001 revisions to
2191 Bloom’s taxonomy of educational objectives [And5]. The table lists the cognitive skills as verbs. In operational terms,
2192 coupling knowledge with skills normally occurs when accomplishing a task.
2193
2194

Table 4.2. Elements of (Cognitive) Skills

Remembering	Understanding	Applying	Analyzing	Evaluating	Creating
Exhibit memory of previously learned materials by recalling facts, terms, basic concepts, and answers	Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions.	Solve problems to new situations by applying acquired knowledge, facts, techniques, and rules in a different way	Examine and break information into parts by identifying motives or causes; make inferences and find evidence to support solutions.	Present and defend opinions by making judgments about information, validity of ideas, or quality of material.	Compile information together in a different way by combining elements in a new pattern or proposing alternative solutions.

2195
2196
2197 To round out the knowledge, skills, and dispositions dimensions of competency, Table 4.3 displays eleven prospective
2198 dispositions derived from the literature. Disposition, as an intrinsic component of competency, represents the

2199 opportunity to express institutional and programmatic values expected in in the workplace. Dispositional expectations
 2200 enrich the description/assessment of competency and/or the related pedagogy. Ascribing a disposition to a competency
 2201 indicates a clear commitment to self-reflection and examination that distinctly distinguishes a competency from a
 2202 learning outcome.

2203
2204

Table 4.3. Elements of Dispositions

Element	Elaboration	Element	Elaboration
Proactive:	With initiative, self-starter, independent	Adaptable:	Flexible; agile, adjust in response to change
Self-directed:	Self-motivated, determination, independent	Collaborative:	Team player, willing to work with others
Passionate:	Conviction, strong commitment, compelling	Responsive:	Respectful; react quickly and positively
Purpose-driven:	Goal driven, achieve goals, business acumen	Meticulous:	Attentive to detail; thoroughness, accurate
Professional:	Professionalism, discretion, ethical, astute	Inventive:	Exploratory. Look beyond simple solutions
Responsible:	Use judgment, discretion, act appropriately		

2205
2206
2207
2208
2209
2210
2211
2212
2213
2214

The set of dispositions is an essential characteristic of a well-structured competency, and it has an intricate involvement in statements related to workplace or academic activities. People inherently know these elements of human behavior. While it may be difficult to teach disposition, faculty members should instill these concepts in their students through assessment design, exercises, sustained practice, readings, case studies, and examples. The workplace and society assume that dispositions are an expectation from every competent computing graduate.

4.4: From Competencies to Curricula

2215
2216
2217
2218
2219
2220
2221
2222
2223
2224

A coherent competency model permits the definition of a computing curricula (i.e., structured collections of learning experiences) in a manner that benefits its constituencies: students, benefactors, faculty, administrators, employers, accreditors, lawmakers, and society. It is of interest to examine how key stakeholders can identify and author competencies as well as develop curricula based on the outcome expectations associated with competencies. This section summarizes the more comprehensive discussion in Appendix E and reviews topics that are essential for enabling the practical definition and use of competencies.

4.4.1: Identifying and authoring competencies

2225
2226
2227
2228
2229
2230
2231
2232
2233

Differing stakeholder groups may desire to identify and author collections of competencies. Computing educators at a university may wish to identify a collection of competencies to define the expected outcomes of the university's baccalaureate program(s) in computing. Computing educators who represent professional or academic societies might desire to specify competencies as a means towards establishing the outcome expectations of a national or global level model curriculum. Industry representatives might use a collection of competencies to specify their expectations for degree program graduates either for a specific job or for general use.

2234
2235
2236
2237
2238
2239

As described earlier, stakeholders can specify competencies using either narrative competency statements or a component specification that separately identifies the knowledge-skill pairs and dispositions. For most purposes, the process of identifying and authoring competencies involves elicitation of required narrative competency statements in collaboration with those who best know the expectations that program graduates will face both soon after graduation and throughout their careers. Educators writing competency statements for example, might collaborate with employers, students, or educational authorities and/or bodies.

2240
2241
2242
2243
2244

The methods and techniques for discovering competencies for a program specification are quite similar to those of systems requirements elicitation, including interviews, surveys, evaluation of existing requirements, as well as an analysis of industry recommendations (e.g., SFIA, e-CF) and academic model curricula (e.g., ACM/AIS/IEEE-CS recommendations).

2245
2246
2247

In order to define the highest level or most abstract competencies of a program, a course, or other curricular unit, it is necessary to articulate the knowledge, skill, and disposition components associated within a context. In a free-form

2248 competency statement, the focus is typically on the general outcome of the competency in the expected context.
2249 Expressing a competency in this manner, the knowledge, skill, and disposition components might not contain full or
2250 detailed exposure. Instead, users may need to infer the details from the free-form statement. Therefore, articulating
2251 the context of a competency is always crucial since it provides the motivation for the stakeholder, making it meaningful
2252 to learn and perform that competency.

2253
2254 The literature offers some insight for developing competency statements [Per5]. In this setting, writers of competency
2255 statements should:

- 2256 1. Stipulate them as learner-oriented, essential competencies.
- 2257 2. Specify them in “clear, specific, unadorned, and concise language,” that are measurable.
- 2258 3. Structure them as action oriented and begin with “the verb that most precisely describes the actual, preferred
2259 outcome behavior to be achieved.”
- 2260 4. Construct them to be consistent with “standards, practice, and real-world expectations for performance,” thus
2261 reflecting what “the practitioner actually needs to be able to do.”
- 2262 5. Formulate them to contribute to a “cluster of abilities needed by the graduate to fulfill the expected overall
2263 performance outcomes.”

2264
2265 Component specifications fully aligned with competency statements are essential for comparison and analysis
2266 purposes. In addition, the process of translating a free-form competency statement into a component specification may
2267 reveal non-desirable characteristics of the statement and can offer opportunities for significant improvement. The
2268 process of deriving component specifications from free-form statements is an iterative one and requires willingness
2269 and ability to interpret the statements in a way that allows identification of components inferred from the narrative
2270 statement.

2271
2272 In some cases, it could be useful to start from the competency components. Identifying the knowledge, skills, and
2273 dispositions components of a competency before constructing a competency statement could also be a good starting
2274 point. This is especially true in cases when the identity of a target competency is not fully clear and first requires
2275 calibration at a component level.

2276
2277

2278 **4.4.2: Competency Specifications and Curricular Specifications**

2279
2280 Competencies alone do not address a question of how the educational experiences needed to enable students to acquire
2281 expected competencies by the time of graduation can be determined. Outcome expectations specified as competencies
2282 require transformation into a curriculum form consisting of educational activities that help to scaffold students’
2283 progression in various types of outcome areas.

2284
2285 Past experiences associated with processes that derived educational experiences from competencies can be helpful
2286 (e.g. the MSIS 2016 report, the International Academy of Astronautics (IAA) Space Industry Systems Engineering
2287 competency model, and the business curricula for competency specifications [Chy5].) Guidance from these
2288 experiences includes the following:

- 2289 ○ Determine the characteristics of learning experiences that constitute a curriculum based on outcome
2290 expectations specified with competencies.
- 2291 ○ Indicate program competencies as a foundation for curricular specification benefits from existing competency
2292 models (e.g., those developed by industry, government, or professional societies).
- 2293 ○ Develop educational experiences that require not only identification of competencies, but also specification
2294 of the expected attainment levels from novice to expert.
- 2295 ○ Derive an initial set of learning outcomes associated with each competency; then, organize the learning
2296 outcomes into learning experiences. The sets of learning outcomes within each learning experience determine
2297 the topics in which students should engage and the pedagogical forms expected from the engagement.
- 2298 ○ Assess continuously the extent to which the implemented learning experiences enable students to attain the
2299 expected competencies at the expected level.

2300
2301 The opportunity for students to develop skills and dispositions is a positive but potentially resource-intensive effect
2302 of specifying program outcomes with competencies. In many cases, such an approach requires a different set of
2303 pedagogical assumptions and methods compared to a mostly knowledge-based specification of assessable outcomes.

2304 In practice, competency-based outcome specification can lead to a broader set of types of learning experiences. These
 2305 often include a much stronger focus on various forms of experiential learning, from interactive simulations, to
 2306 intensive projects, to field experiences, and to internships and cooperative programs with industry. Domain-specific
 2307 skills and dispositions require a learning environment that is different from a traditional classroom environment.
 2308
 2309
 2310

2311 **4.5: Practical Applications of Competency**

2312
 2313 People can assemble competencies in collections such as curricula, curriculum standards, specific job requirements,
 2314 or requirements for categories of jobs. All such specifications are similar in nature and structure. Hence, it is possible
 2315 to define the general notion of a “competency target” that reflects an entity defined by a collection of competencies.
 2316
 2317

2318 **4.5.1: Competency Targets**

2319
 2320 Table 4.4 identifies four competency targets. The Singular-Aggregate dimension of the table reflects whether the target
 2321 is for a single entity or for a category of entities. The Education-Workforce dimension reflects whether the target
 2322 relates to an education product or to a workforce product.
 2323
 2324

Table 4.4. Competency Targets

	Education	Workforce
Singular	Programs	Jobs
Aggregate	Subdiscipline	Careers

2325
 2326 Each of these four targets represent an important application of competencies. *Programs* are individual computing
 2327 educational programs delivered by colleges and universities. *Subdiscipline* represents curricular standards for each
 2328 computing subdiscipline developed by the professional societies such as future curricular reports for computer
 2329 engineering or computer science. *Jobs* reflects a specific work opportunity where industry or government can specify
 2330 the requirements in terms of a set of competencies. *Careers* is a category composed of similar jobs where industry or
 2331 government can specify the requirements across a category in terms of a set of competencies.
 2332
 2333

2334 **4.5.2: Target Proximity**

2335
 2336 Since a set of competencies can characterize a target, it is possible to view the structured K-S-D portion of each
 2337 competency’s specification as a point in 3-D space, and a set of competencies as a point cloud. This approach lends
 2338 itself to a visualization of competencies that require further exploration as presented in Chapter 5 and Appendix G.
 2339

2340 While visualization of competencies in this model may provide insights, the idea of considering the proximity of
 2341 targets is also a promising concept. Developing a specific distance metric between two targets is a potential area for
 2342 future research and is based on the idea of obtaining an ordered value for this metric. Such a distance metric could
 2343 enable pairs of targets ranked in terms of “closeness” similarity. For example, suppose students are searching for an
 2344 educational program (i.e., *Programs*) to prepare themselves to be network administrators (i.e. *Careers*). If competency
 2345 specifications exist for both the potential programs and the desired career, then the distance metric can rank programs
 2346 in terms of how close they are to the desired career. The educational program that is the closest distance to the network
 2347 administrator career target could be the optimal degree program for the students.
 2348

2349 It is possible to use a distance metric to support comparisons among all four types of targets. The following scenarios
 2350 provide opportunities for target proximity.

- 2351 ○ For education providers, there is the opportunity to reduce the distance between the competencies associated
 2352 with a program and the targeted jobs and careers by that program. The articulation of that distance could
 2353 allow providers to make changes that close the gap and positively establish a reduction in that distance.

- 2354 ○ For education providers, there is the opportunity to calibrate programs with national and international
- 2355 standards for curricula in various fields by evaluating the distance between the provider’s program and the
- 2356 curricular standard for the subdiscipline.
- 2357 ○ For pre-college students, there is the opportunity to select a program based on the program whose
- 2358 competencies are the shortest distance from the desired job or career.
- 2359 ○ For hiring organizations, there is the opportunity to quantify the distance between the competencies required
- 2360 for a position, and the competencies exhibited by various candidates based on their completion of various
- 2361 educational programs and processes.
- 2362 ○ For college graduates, there is the opportunity to search for jobs based on the distance between personally
- 2363 held competencies and the competencies required for targeted careers.
- 2364

2365 Based on pairs of targets, Table 4.5 conceptualizes several questions that could be addressable by this framework.

2366 Table 4.5. Framework Questions

Target #1	Target #2	Practical Exemplar Question
Program	Career	How well does ABC University’s information technology program prepare someone to be a network administrator?
Subdiscipline	Career	How well does a computer engineering degree from XYZ University prepare someone to be a chief information security officer?
Program	Job	How well does ABC University’s computer science program prepare someone to be a senior programmer at ACME Corporation?
Program	Career	How well does ABC University’s information systems program prepare a current business student to develop a career in programming?
Program	Program	What are the differences between ABC University’s computer engineering program and XYZ University’s software engineering program?
Program	Subdiscipline	How closely aligned is ABC University’s current computer science program to the (hypothetical) competency-based curriculum espoused the pending CS202x report?

2368

2369 4.5.3: Consequences of Target Proximity

2370 The target proximity approach attempts to unify the education and workforce sides of computing. The past approach

2371 was to define computing curriculum standards in terms of knowledge areas, knowledge units and learning outcomes.

2372 That approach complements academia and uses the classical role and scope of higher education as the curator of

2373 knowledge.

2374 On the workforce side, there have been successful attempts to define job requirements in terms of competencies

2375 through the development of competency frameworks. Recall the examples mentioned earlier and the Skills Framework

2376 for the Information Age (SFIA), the European Competency Framework (e-CF), and the i Competency Dictionary

2377 (iCD).

2378 As noted in the earlier part in this chapter, computing educators have been transitioning to competencies for several

2379 years for some of the recent model curricula. However, the process is a long way from utilizing a common language

2380 that transcends both education and workforce. The lack of a formal structure in many previous notions of competencies

2381 means that constituents have not had a way to quantify and analyze competencies in terms of the questions in Table

2382 4.5. This CC2020 report advocates a transition over time to a common language that stakeholders can utilize across

2383 education and workforce constituencies to understand and minimize the gap between education outputs (graduates)

2384 and the inputs required for a successful contribution to a global workforce in computing.

2385 4.6: Digest of Chapter 4

2386 This chapter deals with the nature of competency – a salient feature of the CC2020 project. It presents several

2387 competency statements to exemplify applications. Competency-based curricula are more expressive in their learning

2388 goals, and more easily translated to the language of graduate job descriptions and industry needs. Recognizing the

2397 knowledge-based approaches taken in many computing curricula to date, recent developments in computing curricula
2398 imply that the components of computing curricula should include not just knowledge and skills but also dispositions,
2399 skill levels, and typical (maybe “practical”) tasks expected of graduates. The use of the competency model can also
2400 assist in potentially automating the comparative analyses and visualization of curricula programs in computing. For
2401 these reasons, the CC2020 task force recommends that future curricular reports adopt this competency-based approach
2402 to describe computing curricula.

2403
2404
2405
2406

DRAFT

2407

2408 **Chapter 5: Visualization and Competency Specification**

2409

2410 This chapter describes the analysis and visualization of competency specifications. The goal is to be able to handle
2411 the digital representation of competency specification, and be able to analyze courses, programs, and curricula that
2412 have been specified with the competency paradigm described in Chapter 4. The competency specifications are stored
2413 in a repository which makes it possible for the analysis and visualization. Furthermore, the analysis and visualization
2414 should be made available to the public in the form of a set of tools that take information from a repository and/or from
2415 the user.

2416

2417

2418

2419 **5.1: On Visualization**

2420

2421 An important initiative in CC2020 is the ability to visualize different characteristics of this project such as curricula,
2422 competencies, and other useful features. To this end, the task force has developed a series of visual representations
2423 and tools for use by the stakeholders of this report. Before embarking on this journey, it is important to address some
2424 preliminaries.

2425

2426

2427 **5.1.1: Some Basic Functions**

2428

2429 The basic functions of this set of tools include the following:

2430

2431

2432

2433

2434

2435

2436

1. Content Management: User(s) may enter competency specification(s) in various formats.
2. Reporting/Presentation: User(s) may retrieve, display, format, and disseminate representations of competency specifications.
3. Competency Analysis: User(s) may query repository content specifying any category attributes or specification content and represent the query results as listings, comparisons, or visualizations for the purpose of analyses.

2437 The first function (content management) supports the collection and curation of glossaries of knowledge, skills, and
2438 dispositions along with synonyms and translations. Repository contents may be input manually or mechanically
2439 imported/exported using published formats and protocols (see Appendix F). This will accommodate not only
2440 competencies specific to formal curricular guideline development but also industry characterizations of professional
2441 and employment competency.

2442

2443 The second function (reporting/presentation) provides a facility for representing competency specifications formatted
2444 for copying into formal organizational or institutional documents such as academic programs, accreditation standards,
2445 or professional licensure reports.

2446

2447 The third function (competency analysis) is concerned with analyzing and visualizing competency specifications. This
2448 may be done individually or collectively in comparison with each other. “Low-level” analysis is based on individual
2449 competency specifications, and “high-level” analysis is based on collections of competency specifications. This will
2450 make the tool useful for various users, e.g., government, academia, employers, prospective/current students as well as
2451 their advisors and parents.

2452

2453

2454 **5.1.2: Stakeholders and Visualization**

2455

2456 The illustrations in this subsection present predictable queries as examples of interest to various user communities along
2457 with prospective visualizations supporting analysis of the query results. The user communities include the following
2458 five stakeholders as discussed in Chapter 1: prospective students, current students, industry professionals, educators,
2459 and educational organizations and authorities

2460

2461 Some typical questions that each of these users may have are as follows.

- 2462
2463 A prospective student may ask:
2464 • Which curriculum would best fit my interests?
2465 • Are there differences between universities that offer the same curriculum?
2466 A current student may ask:
2467 • Which courses in the information systems curriculum of your university have a greater emphasis (at a higher
2468 level or longer duration) compared to the information systems curricular guidelines?
2469 Industry professionals may ask:
2470 • Does this university’s curriculum provide competencies that will be necessary to work for my company?
2471 • Which curriculum might potentially provide professional education for the company’s employees?
2472 Educators may ask:
2473 • How does my program fit with an international curricular guideline?
2474 • Is there a particular area where my curriculum is different from the guidelines? Where I might consider
2475 changes?
2476 Educational authorities may ask:
2477 • Does this curriculum comply to the guidelines for curricular guideline X?
2478 • I want to rank universities/departments based on curriculum. Do universities that say they teach X actually
2479 teach Y?
2480
2481

2482 **5.2: Competency as a Basis for Visualization**

2483
2484 The procedure used in the following discussion assumes that data is available for use and analysis. From this basis, it
2485 is possible to visualize competencies for stakeholder use.
2486
2487

2488 **5.2.1: Data and Use**

2489 Data form the basis of the analysis and visualization. A competency specification (competency statement, knowledge
2491 element(s) paired with skill level, and disposition(s)) forms the most basic data. A repository that stores competency
2492 specifications will be central to the analysis and data. The elements of knowledge, skills, and dispositions could be
2493 those that have already been described in Tables 4.1, 4.2, and 4.3.
2494
2495

2496 **5.2.2: Visualization and Competency Specification**

2497 Using a table format, an example competency specification is given in Figure 5.1. Ref# and Title denote the reference
2498 number and title, respectively, of a competency specification. The other three columns show the competency
2499 statement, a list of dispositions, and a list knowledge-skill pairs – all together forming a single competency
2500 specification. Thus, specification CA1 has two disposition elements D-1 and D-2, and four knowledge-skill pairs,
2501 where knowledge K(X-1) has the associated skill B-3, knowledge K(X-2) has the skill B-4, and so forth.
2502
2503

Ref#	Title	Competency Statement	Dispositions	Knowledge-Skill Pairs	
CA1	<i>Title1</i>	<i>statement1</i>	D-1	K(X-1)	B-3
			D-2	K(X-2)	B-4
				K(X-3)	B-3
				K(X-4)	B-2

Figure 5.1. Example Competency Specification in a tabular format

2504
2505

2506 Note that the CC2020 competency model supports any individual or collection of competency specifications that may
 2507 represent curriculum content: job descriptions, plans of study, academic degrees, training certifications, professional
 2508 accreditations and licensure, and standards of performance evaluation. Thus, the range of questions that stakeholders
 2509 can pose is limited only by the breadth of the competency information that is stored in the repository.
 2510

2511 In the following section, examples of plausible visualizations of likely queries made by representative stakeholders
 2512 are presented.
 2513

2516 5.3: Visualization Examples of Stakeholder Queries

2517
 2518 The questions or queries typically made by each of the stakeholder communities can suggest various visualizations of
 2519 the query results. These representations are chosen to demonstrate the expressive potential of graphic communication.
 2520 There is no intent to describe the computation required for the data to underpin the graphic. They are, however, all
 2521 conceived as derived from the competency repository structure that implements the competency specification syntax.
 2522 See Appendix G for use cases as well as other visualizations that are considered in the CC2020 project.
 2523

2525 5.3.1: Students

2526
 2527 Suppose that a prospective student asks the question “Which curriculum would best fit my interests?” Assuming that
 2528 the student has a set of knowledge elements that she is interested in (e.g. “software fundamentals”, “systems
 2529 modeling”, and “users and organizations”), she can search to find a visualization such as the one shown in Figure 5.2
 2530 that illustrates how much each curriculum would incorporate those knowledge elements. In Figure 5.2, each
 2531 knowledge element is mapped for each of the curricular guidelines. The size of the blue squares indicates the extent
 2532 to which the knowledge element is relevant in the corresponding curriculum. The student can then visually compare
 2533 the relevancy of each of the three knowledge elements and make her decision.
 2534

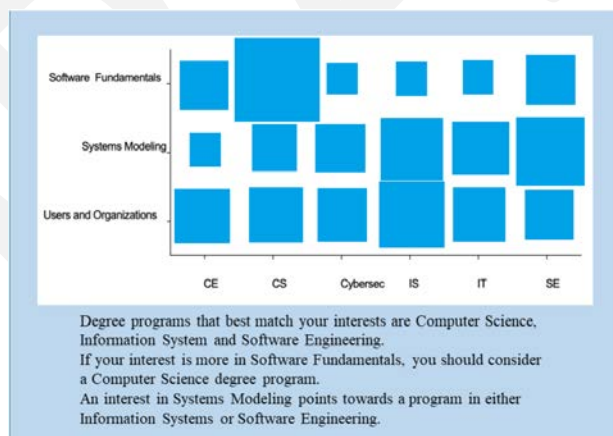


Figure 5.2. Mapping of knowledge elements to curricular guidelines

2535
 2536
 2537

2538 5.3.2: Industry

2539
 2540 A user from industry could ask the question, “Of CS and IT, which curriculum might potentially provide professional
 2541 education for the company’s employees?” A radar chart such as shown in Figure 5.3 would be helpful to compare the
 2542 two curricula based on six knowledge elements. The distance from the center indicates the skill level related to each
 2543 knowledge category. The radar chart has been augmented with the specification from the user (“User curriculum” in
 2544 this example.) In the visualization, it appears that IT is the best match for the user’s required knowledge levels. This
 2545 is because there is a complete coverage of the user’s specifications and the curriculum content; that is, the blue CS

2546 surface completely overlaps the user's green specification surface.
2547

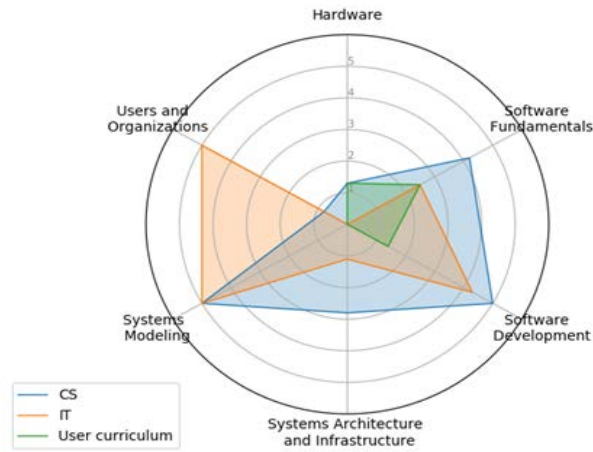


Figure 5.3. Comparison of CS and IT

2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561

5.3.3: Computing Educator

A computing educator has the question “How does my program fit with an international curricular guideline?” Figure 5.4 shows a comparison of a particular institution’s evaluation of its program as a solid line to the evaluation based on the knowledge areas listed in Table 1 of Appendix D. In this case, the evaluation denotes the weight of each knowledge element in the CS subdiscipline. The figure shows how this institution matches the guidelines and where the institution differs. For example, compared to the “standard” CS curriculum, this institution has a stronger emphasis on knowledge elements such as Enterprise Architecture, Embedded Systems, and on hardware related elements such as Circuits and Electronics.

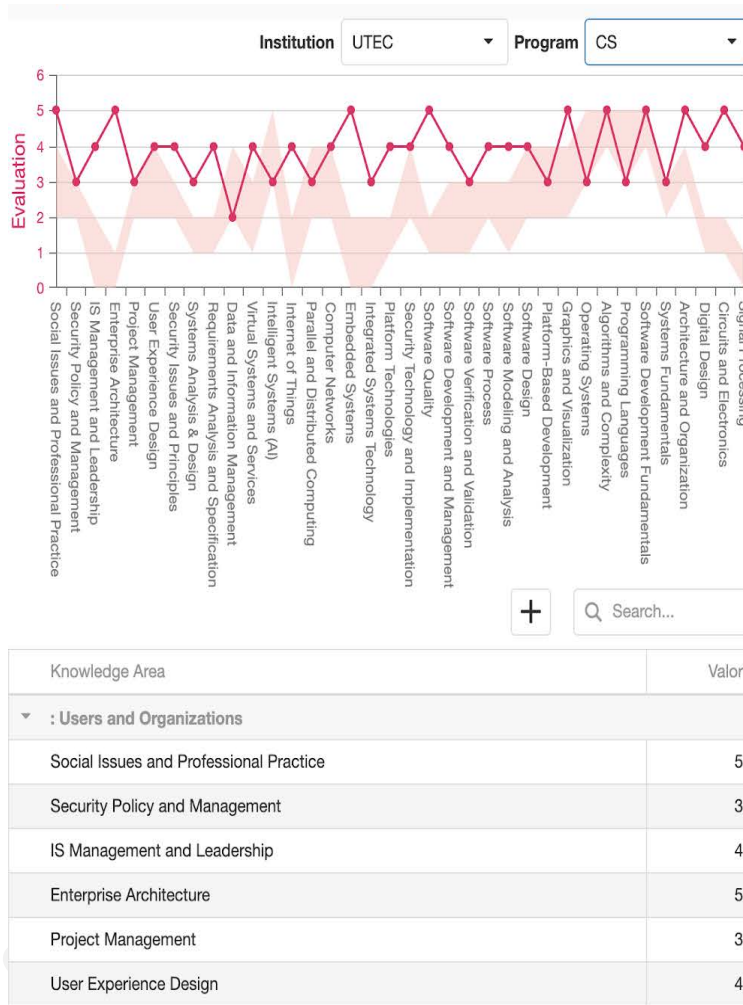


Figure 5.4 Comparison of an institution’s evaluation against the knowledge table evaluation

2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581

5.3.4: Educational Authority

Educational authorities could also use Figure 5.4 to answer the question “Does this curriculum comply with the guidelines for curriculum X?” Figure 5.4 shows that none of the institution’s evaluation falls below the minimum value of the “standard” CS curriculum. This suggests that this institution’s CS curriculum complies to the standard CS curriculum. Note that the same figure can be used by different stakeholders to address different questions.

5.4: Other Related Visualizations

This section gives examples of other related visualizations. These are not necessarily based on the competency paradigm.

5.4.1: Visualization of a full curriculum

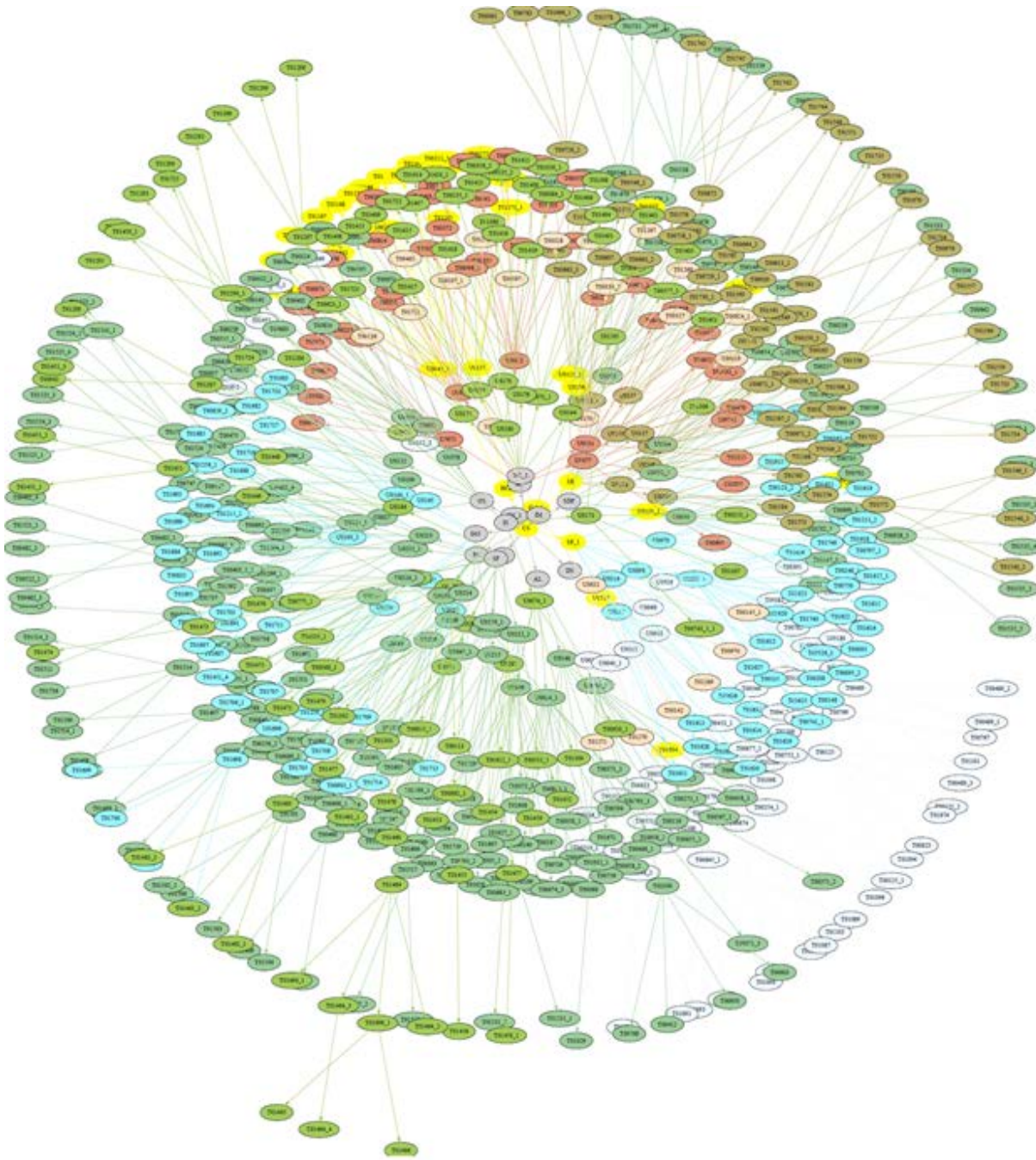


Figure 5.5. Graph-based structure of the core components of CS2013.

2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592

Figure 5.5 shows a graph-based structure of the core components of CS2013. It centers around the CS node that links the knowledge areas (KAs), their core knowledge units (KUs) and their respective topics [Mar2]. KA nodes are near the center and colored gray. KU nodes are placed just outside of the KA nodes having labels starting with U, and topics nodes are placed in the outer part having labels starting with T. Note that the color allocation (other than gray and yellow) in Figure 5.5 only serves to differentiate KU and topic nodes that belong to different KAs. The meaning of yellow will be described in Appendix G.4.

Figure 5.5 by itself will not have much meaning as it contains an enormous amount of data, and the reader will not be

2593 able to see its details. But with functions such as coloring, filtering, and zoom-in/out, it can become much more
 2594 powerful. Section 4 in Appendix G gives an example of obtaining a visualization of a subset of the CS2013
 2595 components related to Human Computer Interaction.
 2596
 2597

2598 **5.4.2: Visualization of the Knowledge table**
 2599
 2600
 2601

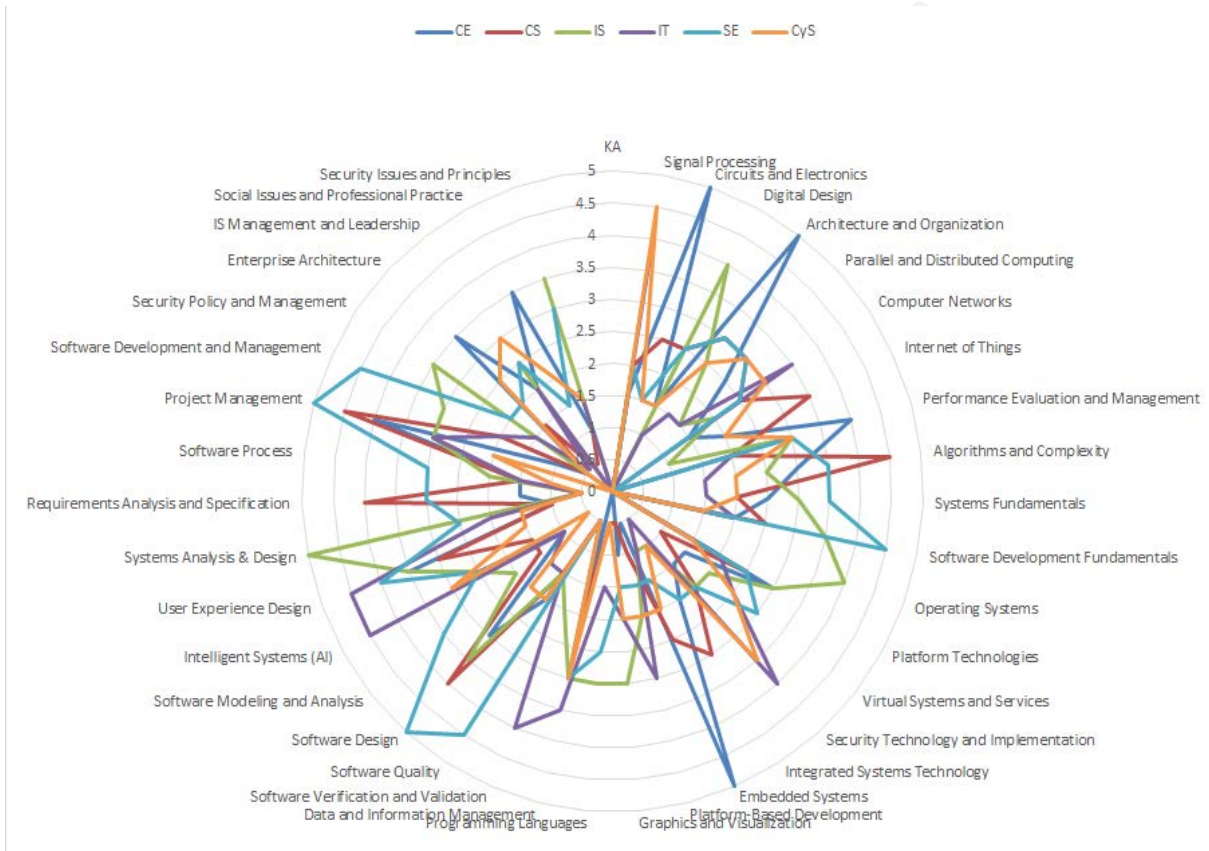


Figure 5.6. Radar Chart showing maximum emphasis of knowledge areas

2602
 2603 Many types of visualizations can be made for the same set of data. Figure 5.6 presents a radar chart that shows the
 2604 maximum emphasis for the knowledge areas. Section 3 in Appendix G also gives a bar chart and line chart for the
 2605 same data set, as well as a ribbon chart which could also be used for the same data set.
 2606

2607 Note that similar to Figure 5.5, Figure 5.6 includes a large amount of data, i.e., data for all curricular guidelines against
 2608 almost 40 knowledge areas. This figure may look “nice”, but one can easily understand that Figure 5.6 has too much
 2609 data for it to be practical. Although the spokes are different, Figure 5.3 is much easier to understand as it has less
 2610 amount of data.
 2611
 2612

2613 **5.4.3: Visualization of Competency**
 2614

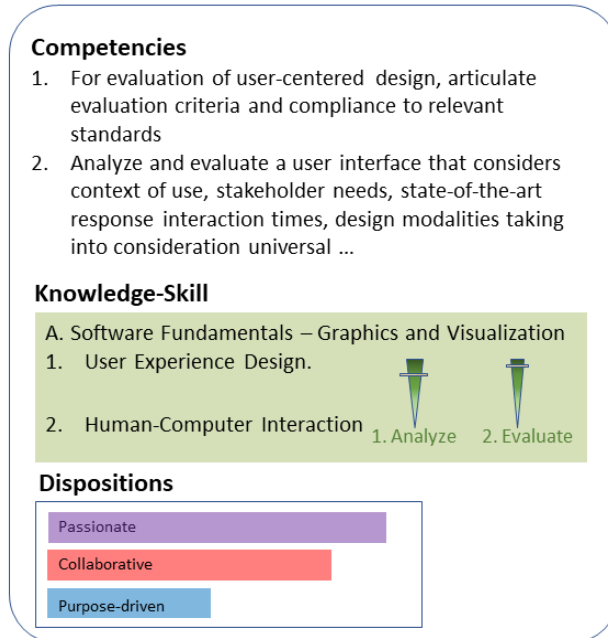


Figure 5.7. Sketch of a competency visualization

2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627

Figure 5.1 gave a table format of a competency, but other possibilities can also be considered. Figure 5.7 gives one such example. Here, two competencies are first described in conventional natural language. Additionally, the figure shows that the two competencies are concerned with two knowledge elements: User Experience Design and Human-Computer Interaction. To the right of these, the skill level of User Experience Design is Analyze and that of Human-Computer Interaction is Evaluate. Below these, three dispositions are shown that are concerned with the competencies, and the importance of each of these three are represented by the length of each of the bars. Thus, the Passionate disposition is more important than the Collaborative disposition, which in turn is more important than the Purpose-driven disposition.

5.5: Challenges Concerning Visualization

2628
2629
2630
2631
2632
2633

The ability to visualize aspects of curricula and competencies present several challenges. These challenges can include the interpretation of nomenclature and vocabulary or the comparison of two entities.

5.5.1: Consistent Vocabulary

2634
2635
2636
2637
2638
2639
2640
2641

One issue that exists is the issue of standardized vocabulary for knowledge. For example, “refactoring” appears in both CS2013 and SE2014. In CS2013, the word refactoring is part of “Software Design” and “Software Evolution” knowledge units. In SE2014, refactoring is part of “Software Process” knowledge unit but is not in the “Software Design” knowledge unit. This suggests that refactoring may have a possibly subtle different meaning between the two curricula, or that “Software Design” has a subtle different meaning.

2642
2643
2644
2645
2646
2647

In a perfect world, an ontology would be developed that all computing curricular guidelines would adhere to. This unfortunately may not be feasible as there is an issue of

- (1) Can such an ontology be developed?
- (2) Would all computing sub-disciplines actually adhere to it?
- (3) How does one handle emerging sub-disciplines?

These three issues are interesting topics that future work should consider.

2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696

5.5.2: Entity Comparison

There is also an issue of what it means to “compare” entities. A simple comparison would be to check for equality. But that may not always be appropriate. For example, as the skill level has an ordering aspect to it, if an element of knowledge K is required to be at the skill level of “understand”, then any competency whose knowledge K is at a higher skill level, e.g., “apply”, “analyze”, “evaluate”, “create”, should satisfy or be sufficient for comparison purposes (for just that knowledge K).

Another issue is the comparison of composite competency specifications. Comparing two atomic competency specifications may be fairly straightforward. We can just compare the respective knowledge-skill pairs and dispositions of the two atomic competency specifications. But there could be multiple ways to compare composite competency specifications. For example, one possibility is to “drill down” and compare the “leaf” competency specifications. Another possibility is to compare at the “top level” competency specification.

5.5.3: Visualization types

Visualization can take many forms for even the same data set. For example, Section 3 of Appendix G shows several visualizations of the same data. Different people may prefer different types of graphs to understand a comparison. In fact, although color is an important part of visualizations, the choice of color may actually hinder understanding for someone with color vision deficiency. For example, some people will confuse red and green, even if most people consider these colors to be obviously different.

Another important point is the amount of data that can be shown in one visualization. As mentioned in sections 5.4.1 and 5.4.2, too much information in one visualization can make it difficult to understand what the visualization wants to say, even though it may look “nice”.

Thus, there are two challenges here. One is to have a feedback mechanism to understand which visualizations are better than others. Another challenge is to have a mechanism that will allow users to change the visualization which could mean “switching” between different types of visualizations, changing the use of colors, or filtering the amount of information that is shown. Both are considered to be part of future work.

5.6: Digest of Chapter 5

This chapter describes the analysis and visualization of curriculum information, where the information takes the form of competency specifications. The basic functions, i.e., content management, reporting/presentation and competency analysis, that the analysis and visualization tool set will have are first described. The analysis and visualization target stakeholders are reviewed along with some typical questions they might ask. As the analysis and comparison are to be done on competency specifications, this chapter next describes it in a tabular form. Several visualization examples are also given. Each visualization helps answer a question that a stakeholder may have, and the same visualization can answer different questions from different stakeholders. Other visualization examples are given in Appendix G. Finally, the chapter addresses some challenges that need to be overcome for the tool set to evolve and be continually useful.

2697

2698 **Chapter 6: Global Considerations**

2699

2700

2701 This chapter addresses some of the global issues surrounding computing education. Some of these issues include the
2702 lack of a common nomenclature among different countries and regions, the different forms of degree programs across
2703 the world, and the various dynamics that can influence the ability of universities to produce competent computing
2704 graduates.

2705

2706

2707

2708 **6.1: Global Context and Computing Programs**

2709

2710 There is a need to express curricular models and recommendations in a neutral manner that is acceptable and
2711 understandable by all people. Terminology and nomenclature are one aspect of this. Another is how any
2712 recommendation reflects the cultural context on a global scale.

2713

2714

2715 **6.1.1: Computing ‘Tower of Babel’**

2716

2717 The Tower of Babel is a well-known biblical accounting where populations who were building a tower to reach heaven
2718 had their singular speech confounded so that they could no longer understand each other. In some way, this seems to
2719 be the situation regarding computing today. It is neither intended nor possible for this report to determine a single
2720 terminology applicable to computing education worldwide. This report, initially written in English, may not be the
2721 *lingua franca* (common language; literally, French language) of some countries in the world. Although this report has
2722 been written using English terminology, it does not presume to tell speakers of other languages which terminology
2723 they should use in those languages.

2724

2725 For example, in most countries of Europe, a word that is associated with computing degrees translates directly into
2726 English as *informatics*. In 1957 Karl Steinbuck coined the German word *Informatik* [Ste1] and subsequently, other
2727 languages adopted the word. Examples are *l’informatique* in French, *informatica* in Italian and other languages,
2728 *informática* in Spanish, and *informatikës* in Albanian. In translating these words into English, they are sometimes
2729 translated as ‘computer science’ in deference to the US terminology. However, there is an increased tendency to use
2730 the word as ‘computing’ as an equivalence to ‘informatics’ lately. The informatics family of names developed
2731 independently of, and approximately at the same time as, computer science in the United States. The term ‘computer
2732 sciences’ first appeared in print in 1959, and it was another three years before the appearance of the first study program
2733 called ‘computer science’ [Ted1]. When not used as the name of a degree, universities sometimes use informatics as
2734 the overarching name for the academic discipline such as the School of Informatics at the University of Edinburgh or
2735 the Department of Informatics at Sussex University, as well as within countries.

2736

2737 In Latin America, a preference exists to include the word ‘engineering’ in degree titles. A working group organized
2738 to study the ‘computing engineering’ degrees in Latin America and elsewhere concluded that it would be unreasonable
2739 to expect everyone to use a common set of program names. Instead, it developed a common set of categories to
2740 describe the content of degrees. For example, applying those categories to the systems engineering degree in Uruguay,
2741 it found that the degree had good coverage of concepts from computer science and software engineering, lower
2742 coverage of concepts from information systems and information technology, and, despite its name, weak coverage of
2743 concepts from computer engineering [Ram1]. This example should suffice to make it clear that terminology is not
2744 uniform even when translated into English.

2745

2746 Australasia (which incorporates Australia and New Zealand) provides another example, this time with ‘information
2747 technology’ as its terminology. Australasia uses information technology to refer to the whole of the academic field of
2748 computing. For example, at the time of writing, the entry to the information technology office area at Monash
2749 University in Australia bears the following message.

2750 “Information Technology: algorithm, distributed systems, database, software engineering, network, information systems,
2751 computation, knowledge management, analysis, mobile computing, design, e-business, model, data mining, interface,
2752 business decision support”
2753

2754 A comparison of these terms with those found in a typical northern American information technology degree should
2755 make it clear that the same term means very different things in these two regions.
2756

2757 2758 **6.1.2: Computing Nomenclature** 2759

2760 This report takes a proactive position in attempting to normalize the use of terminology in the computing field. The
2761 public cannot view computing as a valid profession if professionals within it cannot agree on the meaning of the words
2762 it uses. Established sciences and professions such as medicine have definitive meanings regarding the words they use.
2763 This simple understanding does not exist in computing. The computing field is relatively new in comparison, and
2764 many activities have emerged independently. This misunderstanding of terminology will only bring more confusion
2765 to the field. Fortunately, this CC2020 project through Appendix H and other literature on global interpretations of
2766 computing education terminology [Sim1] have helped to dispel such misgivings.
2767

2768 This report necessarily uses specific terminology. The companion reports for identified subfields of computing
2769 necessarily use specific terminology. Unfortunately, there is no universal terminology in computing or in computing
2770 education, even within the English-speaking world. To some individuals, in a specific context, terms such as computer
2771 engineering, computer science, information systems, information technology, software engineering, and informatics
2772 have reasonably clear meanings. However, to other individuals in other contexts, they can have quite different
2773 meanings - and those different meanings have just as much legitimacy.
2774

2775 People need to be conscious of all this variation when writing about degrees, especially within a strictly local audience
2776 or when reading about degrees that are not from one’s own region. For these reasons, the terminology used in this
2777 report is for convenience. It generally aligns with the terminology used in northern America. It is not a prescription
2778 on how people and universities around the world should name their degrees, majors, or individual courses of study.
2779

2780 To disrupt this misunderstanding, the CC2020 task force has advocated that the public use the word ‘*computing*’ to
2781 describe the entire field. Such an adoption will take some time to become universal. However, using the word
2782 judiciously will eventually begin a convergence to a computing profession. For example, the word ‘engineering’ has
2783 universal understanding. Computing should have a similar understanding.
2784

2785 There are many other words that require clarification. Appendix H provides an equivalence table of terminology. The
2786 table should provide some guidance in trying to understand the meaning of computing words and how people use
2787 them in a global context.
2788

2789 2790 **6.1.3: Cultural Sensitivity and Diversity** 2791

2792 One should understand computing degree programs within the global contexts. Thinking that “my program” is the
2793 only way a computing program should exist can be counterproductive, especially when trying to engage in cooperation
2794 and understanding with different people. It is important to be aware that cultural similarities and differences do exist
2795 between people and the computing programs they represent.
2796

2797 Universal acceptance of global diversity is essential in all fields of endeavor, particularly the computing field that is
2798 so diverse. Those who may not be sensitive to cultural diversity should explore ways to acquire more knowledge on
2799 the situation. Computing graduates will likely interact with professionals on a global scale, so developing a sensitivity
2800 to global customs and traits, communicating effectively with peers, listening carefully, and being sensitive to time
2801 zones and holidays go a long way in bridging cultural sensitivity gaps. Graduates of computing programs can benefit
2802 greatly by learning more about global customs and etiquette of the people with whom they will be working.
2803

2804
2805

2806 **6.2: Global Economics and Computing Education**

2807

2808 The global digital economy continues to drive job creation and sustainment. The expansion of this digital economy
2809 has resulted in substantial demand for an increased volume within the labor pipeline. This situation has resulted in
2810 labor shortages, whereby the output of baccalaureate programs has been insufficient to meet the demands of the
2811 workforce. The education community has adapted to these forces by creating alternative pathways to education and
2812 training based on shorter duration programs. Community or two-year college programs (programs generally unique
2813 to the United States) have seen substantial increases in student enrollments.

2814

2815 Furthermore, the idea of “micro-credentials” through short-term and online programs are becoming increasingly
2816 popular, as are coding “bootcamps” and “academies” devoted to focused, short-term training. The effect of these
2817 market forces on the various computing disciplines are not fully clear. Generally, these shorter-term programs have
2818 areas of emphasis that are based on “just-in-time” market situations related to the various computing disciplines.
2819 However, no standard, expected outcome, or competency exists that has universal acceptance for these short-term
2820 programs.

2821

2822

2823 **6.2.1: Emerging and New Technologies**

2824

2825 The digital revolution has provided humanity with a plethora of new technologies that have improved people’s lives.
2826 Smart phones, medical imagery, aviation and aerospace, contemporary automobiles, communication infrastructure
2827 and tools, and complex video games are just a few applications touched by computing and digital technology.
2828 Computing affects all people in some way with new and emerging technologies that are still in their infancy.

2829

2830 The future promises even greater expectations on the ways computing innovations will affect people’s lives and
2831 computing education. Computing education must be agile enough to address the rapid changes of the field. Acceptance
2832 of a status quo attitude would quickly make such programs either obsolete or ones whose graduates would lack the
2833 necessary skills and human temperament for gainful employment. Modern curricula must change to match any
2834 increase in technological innovation.

2835

2836 One activity that is emerging in universities universally is the creation of makerspace laboratories, especially for
2837 engineering and business environments. Makerspaces – such as used in New Zealand [Min1] – are part of a
2838 constructivist movement that allows students, especially first-year students, to have open accessibility to readily
2839 available materials that provides exposure to modern technology, availability to items for invention and innovation,
2840 and human inquiry. Makerspaces are now emerging in elementary schools and in high schools worldwide. They
2841 modify the emphasis from teaching to learning. Computing educators could take heed in this global movement and
2842 consider creating makerspace laboratories and making their use an initial and integral part of their computing
2843 programs. Figure 6.1 illustrates two examples of makerspace laboratories; Figure 6.1 (a) shows Africa’s Maker
2844 Movement by Open Air [Ope1], Figure 6.1 (b) shows a makerspace lab at Lindenwood University [Lin1].

2845

2846



(a) Makerspace at OpenAir-Africa’s Maker Movement



(b) Makerspace Lab at Lindenwood University

Figure 6.1. Examples of Makerspaces

2847
2848

6.2.2: Forces Shaping Academic Programs

2850

This document describes seven basic categories of baccalaureate computing degree programs: computer engineering, computer science, cybersecurity, data science, information systems, information technology, and software engineering. Few academic institutions offer more than three of these types of programs, although that situation might change over the next decade. As was the case when CC2005 was written, universities offering baccalaureate degree programs tend to be cautious and conservative. The complex nature of academic degree programs makes it difficult to implement significant changes quickly.

2857

Baccalaureate programs at universities usually compete for prospective computing students, sometimes within the same institution. Such external and internal academic forces can affect the quality of computing programs because some programs will lower their academic standards to enroll more students. Some institutions of higher learning even create entities within the institution (e.g., schools of continuing education) that offer abridged courses similar to those offered in an academic program in the institution. Such courses usually do not apply toward an academic degree. Students should be wary of paying tuition and fees for such experiences.

2864

Depending on the goals and content of a computing program, prospective students must make judicious choices in selecting which program best serves their aspirations. Students who are weak in mathematics might not want to undertake a computer engineering degree or one in data analytics. Students who dislike programming or do not have an affinity for it, might not wish to consider computer science or software engineering programs. It is important for prospective students to be alert of the hype that may exist from academic institutions in their efforts to attract students to their programs.

2871

2872

6.2.3: Degree Names and the Workplace

2873

Degrees are not the only type of qualification that students can attain in computing. Around the world, colleges of various sorts offer certificates, diplomas, and advanced diplomas in different aspects of computing. In some cases, these qualifications serve to qualify their graduates for entry to a traditional university degree; in others they suffice to give their graduates direct entry into professional employment. While these qualifications are typically vocational qualifications, it would be a mistake to think of them as inferior to university degrees [Tan1].

2880

It should suffice to mention that ‘a computing degree’ is not a unique qualification for employment. Degrees and other qualifications in computing are astonishingly diverse in their duration, the extent of their focus on computing, and the scope of other material studied. Degrees are diverse in their terminology with the same name used for very different learning experiences; likewise, a variety of different names can correspond to very similar degrees.

2885

The cacophony of names among degrees and other credentials has caused confusion in the workplace to the extent that many employers tend to ignore the value of a degree name. That is, although possession of a baccalaureate degree is important, the name of degree is of little consequence. Employers are more interested in a graduate’s technical skill set and the human temperament that the graduate possesses. From the viewpoint of students and prospective students, it is best for them to enter a computing program they desire and have an ability to excel rather than studying a computing program that sounds trendy.

2892

2893

6.2.4: Innovation in Computing

2894

The computing field is burgeoning with invention and innovation. Innovation means “the process of translating an idea or invention into a good or service that creates value or for which customers will pay” [Biz2]. Innovative ideas should satisfy a specific need and be economically replicable. Innovators often combine information, imagination, and initiative. With the help of others, they marshal these talents to develop useful products to satisfy the needs of

2895

2896

2897

2898

2899

2900 potential customers.

2901

2902 In a computing context, innovation helps students and professionals to create inventive ways to solve computing
2903 problems. Often innovation is a continuous process. Such dynamic innovation occurs by many incremental advances
2904 in technology or processes such as the incremental improvement of hardware or software. When computing innovation
2905 is radical or revolutionary, its application may become a disruptive technology. Examples of recent disruptive
2906 innovations include blockchain technologies and the internet of things.

2907

2908 Risk-taking is synonymous with innovation; those who create revolutionary technologies must be ready to undertake
2909 risks. Students in baccalaureate programs may find it difficult to become innovators during their studies, although
2910 there are counterexamples to this premise such as Bill Gates and Steve Jobs. Nevertheless, faculty members should
2911 be observant of possible innovators in their programs and make allowances for those students who show genuine
2912 promise for innovative careers.

2913

2914

2915 **6.2.5: Entrepreneurship in Computing**

2916

2917 Entrepreneurship is becoming an important element of study, especially in the computing field. Entrepreneurship is
2918 “The capacity and willingness to develop, organize and manage a business venture along with any of its risks in order
2919 to make a profit” [Biz1]. The key element for success is the entrepreneurial spirit that has characteristics of innovation,
2920 risk-taking, and the drive to succeed in a global marketplace that is changing and becoming more competitive.

2921

2922 An infusion of entrepreneurial experiences in computing programs is very possible. Business schools usually teach
2923 such courses. In its simplest form, faculty members could advise students to take an entrepreneurial course as a
2924 substitute for an elective course. The same action is possible with technical electives; some, not all, students are likely
2925 to benefit more from entrepreneurial experiences than one or more technical experiences. A more aggressive approach
2926 is to construct a minor or cluster for computing students in harmony with business school offerings. For example, a
2927 student taking two entrepreneurial courses, two business courses (e.g., marketing and management), and a two-course
2928 major project experience should suffice in establishing a formal minor experience.

2929

2930 In today’s world, business acumen may be as important as technical computing knowledge. Faculty advisors should
2931 encourage students who are inclined to be risk takers or to be financially inclined to take some combination of courses
2932 for an entrepreneurial educational environment. The experience would likely benefit them throughout their lives.

2933

2934

2935

2936 **6.3: Worldwide Computing Degree Structures**

2937

2938 Differences in the terminology and nomenclature in computing education cause confusion because of different degrees
2939 themselves in the nature. This section illustrates the worldwide standard degree structures.

2940

2941

2942 **6.3.1: Computing Education in Africa**

2943

2944 Computing programs presented in Africa are mostly bachelor programs in science with specializations in computer
2945 science. Some have attributed this situation to the number of universities that have only come into existence in the last
2946 thirty years or so. Older institutions tend to present a wider variety of computing programs. In many cases these
2947 programs are within the same department. Very few universities have departments dedicated to computer science and
2948 information systems (also referred to as informatics), for instance. Computer science programs typically confer
2949 “Bachelor of Science” degrees where the curriculum includes mathematics, possibly statistics and a science.
2950 Information systems results in a “Bachelor of Commerce” degree, conferred where the curriculum includes studies in
2951 economic and management sciences. When the distinction between computer science and information systems is not
2952 clear, the program results as an information technology degree. These programs may include study from disciplines
2953 outside computing. Universities presenting engineering degrees may have a program in computer engineering. A focus
2954 in software engineering tends toward inclusion within a sciences program.

2955

2956

2957

2958

2959

2960

2961

2962

6.3.2: Computing Education in Australasia

2963

2964

2965

2966

2967

2968

2969

2970

2971

2972

2973

2974

2975

2976

2977

6.3.3: Computing Education in China

2978

2979

2980

2981

2982

2983

2984

2985

2986

2987

2988

2989

2990

2991

2992

2993

2994

2995

2996

2997

2998

2999

3000

3001

6.3.4: Computing Education in Europe

3002

3003

3004

3005

3006

3007

3008

3009

3010

The Bologna framework [Bo11] specifies three higher education qualification cycles: bachelor's (three years), master's (two years) and doctoral (three years). An important part of the European approach is the framework for qualifications of the European Higher Education Area. The so-called Dublin Descriptors provide "generic statements of typical

3011 expectations of achievements and abilities associated with qualifications that represent the end of each of a Bologna
3012 cycle” [Bol2 p65] in relation to five categories: knowledge and understanding; applying knowledge and
3013 understanding; making judgments; communication skills; and learning skills. These descriptors provide discipline-
3014 independent descriptions of what each of the degree cycles require.

3015
3016 “Informatics for All” is a new coalition involving ACM Europe, Informatics Europe, and the Council of European
3017 Professional Informatics Societies (CEPIS). Its purpose is to promote the advancement of informatics education within
3018 Europe, primarily at the level of primary and secondary high school education. Following a survey of the state of
3019 informatics education throughout Europe, “Informatics for All” developed a two-level strategy: (1) the view of
3020 informatics must be an important foundational discipline taught to all pupils, and (2) integrate informatics into the
3021 teaching of other disciplines in a manner leading to a deeper form of education in those other disciplines. These
3022 activities are gaining much support within Europe.

3023 3024 3025 **6.3.5: Computing Education in India**

3026
3027 In India, the University Grants Commission (UGC) mainly regulates education in India [Ind1] and it defines the
3028 framework within which universities operate, including the names of degrees that they may award. The university
3029 system structured contains two levels: the university itself, and a set of colleges affiliated with it. The university
3030 determines the curriculum and assessment of most degree programs conducted at affiliated colleges with the colleges
3031 serving as the “delivery” mechanism. There are mainly two broad strategies followed in designing degree programs
3032 for bachelor level computing education in India. Four-year programs are the norm in computer engineering and
3033 information technology degree programs. On the other hand, three-year programs are the norm in computer science
3034 and computer application programs, with some institutions offering an additional fourth year of study typically called
3035 an “honours” program. Further, the master’s in computer applications program is also typically a three-year program
3036 since the university envisions it as the first program for students who have a bachelor’s degree from other streams.
3037 Therefore, although named as a “master’s program.” it often becomes a first-level degree in computing.

3038
3039 India operates 895 universities, 42,338 colleges, and 3,225 engineering institutes [Ind1, Ind3]. Table 6.1 illustrates
3040 enrollment figures for three categories of study.

3041
3042 Table 6.1. Enrollment data for university studies in India (2017)

	Boys (x100,000)	Girls (x100,000)	Total (x100,000)
UGC (All inclusive)	15.27	14.16	29.43
UGC (Non-Engineering CS)	Not Available	Not Available	9.68
AICTE	5.30	2.20	7.50

3043
3044 The National Assessment and Accreditation Council (NAAC) [Ind2] is an autonomous institution under the UGC that
3045 is responsible for quality assurance of higher education institutions in India. Additionally, the All India Council for
3046 Technical Education (AICTE) regulates technical streams like technology, engineering and pharmacy [Ind3]. The
3047 National Board of Accreditation (NBA), which is an autonomous institution with the AICTE, promotes international
3048 standards of technical education in India [Ind4].

3049 3050 3051 **6.3.6: Computing Education in Japan**

3052
3053 In Japan, computing related bachelor’s degree programs are of two types: those that focus on computing such as
3054 computer science, and those whose primary focus are in other fields. Most of the former type come under the broad
3055 umbrella of either a Bachelor of Engineering degree or Bachelor of Science degree. Some universities have more
3056 specific names, such as a Bachelor of Informatics or a Bachelor of Computer Engineering [Bac1]. For those of the
3057 second type, the degree name may be, for example, Bachelor of Business and Informatics, but the actual focus of the
3058 degree may be in fields such as business and design. In those cases, computing (informatics) would be a comparatively
3059 minor part of the degree. This situation also appears in a published survey [Kak1], where they found that nearly half
3060 of the students in a “computing” department are learning computing domains other than those defined in CC2005.
3061 Such students belong to interdisciplinary departments such as a department focusing on business with a computing
3062 component.

3063
3064
3065
3066
3067
3068
3069
3070
3071

Even for the first type degree, there is a wide difference between universities on how students achieve a degree, especially at the beginning. At some universities, students start with a basic set of courses such as physics, chemistry, mathematics, and informatics in their first year of study. They then begin their actual major in their second year. At other universities, students will start their actual major in their first year. Computer programming, which is basic for any computing related degree, accentuates this difference; some universities have their first programming course in the student's first year while others have it in the second year.

3072
3073

6.3.7: Computing Education in the Middle East

3074
3075
3076
3077
3078
3079

Middle Eastern countries have started efforts to make computing a core or compulsory course in the secondary schools. Led by Saudi Arabia, the course covers fundamental topics in the computing field, an introduction to programming, as well as technologies and programming for smart devices. The Kingdom's Ministry of Education recently approved this implementation of a digital transformation plan. This computer course consecutively became part in intermediate and elementary schools to produce a technologically advanced generation.

3080
3081
3082
3083
3084
3085
3086
3087
3088

In Middle Eastern countries, university computing programs generally follow ACM/IEEE curricular guidelines. For example, since the mid-2000s, most countries have followed the Curriculum Guidelines for Undergraduate Degree Programs in Information Technology, known as IT2017 [Acm07] and formerly by IT2008 [Lun1]. These reports recommend computing areas beyond programming and provide the potential to conduct projects, internships, and research together with an emphasis on components to enhance the practical experience of students. Such degree programs also foster adaptability to change in job market needs by providing in-depth knowledge through specific concentrations that are easily interchangeable. Hence, respected IT programs in the Middle East have enjoyed success with these principles and they serve as models for IT programs in the region.

3089
3090
3091
3092
3093
3094
3095
3096
3097

The importance of computing for development and modern technology and the need for specialists in this area is growing, due to the penetration of computers in all aspects of life. In response to the reality of the projected need for national competence in the field of computing, a prominent university in Saudi Arabia created two new futuristic and specialized academic programs: computer science and computer engineering. While the program titles are not new, their futuristic approach has generated great demand because they are addressing new technological areas. The experiment has led to three additional programs in information systems, software engineering and information technology. Now, other universities in the country are following these innovations. The concept is spreading to other universities in the region to prepare graduates professionally for current and new technologies for the next decade.

3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109

In Israel, for more than twenty years, high schools have taught computer science as an elective, just like physics, biology and chemistry. In recent years, elementary and junior high schools are introducing computer science in an extended piloting stage. Curricular designers allow exposure for learners to both theoretical and practical facets of the discipline. The underlying principles of the curricula emphasize that computer science is a full-fledged autonomous scientific subject, which should concentrate on the key concepts and foundations of the field. Ideally, schools should teach two different programming paradigms. Those certified to teach the subject must have formal education of the discipline and formal teacher training. For others, the government has established a teachers' center, so no teacher feels isolated. The center serves as a vehicle where teachers can contact their colleagues, find materials, and receive invitations to attend workshops and conferences. The Ministry of Education, computing scientists, computing education researchers affiliated with universities and colleges, as well as computer science teachers are all involved in this challenging effort. At the university level, computing degree programs use the ACM/IEEE curricular recommendations as guidance for their curricula.

3110
3111

6.3.8: Computing Education in North Americas

3112
3113
3114
3115
3116
3117
3118

Two-year degrees and four-year degrees encompass all possible degree structures in northern America. Most four-year degrees in northern America (Canada and the United States) have the designation of Bachelor of Science (BS) degree, Bachelor of Arts (BA) degree, Bachelor of Engineering (BE) degree, or other baccalaureate degree descriptors. Computing topics within a community college program should be equivalent to at least one full year of study in the four-year program of study. It also includes relevant mathematics and science as other important components of a

3119 computing program. Additional program requirements, often called general education requirements, depend on the
3120 characteristics and mission of the program and the institution. Almost half the undergraduates in the United States are
3121 enrolled in two-year colleges and more than half of all first-time college first-year students attend community and
3122 technical colleges. Students in two-year community college programs often earn associate degrees [Aac1].
3123 Articulation agreements often exist between institutions of the two-year programs and the four-year programs to
3124 facilitate seamless transfer from two-year programs to four-year programs.
3125
3126

3127 **6.3.9: Computing Education in South Americas**

3128

3129 In South America and Latin America, a typical degree is four years in duration, with the first year devoted to general
3130 subject matter such as literature, writing, mathematics, and logic. The subsequent three years focus on computing
3131 topics. Additionally, degrees in Latin America have several hybrid names mostly oriented on teaching ways to use
3132 technology. Students in Latin America choose a degree path before beginning university studies. Most of the tightly
3133 focused degrees begin with the very first semester. The best example following international recommendations is
3134 Brazil where it is possible to see clear groups of well-defined programs. Almost all computing programs orient
3135 themselves to guidelines presented in computing curricula from ACM and IEEE. Furthermore, because of historical
3136 reasons, degrees in Chile distinguish between civil (people) engineers and military engineers. Similar situations occur
3137 in Peru, Colombia, Equator, and Venezuela where decades ago IBM had influenced the early programs whereby the
3138 degree designation was systems engineering, a tradition that continues even today.
3139
3140

3141 **6.4: Professionalism and Ethics**

3142

3143 Professionalism and ethics should be a permanent element of any computing curriculum. The following discussion
3144 taken from the IT2017 report sheds some light on ways these elements can be part of a computing program of study.
3145
3146

3147 **6.4.1: Ethics in the Curriculum**

3148

3149 The incorporation of professionalism and ethics must be a conscious and proactive effort because much of the material
3150 blends into the fabric of existing curricula. For example, the introductory courses in the major could include discussion
3151 and assignments on the impact of computing and the internet on society and the importance of professional practice.
3152 As students progress into their second-year courses, they could start to keep records of their work as a professional
3153 might in the form of requirements, design, test documents, and project documents such as charters and project reports.
3154
3155

3156 Additional material such as computer history, digital libraries, techniques for tackling ill-defined problems, teamwork
3157 with individual accountability, real-life ethical issues, professional standards and guidelines, legal constraints and
3158 requirements, and the philosophical basis for ethical arguments may also appear either in a dedicated course or
3159 distributed throughout the curriculum. The distributed approach has the advantage of presenting this material in the
3160 context of a real application area. On the other hand, the distributed approach can be problematic in that faculty often
3161 minimize professionalism and ethics in the scramble to find adequate time for the technical material. Projects,
3162 however, may provide a natural outlet for much of this material particularly if faculty can recruit external clients
3163 needing non-critical systems. When they engage in service-learning projects in the community or work with external
3164 clients, students begin to see the necessity for ethical behavior in very different terms. As a result, those students learn
3165 much more about ways to meet the needs of a client's ill-defined problem. However, no matter how teachers integrate
3166 professional practice into the curriculum, it is critical that they reinforce this material with appropriate assessments.
3167

3168 For departments with adequate numbers of faculty members and resources, courses dedicated to teaching professional
3169 practice may be appropriate. For limited resources, this content should be covered in courses like professional practice,
3170 ethics, and computer law, as well as senior capstone and other appropriate courses. Additionally, more advanced
3171 courses on project management, financial management, quality, safety, and security may be part of the experience.
3172 These courses could come from disciplines outside of information technology and they would still have a profound
3173 effect on the professional development of students.

3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212

6.4.2: Professional and Ethical Work

Learning environments that support students in acquiring professional practice experiences include the following elements [Acm07]:

- Assessments.
- Appropriate inclusion of professional practice in traditional course assessments (assignments, projects, exams, presentations, reports, etc.)
- Sound measurements of student work to show student progress and improvement.
- Student involvement in the review and assessment process.
- Participation of professionals from industry, government, or other employers of IT graduates to assess student performance in internships, co-op programs, and on projects with outside clients.
- Standardized tests validated by professional societies.
- Post-graduation alumni surveys of alumni to see how well alumni thought their education prepared them for their careers.
- Program accreditation to demonstrate compliance with certain educational standards for professional practice.
- Course labs that meet employer needs to make sure students acquire professional experiences.

The assessment process should encourage students to employ good technical practice and high standards of integrity and ethics. The assessment process should hold students accountable on an individual basis even if they work collectively in a team. It should have a consistent set of measurements, so students become accustomed to using them and they learn how to associate them with their progress.

6.5: Digest of Chapter 6

This chapter encourages readers to be aware of the wealth of contexts in which computing education takes place around the world. It looks briefly at some of the terminology, establishing that the same word can be used to mean different things, and that different words can be used to mean the same thing. The report proposes that stakeholders accept the word ‘computing’ as the name of the overarching discipline. Furthermore, there is no such thing as a standard degree structure, and this chapter has elaborated on many differences between countries and regions. The extent to which educational needs might be driven by economic needs and the universities’ responses to the latter is also discussed here. This chapter makes the case that ethics and professionalism must be explicitly incorporated into the computing curriculum.

3213

3214 **Chapter 7: Curricular Design – Challenges and Opportunities**

3215

3216 This chapter highlights some of the contemporary challenges for the development of modern computing curricula
3217 programs. It also addresses ways in which industry and government can play a special role in generating modern
3218 programs through professional advisory boards, work-study programs, and internships. Academic institutions must
3219 also be proactive in supporting strong, contemporary computing programs for the benefit of its graduates.

3220

3221

3222

3223 **7.1: Transforming to Competencies**

3224

3225 The CC2020 project has provided an overview of the computing education landscape related to undergraduate,
3226 (baccalaureate) programs. This overview is global in scope. Furthermore, the task force has encompassed many
3227 perspectives with the goal of providing a modern update to its predecessor, CC2005. The CC2020 project has also
3228 presented possible frameworks for future curricular reports.

3229

3230

3231 **7.1.1: Distinguishing Competency from Knowledge**

3232

3233 Competency—computing competency—is the central theme for the frameworks presented in this report. Competency
3234 incorporates human attributes (disposition) and technical skill with knowledge. Universities and their faculty members
3235 are experts in knowledge. To prepare students to be competent professionals after they graduate, it is important to
3236 instill competencies as an essential element of their education. In this sense, the CC2020 project and this report
3237 encourage computing programs to establish a proper environment and to necessitate that future curricular reports
3238 incorporate competency as part of their structure and recommendations.

3239

3240 In terms of curriculum, content knowledge refers to the body of knowledge and information (facts, concepts, theories,
3241 and principles) that teachers teach and that students should learn in a given subject. It is not concerned with related
3242 skills. This content forms a knowledge base which is a collection of information about a particular subject. Hence,
3243 knowledge-based learning is one of the methods of learning that teachers and education departments employ to
3244 provide students with the best learning environment possible.

3245

3246 Knowledge is only one of three components or dimensions of competency. As already mentioned, competency also
3247 requires dispositions and technical skills. Again, faculty members are good at conveying knowledge. Developing
3248 student technical skills is a challenge. However, faculty members have little experience with skill and dispositional
3249 development and are often at a loss in developing dispositional qualities for their students. The CC2020 project and
3250 this report demystifies that conundrum.

3251

3252

3253 **7.1.2: Curricular Dynamics**

3254

3255 Computing curricula are always in a state of flux. The continually changing field of computing is still in its infancy
3256 with new ideas and inventions developing almost daily. Hence, computing curricula must be agile and be able to
3257 respond to change. Students and graduates of computing programs must be able to face change and become inventive
3258 in contributing to that change.

3259

3260 One way to address this challenge is to include experiences in innovation, entrepreneurship, and maker-space activities
3261 in computing programs. While foundation or core courses are important, what might be equally important is to have
3262 students experience new technologies, inventive creations, and even space to ponder with their own imagination.
3263 Engineering disciplines have been doing this for some time with their introduction to engineering exploration
3264 laboratories in the very first semester of study. Non-engineering computing programs are just experimenting with this
3265 proven idea. Computing programs should have a solid conceptual foundation and be responsive to meet the challenge
3266 of developing modern and futuristic student experiences if they expect the graduates of their programs to succeed in
3267 a quickly changing world of computing.

3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279

7.1.3: Transferring Computing Competencies

The role of academics and the way they enable computing competencies is important to produce capable graduates of computing programs. As discussed in Chapter 4, computing competency is a triad of computing knowledge, skill, and disposition. There is no single method to enable competency which is a combination of these three triadic elements. The goal is to produce graduates of computing programs that are proficient at the time of graduation to enter the workplace, to attend graduate school, or to contribute constructively in some way to society. The following discussion provides some suggestions in transferring competency to students.

3280
3281
3282
3283
3284
3285
3286

7.1.4: Knowledge Transfer

The transfer of knowledge is the cornerstone of academia and universities. Academicians have been transferring knowledge to students for centuries and millennia. They do this through personal knowledge, use of textbooks, personal notes, and other mechanisms for knowledge transfer. They administer tests or examinations to ascertain whether students have acquired the requisite knowledge. This is a classical mode for student learning.

3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299

In today's world, students can acquire knowledge on a subject via non-classical methods. One obvious method is using the internet to search for supportive or extra material on a topic such as video clips, wikis, experiences on professional development, MOOCs, and other supportive online materials available to the public. Encouraging students to explore such additional materials helps them develop lifelong learning skills when students must continue to learn long after they graduate from their computing programs.

Academicians should also move away from the lecture-mode of knowledge transfer. They should encourage students to learn in small groups (e.g., pair learning), construct learning groups (e.g., teams) of three or four students, and other learning strategies. Exploring with new methods of learning can augment the learning of knowledge and allow students to interact with each other to develop new skill sets as well as developing communication and teamwork skills by studying with others.

3300
3301
3302
3303
3304
3305
3306

7.1.5: Skill Transfer

While academicians have a mastery of knowledge, they are sometimes weak in transferring specific skills to students. University professors often assume students will develop skills on their own and without direction. With such assumptions, how would one know whether students have developed certain skills expected in the study of computing?

3307
3308
3309
3310
3311
3312

All computing academic units and departments should specify a set of skills all students should expect to master by the time of graduation. Due to the uniqueness of individual computing programs, it is not possible to specify how their academic units would implement the development of such skills. Notwithstanding, computing academicians and academic units should provide instruction on ways students would develop these skills as an important element of computing competency.

3313
3314
3315
3316
3317
3318
3319
3320

7.1.6: Disposition Transfer

Computing academicians often lack understanding of dispositions or ways to convey dispositions in computing programs. This is understandable because such understanding was not part of their own education. Chapter 4 identified eleven dispositions which include: Proactive, Self-directed, Passionate, Purpose-driven, Professional, Responsible, Adaptable, Collaborative, Responsive, Meticulous, and Inventive. The set of all dispositions far exceeds these eleven. With very few exceptions, academicians simply do not know how to transfer these dispositions to students.

3321
3322

Because of the uniqueness of individual computing programs, it may not be possible to specify how their academic units would develop these dispositions. Notwithstanding, computing academic units (e.g., departments) should specify

3323 the set of human behaviors (which is broader than the ten dispositions stated) expected of their students by the time
3324 of graduation. Computing academicians and academic units could provide instruction on ways students acquire these
3325 traits as an important element of computing competency. They may do so by personal and peer examples, by viewing
3326 workplace attitudes, or by attending seminars on behavior as professionals.
3327

3328 Students could take courses offered in other academic units such as in social science and psychology that could be
3329 useful in developing dispositions. This may be problematic. Although courses in these areas may cover conceptually
3330 (at a knowledge level) topics related to these dispositions, that does not mean that they help students develop them.
3331 Additionally, some of dispositions may not transfer well across contexts; a disposition demonstrated in one class may
3332 not transfer to the authentic context in another. Only through repeated practice across domains will a transfer occur.
3333 Furthermore, people learn dispositions through modeling and enculturation. Institutions purposely need to build and
3334 develop these traits over time through "collaborative" or "responsive" activities over time through course experiences,
3335 internships, and other interactive experiences.
3336
3337

3338 **7.1.7: Need for Local Adaptation** 3339

3340 The task of designing any curriculum is a difficult one, in part because so much depends on the characteristics of an
3341 individual institution and the interests and expertise of its faculty members. Even if every institution could agree on a
3342 common set of knowledge, skills, and dispositions for undergraduate education, many additional factors would
3343 influence curriculum design. These factors include the following.

- 3344 ▪ *Type of institution and the expectations for its degree programs:* Institutions vary in mission, structure, and scope
3345 of undergraduate degree requirements. A curriculum that works well at a small college in one country may be
3346 completely inappropriate for a research university elsewhere in the world.
- 3347 ▪ *Range of postgraduate options that students pursue:* An institution whose primary purpose is to prepare a skilled
3348 workforce for a profession is likely to have different curricular goals than one seeking to prepare students for
3349 research and graduate study. Each individual school must ensure that the curriculum it offers allows students the
3350 necessary preparation for their eventual academic and career paths including those outside their current interest.
- 3351 ▪ *Preparation and background of entering students:* Students at different institutions—and often within a single
3352 institution—vary substantially in their level of preparation. because of this, computing departments often need to
3353 tailor their introductory offerings so that they meet the needs of their students.
- 3354 ▪ *Faculty resources:* The number of faculty members supporting a computing program may vary from fewer than
3355 five to a hundred or more at a large research university. Program size heavily influences the flexibility and options
3356 available to a program. Independent of the program size, faculty members need to set priorities for ways in which
3357 they will use their limited resources.
- 3358 ▪ *Interests and expertise of the faculty:* Individual curricula often vary due to the specific interests and knowledge
3359 base of the department.
3360

3361 Creating a workable curriculum requires finding an appropriate balance among these factors, a balance which will
3362 require different choices at every institution. No single curriculum can work for everyone. Every college and university
3363 will need to consider the various models proposed in this document and design an implementation that meets the needs
3364 of their environment.
3365
3366
3367

3368 **7.2: Industry Engagement for Workplace Competencies** 3369

3370 An important way that industry can support the education process is to play a greater role in helping students. These
3371 professionals can offer support in several ways such as:

- 3372 ▪ Provide faculty members with the tools and insights to develop student competencies in the subjects they teach;
- 3373 ▪ Function in the role of mentors to students working on projects.
- 3374 ▪ Give special presentations to classes telling students and faculty about their firm, their work, and their
3375 development processes.
- 3376 ▪ Take part-time positions as part-time instructors to strengthen a university's course offerings by conveying
3377 material through a practical approach.

- 3378 ▪ Conduct site visits.
- 3379 ▪ Provide in-house training materials and/or classes to faculty and students in specialized research, process, or
- 3380 software tool areas.
- 3381 ▪ Explore industry-sponsored capstone experiences.
- 3382 ▪ Serve on industrial advisory boards, which service that allows them to provide valuable feedback to the department
- 3383 and institution about the strengths and weaknesses of the students. In each of these ways, enterprises in the private
- 3384 and public sectors can establish important lines of communication with the educational institutions that provide
- 3385 them with their future employees.
- 3386

3387 In addition to the various opportunities that take place on campus, industry and government also contribute to the

3388 development of strong professional practice by bringing students and faculty into environments outside of academia.

3389 For example, students and faculty may take field trips to local firms and begin to establish better relationships.

3390

3391 For faculty, their cooperation with industry and government can serve as a vehicle for developing student

3392 competencies in their courses. Such connections also provide opportunities for mutual benefit and they create a higher

3393 level of trust between the faculty member and the company. Because of these initiatives, employers, students, and

3394 faculty know more about each other and are more willing to promote the program.

3395

3396 Over a longer term, cooperative, practicum, and internship opportunities give students a better understanding of what

3397 life on the job will be like. In addition, students may become more interested in their studies and use that renewed

3398 interest to increase their marketable potential. Students may also form a bond with specific employers and be more

3399 likely to return to that firm after graduation.

3400

3401

3402 **7.2.1: Professional Advisory Boards**

3403

3404 Professional or industrial advisory boards are essential for the development of strong and meaningful computing

3405 programs. Professionals from industry and government are a great resource for insight on the needs of the workplace.

3406 These groups can become strong catalysts for bridging the computing program to needs of industry and government.

3407 They also establish personal connection between the computing program, its students, and the professional world.

3408

3409 Every computing program should have a professional advisory board. Ideally, advisory boards should meet once each

3410 semester, but an annual meeting is also appropriate. Its chairperson should not be a faculty member from the program.

3411 It is important to capture board activities by taking minutes of all meetings. Updates by email or other online media

3412 are also possible. A professional advisory board should also monitor the goals of the computing program to ensure

3413 they are in harmony and in balance with the mission of the institution and the requirements of the workplace.

3414

3415

3416 **7.2.2: Work-Study and Cooperative Programs**

3417

3418 All computing programs should consider the possibilities of including a work-study or cooperative (co-op) program

3419 as part of their curricula. Typically, these programs allow students to enter industry or government before they

3420 graduate. The experience could be one or two semesters when the student is academically mature, usually during the

3421 third year of a four-year program. These programs often provide student credit and they also allow students to earn

3422 wages as they contribute to the company or government. Some universities make cooperative experiences a

3423 requirement and they tailor their sequence of courses accordingly.

3424

3425 Cooperative programs do have their challenges, despite their benefits. One challenge is that students will likely

3426 graduate beyond the normal period (e.g., four years). Those students who undertake a one-semester cooperative

3427 experience might lose two semesters of time if the program does not offer required courses every semester. Each

3428 computing program should evaluate whether a cooperative work-study program is suitable for its needs and for the

3429 benefit of its students.

3430

3431

3432 **7.2.3: Internship Programs**

3433
3434 In contrast to cooperative programs that usually last an entire semester, programs should seriously consider internship
3435 programs as a required component of the computing curriculum. Internships are experiences that take place over a
3436 short period such as during a summer when regular classes are not in session. Internships could also be part-time
3437 experiences: in this case, students could join a company one day a week or two half-days a week during a semester.
3438 Internships are rather popular and many computing programs around the world require them as part of student learning.
3439 Students usually receive credit for an internship and in most cases, industry pays students for their services.
3440

3441
3442
3443 **7.3: Institutional Adaptations**

3444
3445 This report and the related curricular volumes provide a significant resource for colleges and universities seeking to
3446 develop or improve their undergraduate programs. Implementing a curriculum successfully, however, requires each
3447 institution to consider broad strategic and tactical issues. This section enumerates some of these issues and illustrates
3448 ways to address these issues.
3449

3450
3451 **7.3.1: Attracting and Retaining Academic Educators**

3452
3453 One of the most daunting challenges that computing departments face is the problem of attracting, and then retaining,
3454 qualified faculty members. In computing, there are sometimes more advertised positions than the number of highly
3455 qualified candidates. The shortage of faculty applicants, coupled with the fact that computing people command high
3456 salaries outside academia, make it difficult to attract and retain faculty members. Institutions should develop
3457 aggressive plans to both recruit and retain faculty members; incentives such as hiring packages and modified teaching
3458 responsibilities may prove advantageous for this endeavor. Additionally, active participation in professional
3459 organizations provides networking opportunities with leaders of peer programs, which in turn may result in greater
3460 visibility and access to potential faculty candidates. Other possible strategies include collaborative and/or
3461 interdisciplinary efforts with other programs and/or institutions.
3462

3463 While a computing program may draw on faculty from related disciplines, as a professional field there must be a core
3464 faculty with appropriate professional training and experiences. Additionally, faculty members must maintain currency
3465 with developments in the field. Institutions must make appropriate accommodations for the professional development
3466 of faculty, whether achieved through research, conference participation, sabbaticals (perhaps in industry), consulting,
3467 or other activities. Institutions must also recognize, respect, and reward teaching faculty members in the same way it
3468 does for research faculty members.
3469

3470
3471 **7.3.2: Need for Adequate Laboratory Resources**

3472
3473 It is important for educational institutions to recognize that the financial resources required to support computing
3474 programs are significant. Software acquisition and maintenance can represent a substantial fraction of the overall cost
3475 of computing, particularly if one includes the development costs of courseware. Acquisition and maintenance of the
3476 hardware and instrumentation infrastructure required for experimentation and hands-on system development by
3477 students can be costly. Providing adequate support staff to maintain the laboratory facilities as well as technical
3478 assistants and tutoring support represent other expenses. To be successful, computing programs must receive adequate
3479 funding to support the laboratory needs of both faculty and students and to provide an atmosphere conducive to
3480 learning.
3481

3482 Because of rapid changes in technology, computer hardware generally becomes obsolete long before it ceases to
3483 function. The useful lifetime of computer systems, particularly those used to support advanced laboratories and state-
3484 of-the-art software tools, may be as little as two or three years. Planning and budgeting for regular updating and
3485 replacement of computer systems is essential. Computing curricula typically include many required laboratories. The
3486 laboratory component leads to an increased need for staff to assist in both the development of materials and the

3487 teaching of laboratory sections. This development will add to the academic support costs of high-quality computing
3488 programs. Close contacts with relevant industries can lead to the ready availability of interesting and up-to-date case
3489 study material; it also offers opportunities for students to engage in internships. Refreshing laboratory material on a
3490 regular basis serves to motivate and excite new students.

3491
3492 Finally, with the availability of up-to-date reference materials on the internet, institutions should provide access to
3493 such resources as the IEEE Xplore Digital Library, the ACM Digital Library, and the AIS e-library. Webinars, e-
3494 books, online tutorials, MOOCs, and other resources are all increasingly available and relevant; these are available
3495 through, for instance, the ACM Learning Center.

3496
3497

3498 **7.3.3: Transfer and Educational Pathways in the United States**

3499

3500 Multiple factors such as admission criteria, student academic preparedness, the diversity of student population, life
3501 experiences, and different configurations of educational pathways shape access, retention, and degree attainment
3502 across undergraduate computing programs. The transition points from primary and secondary education through
3503 university education vary for successful employment or for further education with advanced degrees throughout the
3504 world. Examining pathways into and through undergraduate computing degree programs from a global perspective is
3505 valuable to know.

3506

3507 *Four-year Transfers*

3508 Understanding the entry points into, and the pathways through, undergraduate computing programs helps structure
3509 these pathways to serve students, companies, and the computing industry. Specifically, appreciating variations and
3510 compatibilities in these pathways across regions of the world helps enable and support a global computing workforce.

3511

3512 The CC2020 task force has considered pathways toward undergraduate computing programs and considered the rigor
3513 of some computing programs. The task force defined a “demanding program” as one that requires a significant number
3514 of courses beyond general education in which more than 50% of the program relates technically to computing. It
3515 hypothesized that fewer students transfer into demanding programs as compared with transfers into non-demanding
3516 programs. If a program is too demanding, students may be less likely to complete the program successfully. Transfers
3517 would likely take place only between universities of equal caliber. There are many exceptions, of course.
3518 Notwithstanding, one would expect that successful transfers between two demanding computing programs and
3519 transfers between two nondemanding computing programs would be highly more likely than a transfer from a non-
3520 demanding program to a demanding program.

3521

3522 *Technical Institute Transfers*

3523 It is unlikely that a person enrolled in a program at a technical institute could successfully transfer to four-year
3524 computing program at a university. The contexts of the two programs would be very different. Students attending
3525 technical institutes would likely not have studied mathematics and science at university levels. Additionally, those
3526 students would likely not have the requisite general education courses expected from this experience. Hence, course
3527 transfer would likely not occur. In many parts of the world, transfer from technical institutes to university computing
3528 programs is almost non-existent.

3529

3530 *Community College Transfers*

3531 In countries where community colleges or two-year college programs thrive, transfer to a university computing
3532 program from a community college is very common and even encouraged, especially in Canada and the United States.
3533 In the United States, for example, students in some states have a legal right to transfer credit for the same course from
3534 a community college to a university program. In fact, the two courses might even have the same code and title such
3535 as CHEM 101: Chemistry I. Indeed, many states distinguish community college transfer programs by sponsoring the
3536 Associate in Science (A.S.) degree as opposed to the Associate in Applied Science (A.A.S.) degree, which is a career-
3537 oriented degree.

3538

3539 Going a step further, it is very common in the United States to have *articulation agreements* between community
3540 colleges and universities. These agreements have a certain legal status in that a student who successfully passes courses
3541 at a community college has a right to transfer that course to a university that is a signatory to the articulation agreement.
3542 Since student enrollment at community colleges is approximately equal to the undergraduate student enrollment at

3543 universities, this vehicle of study is very popular around the country. It is common for a community college to have
3544 articulation agreements with several universities to which student transfer is likely.
3545
3546
3547

3548 **7.4: Program Quality Assurance and Accreditation**

3549
3550 Academic accreditation is a process used to support continuous improvement of institutions and their degree programs.
3551 Accredited degree programs must meet certain external requirements to increase the level of confidence the public
3552 has in them.
3553
3554

3555 **7.4.1: Accreditation Overview**

3556
3557 In some countries, accreditation can occur at different levels of an academic institution. In these cases, institution-
3558 wide accreditations certify that a university meets minimum standards for resources (e.g., laboratories or libraries) and
3559 operating procedures (e.g., admissions policies) required of any legitimate institution of higher learning. Similar
3560 guidelines may exist for an administrative unit within the institution (e.g., a business school) that encompasses degree
3561 programs in related fields. Accreditation for an academic unit that houses a group of programs models institutional
3562 accreditation but with greater specificity.
3563

3564 The most detailed form of accreditation concerns the evaluation of individual degree programs. This involves the
3565 participation of independent organizations or government agencies that establish quality standards and criteria for
3566 degree programs in a specific discipline. Program-specific accreditation involves an evaluation of specific degree
3567 programs and certifies that a degree program meets established criteria and has rigorous processes for ongoing
3568 improvement. Accreditation does not exist for every discipline, but it does exist for computing degree programs.
3569

3570 In nations where accreditation can occur at different levels, an organization (e.g., a government-related entity) may
3571 accredit a university, but not its computing degree programs. For example, a university in the United States may have
3572 unaccredited degree programs even though the university has accreditation statues. A different organization would
3573 conduct accreditation of computing programs. The distinction to keep in mind is that the accreditation of a college or
3574 university *does not imply* that its computing degree programs meet the standards of quality established for the
3575 computing disciplines unless the computing programs have a program-specific computing accreditation.
3576
3577

3578 **7.4.2: Benefits of Program-Specific Accreditation**

3579
3580 Discipline-specific or program-specific accreditation provides two important benefits for programs and for the
3581 institutions in which they reside. These include the following.

- 3582 ▪ It certifies that a degree program meets minimum quality standards established by independent professional
3583 or scientific societies or by government agencies. This helps an institution market its programs, and it gives
3584 the public and prospective students reason to be confident in a degree program's quality.
- 3585 ▪ The program receives an onsite consultation by a visiting team that provides expert opinions about a
3586 program's strengths and weaknesses and about its specific needs for improvement. This interaction helps an
3587 institution have full understanding of how its programs are performing and what institutions must do to
3588 improve their quality.

3589 Thus, accreditation provides the benefits of both a marketing aid for attracting students and an expert consultation
3590 focused on improving quality. Notwithstanding, some institutions may not need or desire the former benefit.
3591

3592 Some institutions commit to accreditation solely because the accreditation process helps them maintain and improve
3593 the quality of their programs which, in turn, further cements their reputation. In some nations, institutions have no
3594 choice because accreditation is a requirement for program existence. Discipline-specific accreditation processes
3595 determine whether a candidate degree program meets certain criteria. Not only does accreditation determine whether
3596 the program provides enough qualified teachers with acceptable workloads, it also determines how the program uses
3597 materials and assignments, how it evaluates assignments and examinations, and how it engages itself in continuous
3598 evaluation and improvement.

3599

3600 Professional bodies also use program accreditation to ensure that degree programs meet, at least in part, the
3601 requirements for membership in their profession. In some cases, graduation from an accredited degree program is a
3602 requirement for individuals before they can practice in a profession. This means that it is not sufficient for students
3603 who wish to practice a profession simply to earn a degree in the appropriate discipline; rather, they must have earned
3604 that degree from an accredited degree program. A given degree program does not choose whether its accreditation
3605 status has such professional elements; the accreditation process determines what is customary for its discipline in its
3606 nation.

3607

3608 Perhaps the greatest misconception about accreditation is the belief that institutions pursue program accreditation only
3609 to obtain a credential for public image. Those unfamiliar with discipline-specific accreditation often do not understand
3610 the important role that the accreditation process plays in helping a program know what it must do to improve the
3611 quality of both its offerings and its graduates.

3612

3613

3614 **7.4.3: Quality Assurance**

3615

3616 Program-specific accreditation is a means of demonstrating that a degree program meets an independent standard of
3617 quality, but the meaning of that standard varies. Its rigor is determined by the accrediting body's policies and practices
3618 and by any government regulations that might apply. In some cases, accreditation certifies that a degree program has
3619 met a minimum quality standard. In other cases, there exist both minimum standards and higher standards.

3620

3621 While discipline-specific accreditation addresses program quality, it is important not to reach unwarranted conclusions
3622 about the relationship between accreditation and quality. One must be familiar with both the discipline and the national
3623 context in order to reach appropriate conclusions. Lack of program accreditation does not mean a program is of low
3624 quality. Conversely, an accredited program does not mean a program is of high quality. All accredited programs must
3625 meet minimal requirements according to a given set of criteria; there is no ranking according to quality.

3626

3627 Notwithstanding, there are several aspects that reflect high quality. These include good teachers, a faculty workload
3628 that permits teachers to focus adequately on their classes and remain current in their field, as well as sufficient faculty
3629 support and infrastructure. Additionally, it is important to have evidence of rigorous procedures for monitoring and
3630 improving quality in an ongoing way.

3631

3632 For strong programs, integrating quality monitoring processes with initiatives for improving quality should form a
3633 continuous cycle. Activities include monitoring effort for effect, planning and implementing new improvement efforts,
3634 and evaluating the results; the cycle then repeats. Doing this properly is not difficult. However, it requires a measure
3635 of commitment, discipline, and information sharing to be successful.

3636

3637

3638 **7.4.4: Global Recognition**

3639

3640 Many countries have embraced accreditation. Although the details vary, there is a common belief that a panel of
3641 experts who represent a profession evaluates a program's quality against established standards and criteria produces
3642 strong computing programs. The circumstances vary with respect to whether accreditation is mandatory, strongly
3643 encouraged, or completely voluntary. Some countries have rigorous program criteria and require that accreditation
3644 standards apply to every program offered at any college or university. In other countries, accreditation is voluntary.

3645

3646 The administration of the accreditation process also varies. In some countries (e.g., Australia, Canada, and the U.K.),
3647 professional societies conduct program accreditation for their respective fields. In other countries (e.g., the U.S.) a
3648 designated organization monitors and/or performs accreditation. In some countries (e.g., Estonia and the United Arab
3649 Emirates), a government agency conducts the accreditation process.

3650

3651 In some computing disciplines, accreditation agencies also cooperate across their national borders. Mutual recognition
3652 of evaluation and accreditation processes has encouraged a range of international agreements such as the Washington
3653 Accord for engineering programs, the Seoul Accord for computing programs, the Sydney Accord and the Dublin
3654 Accord for technology programs. Other accords include the European Federation of National Engineering

3655 Associations (FEANI) and the International Register of Professional Engineers (IRPE). Such agreements have a range
3656 of signatories, but they share a common goal to facilitate the movement of professionals across nations. That is, they
3657 recognize the substantial equivalence of programs accredited by these bodies. For example, the Australian Computer
3658 Society ACS) accredits computing programs in Australia; ABET accredits computing programs in the United States
3659 and elsewhere. Graduates from ACS accredited programs and graduates from ABET accredited programs enjoy a
3660 mutual recognition for employment and other professional benefits.

3661
3662 Accreditation in the U.S. is voluntary in the sense that no law or regulation requires a degree program to acquire
3663 accreditation. As a practical matter, it is more voluntary in some computing disciplines than in others. In computer
3664 engineering, for example, a strong sense of a professional community exists, and state-regulated licensing of engineers
3665 can require applicants to hold an engineering degree from an ABET accredited program. In contrast, the computing
3666 community outside engineering is more of a loosely organized network of scientists and researchers than a tightly
3667 organized body of practicing professionals. Historically, there has been no compelling professional pressure for
3668 accreditation of non-engineering computing programs.

3672 **7.5: Degree Names and Job Titles**

3673
3674 Section 6.2 of this report briefly addressed the topic of degree titles and the workplace. This section extends that
3675 discussion in the context of Latin America.

3678 **7.5.1: Job Positions and Titles**

3679
3680 In today’s world, there is often confusion regarding what a person does and what a job position entails. For example,
3681 the phrase “software engineer” is a generic name used by many to identify someone who creates or develops software.
3682 That person may be a mathematician, a physician, a civil engineer, or even a practitioner without any specific
3683 university degree or title. In English, the normal phrase refers to the people that occupy a job position. In other
3684 languages such as in Spanish, it is common to address people based on the title they received when they finished their
3685 undergraduate studies. As an example, in English there may be an announcement that “Company X seeks to hire three-
3686 thousand software engineers” to fulfill a large government contract. In this case, the reference is for people prepared
3687 to create or develop software, regardless the degree title they obtained. In this case, it would be a mistake to assume
3688 that Company X is looking for people with a specific software engineering university degree title.

3689
3690 In the case of licensure, this is the case where a government agency allows an individual the authority to do something
3691 or to practice a profession. For example, for the privilege to drive an automobile, almost all developed countries
3692 require a driver’s license. To practice medicine, dentistry, nursing, or law, a government agency requires a person to
3693 have a license to practice a profession. For professions such as medicine, this licensure occurs after a person completes
3694 formal studies and attains a degree such as a medical degree.

3695
3696 Some universities in different parts of the world issue a “licentiate” degree. It refers to a degree below a doctoral
3697 degree. Terms such as “licentia docendi” refer to permission or license to teach; the term “licentia ad practicandum”
3698 refers to someone having permission (license) for professional practice. The use of “license” can create much
3699 confusion in computing, and it is best to avoid the term except when issued for licensed practice such as a licensed
3700 professional engineer such as a licensed computer engineer or a licensed software engineer. Such professional
3701 licensure requires stringent legal regulations involving formal examinations, acquired university degrees, and years of
3702 professional practice.

3705 **7.5.2: Use of the Word “Engineer”**

3706
3707 The word “engineer” can have diverse meanings in different parts of the world. In some places it has a prestigious
3708 meaning and sometimes equated to the level of ‘doctor’ or other important professional. In other places, it is a just a
3709 normal expression used in a degree title or a job position. In some instances, universities unnecessarily force the word

3710 into degree titles to suggest an element of prestige, often as a scheme to attract students into their programs. For
3711 example, the use of “informatics engineering” or “systems engineering” may not have the program quality
3712 commensurate with its actual meaning. Hence, a university can rename a typical computer science program as an
3713 informatics engineering program would naturally attract more students because of a more appealing or more
3714 “prestigious” name. In Latin America, for example, programs having a degree name of “systems engineering” is
3715 largely due to corporate positions promoted by industry.

3716
3717 No less worrying is to observe places where the word “civil” is in common use. Specifically, “civil engineering”
3718 contrasts with “military engineering” as used in Chile. The word civil refers to people or engineering for the good of
3719 people. Hence a “computer civil engineer” is really a “computer engineer” rather than a civil or construction engineer
3720 who has a computing background. It is important to discourage such practices because the tradition causes confusion
3721 and unnecessary problems especially for international understanding and processes such as student exchange and
3722 accreditation.

3723
3724
3725

3726 **7.6: Technology Trends for CC2020**

3727

3728 This section addresses current and emerging technologies heavily dependent on computing and increasingly strongly
3729 integrated with a broad variety of types of human activity.

3730

3731

3732 **7.6.1: Current and Emerging Technologies**

3733

3734 Current and emerging technologies have a potential of affecting society very significantly such as the way in which
3735 we communicate or interact with each other, conduct commercial activities, organize decision making, and the way
3736 people learn. Some well-established technologies exist as independent areas of study within computing with their own
3737 curriculum recommendations (e.g., cybersecurity, data science and analytics) while others are barely out of research
3738 laboratories (e.g., DNA computing).

3739

3740 The curricula for “cybersecurity” and “data science and analytics” are already specialized areas of study closely
3741 affiliated with computing that either already have their own curriculum reports (CSEC 2017) or have one in
3742 preparation [Dat2]. Both are essential areas of focus by these two areas.

3743

3744 The technologies addressed reflect trends and reports by major global technology consultancies and the World
3745 Economic Forum (WEF). They include Accenture [Acc1], Deloitte [Del1], Gartner [Gar1], KPMG [Kpm1], and the
3746 WEF [Wef1].

3747

3748

3749 **7.6.2: Existing Computing Areas with No Formal Curriculum**

3750

3751 Three areas of computing-driven systems and technology infrastructure are already in existence, but they have not yet
3752 reached the status of an academic discipline with a globally recognized curriculum. Curricula may exist within a region
3753 or country, but not endorsed by recognized institutions such as ACM or IEEE.

3754

3755 *Internet of Things* (or IoT) refers to a “system of interconnections between digital technologies and physical objects
3756 that enable such (traditionally mundane) objects to exhibit computing properties and interact with one another with or
3757 without human intervention” [Bai1]. IoT technologies give physical objects capabilities that allow them to measure
3758 and communicate their states to other similar objects and provide centralized data collection mechanisms to enable
3759 coordinated data-driven action. Some countries such as China have robust IoT degree programs at the undergraduate
3760 and graduate levels.

3761

3762 *Cloud computing* refers to the practice of offering computing capabilities (particularly data storage and processing
3763 power) on the internet or other shared networks as a service, typically charged based on usage and managed by the
3764 service provider. Cloud computing is like a client-server system from decades ago with connectivity provided by the

3765 internet rather than a physical cable. Cloud computing essentially implements the idea of providing computing power
3766 and storage capabilities (infrastructure as a service), infrastructure integrated with platform services (platform as a
3767 service), or applications (system as a service) as a commodity service.
3768

3769 *Narrow artificial intelligence*, also known as weak AI, supports specific tasks in a narrow, well-defined context. Weak
3770 AI already exists on a broad variety of systems to enable and support human decision making. General artificial
3771 intelligence (strong AI) and artificial super intelligence (forms of AI that mimic and generally exceed human
3772 capabilities) do not currently exist. However, the area is already under fierce debate from ethical and moral
3773 perspectives.
3774
3775

3776 **7.6.3: Emerging Computing Areas**

3777 *Digital experience* refers to the practice of providing various organizational stakeholders (such as customers) a
3778 personalized and consistent set of experiences across a range of different digital platforms from small form factor
3779 wearable devices to large workstations and across a variety of situations. The terms used to describe this set of
3780 technologies include digital experiences (as used by Accenture to refer to augmented reality with 5G), multiexperience
3781 (as used by Gartner to refer to multiple channels for interacting with the digital world), and digital experience (as
3782 defined by Deloitte as human experience platforms), such as customized, emotionally intelligent digital experiences
3783 based on individuals' behaviors, preferences, and emotions using an integrated array of AI capabilities. Other
3784 innovative computing areas include distributed ledger technology, artificial intelligence, extended reality, and
3785 quantum computing (DARQ) as Accenture's "key set of new tech" as well as digital reality (identified by Deloitte as
3786 one of its macro forces).
3787
3788

3789 In addition, the area of *interaction technologies* is quickly moving from the traditional forms of point/click/swipe
3790 interfaces to those that most users will find more natural (such as speaking and gesturing, in the future potentially also
3791 thinking). Many of these technologies integrate with other capabilities that allow augmentation of the human
3792 experience with capabilities that naturally would not exist, often referred to with terms such as augmented reality,
3793 virtual reality, or mixed reality.
3794

3795 *Ambient computing* refers to contexts where the interaction experience between humans and technology has a tight
3796 integration with natural human experience that the technology as a separate entity becomes invisible. Various ways to
3797 describe this phenomenon include human augmentation (by Gartner) and digital reality as used by Deloitte to refer to
3798 augmented reality/virtual reality (AR/VR), mixed reality, voice interfaces, speech recognition, ambient computing,
3799 360° video, and immersive technologies. Other technologies include ambient experience as described by Deloitte as
3800 input evolving from unnatural to natural (e.g., speaking, gesturing, and thinking) and the interactions between humans
3801 and technology moving from reactive (e.g., answering questions) to proactive (e.g., making unanticipated suggestions)
3802 as well as wearables, identified as major trends by KPMG and by WEF.
3803

3804 The area of *cognitive technologies* is a label frequently used to refer to a variety of artificial intelligence capabilities
3805 for addressing complex organizational and societal problems. For example, Deloitte specifies categories of these
3806 capabilities to include robotic process automation, textual and auditory natural language processing, machine learning,
3807 and computer vision. Other articulations of these categories include "AI and me" by Accenture and hyperautomation
3808 supported by AI and machine learning by Gartner. Other related technologies include cognitive technologies,
3809 consisting of machine learning, neural networks, robotic process automation, bots, natural language processing, neural
3810 nets, and the broader domain of AI by Deloitte, as well as artificial Intelligence as part of DARQ technologies by
3811 Accenture.
3812

3813 *Blockchain or Distributed Ledger* refers to a set of technologies that allows set of actors to maintain a distributed
3814 record of transactions in a shared data storage environment in a way that is verifiable and permanent. Blockchain
3815 became first well-known as the technology underlying cryptocurrencies, but its potential areas of usage have expanded
3816 to financial services, management of contracts, health records, supply chain logistics, educational achievements, and
3817 many more. Sample reports refer to distributed ledger technologies with various names such as one of the DARQ
3818 technologies by Accenture, practical blockchain by Gartner, blockchain as a distributed ledger technology by Deloitte
3819 macro force, and distributed ledger by the WEF.
3820

3821 The area of *robotics* brings together a broad range of areas of expertise to create non-human artifacts (both physical
3822 and intangible) to perform a variety of tasks in an increasing rich set of contexts. The best-known contexts for robotics
3823 are probably in manufacturing, but the advances have been very rapid recently in many other contexts, including
3824 warehouses, medical work, military operations, and even business processes. The sample reports refer to robotics with
3825 expressions such as broad expansion of context for the use of robotics by Accenture, autonomous things by Gartner,
3826 and symbiotic robots by Info Tech.

3827
3828 *Quantum computing* incorporates a broad range of activities across a broad range of academic disciplines and industry
3829 research laboratories towards a new type of computing model. The process harnesses quantum phenomena at the level
3830 of subatomic particles to solve complex problems at a scale that would not be possible with traditional computing
3831 models. Deloitte has selected quantum computers as one of its macro forces and defines its core contribution as its
3832 ability to solve certain highly complex problems that are too large and messy for current supercomputers in a broad
3833 range of areas from data to material sciences.

3834
3835 While *data privacy and digital ethics* are not technologies per say, it is important to note that each new generation of
3836 digital technologies, and all mentioned in this subsection, raise important questions about the relationship between
3837 humans and technology. Privacy is often the first context for these questions, but the range of questions is, in practice,
3838 much broader [Mar4]. Transparency and traceability by Gartner and data equity by Info Tech are two aspects in
3839 support of privacy and ethics.

3840
3841
3842

3843 **7.7: Digest of Chapter 7**

3844

3845 This chapter addresses some of the challenges faced by institutions in adapting their computing curricula to the
3846 current environment. It emphasizes the move from knowledge-based teaching to competency-based learning,
3847 accepting that this move will need to be managed differently in different educational contexts. It makes clear the
3848 need for universities to engage with industry in adapting their curricula and outlines a number of ways in which this
3849 engagement might take place. It explains what institutions must do to maintain currency in their programs and
3850 suggests some ways in which they might deal with the ever-changing needs and expectations of the people for
3851 whom the degree programs are designed.

3852
3853
3854

3855

3856 **Chapter 8: Beyond CC2020**

3857

3858 CC2020 has surveyed the computing discipline and provided a structural view of computing that incorporates
3859 several sub-disciplines, including some that are recently emergent. This view of computing is based on programs
3860 that exist in 2020, which are now starting to include cybersecurity and data science. However, the important
3861 contribution here is not a definition of the discipline as it currently stands, but the establishment of a foundation for
3862 curricular specification that is based on competencies. This competency-based view of computing has not yet been
3863 developed, but a direction has been defined whereby competencies will be commonly used in the future. The pivot
3864 toward competencies in future computing curricular work is the most important contribution of this work. For this
3865 pivot to have impact, dissemination of these ideas is going to be critical. Aspects of this dissemination and pivot
3866 toward competencies are discussed below.

3867

3868

3869

3870 **8.1: Public Engagement and the CC2020 Project**

3871

3872 It is important that the efforts of the CC2020 project be available to the public worldwide. For the CC2020 project to
3873 be a success, it must engage the public. One means for accomplishing this goal is through an interactive website.
3874 Another is through a vigorous dissemination program sponsored by professional organizations, industry, and
3875 government.

3876

3877

3878 **8.1.1: CC2020 Project Website**

3879

3880 The CC2020 project has established a preliminary website [Ccw1] where the public can obtain information regarding
3881 the project. Such information includes a current copy of this CC2020 report, information about the structure of the
3882 CC2020 project, samples of competencies and visualizations, and other accompanying material.

3883

3884 An important addition to the project website will be its ability for students, parents, industry and government
3885 professionals, and faculty members to have a dynamic interaction with the project website. This includes comparisons
3886 of programs in different computing disciplines, comparison of programs within the same computing discipline,
3887 contrasting competencies, and other interesting activities. This dynamic dimension of the project is and will be a work
3888 in progress beyond the publication of the CC2020 report.

3889

3890

3891 **8.1.2: Relating Curricula and Competencies**

3892

3893 As noted in earlier chapters of this report, the notion of *competency* is the distinguishing feature of the CC2020 project,
3894 in contrast with the CC2005 project that focused on knowledge and knowledge-based learning. It is important for
3895 future curricular activities and development of curricular reports to embrace the use of competency in their work. The
3896 task force is aware that such future work will require greater effort in reaching a proper balance of dispositions, skills,
3897 and knowledge. However, since most (perhaps 99%) of graduates from computing programs will enter industry,
3898 government, or other workplace institutions, it behooves all curricula developers to embrace the competency-based
3899 approach.

3900

3901

3902 **8.1.3: Project Dissemination**

3903

3904 The CC2020 project requires dissemination on a global scale. Such an undertaking requires the support of professional
3905 organizations and societies as well as educational institutions to underwrite this effort, which should be an ongoing
3906 undertaking several years after the publication of this CC2020 report.

3907

3908 The dissemination activity should spur new interest in competency-based learning and curricular structures. The
3909 activity should generate new research for grant opportunities in achieving graduates who are competent to enter

3910 industry as well as being prepared for graduate or post-baccalaureate education. The CC2020 project should become
3911 a catalyst for these endeavors.

3912
3913
3914

3915 **8.2: Competency in Future Curricular Guidelines**

3916
3917
3918
3919
3920
3921
3922
3923

This report highlighted competency – one of the salient features of the CC2020 project. It also examined various
competency statements. Such statements may be useful in developing a uniform formalization of various disciplines.
As presented, the competency-based approach makes it possible to compare computing disciplines and facilitate those
comparisons. Recall that competency implies attaining a level of professional excellence and performance that goes
beyond having only knowledge in a field. These extensions include technical skills and human attributes to function
in the workplace at an acceptable level of performance. It is now important to extend the competency-based concept
toward the development of future curricular guidelines within a common frame of reference.

3924
3925

3926 **8.2.1: Recent Curricular Development**

3927
3928
3929
3930
3931

The CC2020 task force believes that the use of competency in current and future computing curricular reports should
be an important result of the CC2020 report. In today’s world, graduates must be able to perform in the workplace
with the technical skills and the human qualities in addition to knowledge.

3932
3933
3934
3935
3936

The cybersecurity project was an ongoing curricular effort that occurred in parallel with the CC2020 project. ACM
published this report, called CSEC2017 [Acm08], in December of 2017. The project used the traditional Ka-Ku-Lo
approach to develop its recommendations and its learning outcomes. Another effort parallel to the CC2020 project is
the ACM data science project. The leaders of this effort have made a commitment to adopt competency as an ongoing
theme and as a result of the project.

3937
3938

3939 **8.2.2: Future Curricular Development**

3940
3941
3942
3943
3944
3945
3946

Given that most graduates of computing programs enter the workplace, it is very important that all computing
programs prepare their graduates properly so they can perform as professionals and engage in productive careers.
While the CC2020 project can only suggest its beliefs, its task force is confident that computing organizations and
programs worldwide will heed the suggestions made in this report and transform their activities where competency
becomes central to their future undertakings.

3947
3948
3949
3950

The curricular recommendations for information systems are currently undergoing a revision with a planned
completion in 2020 or 2021. Naturally, other curricular updates include software engineering, computer science, and
computer engineering. The task force is hopeful that all future curricular endeavors adopt the competency-based
approach.

3951
3952
3953

3954 **8.3: Competency Advocacy**

3955
3956
3957
3958
3959
3960
3961
3962

The concept of using a common competency language to specify curricula, jobs, and careers provides an opportunity
to bring all computing education stakeholders under a common umbrella. For this effort to be successful, all the
stakeholders should reach consensus on the details of this language. For new curricular efforts that will emerge, authors
should develop techniques that will support the deployment of a uniform competency-based approach. Industry,
academia, and professional societies will need to develop these techniques together. The effort should incorporate a
community of interest to oversee this development.

3963
3964

It will be necessary for the computing professional societies to take the leadership needed to develop model curricular
standards using competencies. The computing professional societies should be part of any coalition of stakeholders,

3965 and the professional societies should mandate the use of competencies in developing future model curricula.
3966 Competencies should use a structure like the one advocated in this report, with both a prose statement and explicit
3967 knowledge, skills and dispositions components that can contribute analytically for visualization and comparison.
3968

3969 This project has planted the seeds for a high-quality public website that enables appropriate analysis of competency
3970 targets, and to provide career exploration and advice. This website should provide information about different types
3971 of computing careers as well as information on different types of degree programs that could prepare someone for a
3972 computing career. A variety of capabilities can be part of this site such as the ability to compare university programs
3973 in terms of their degree of similarity, and the degree of similarity between a program and a standard curriculum (such
3974 as the model curriculum for information systems), and the degree of similarity between an educational program and
3975 particular jobs and careers.
3976

3977 The efficacy of using competencies will occur as users gain experience with the approach. Once competency-based
3978 specifications of the various computing disciplines exist, then it would be possible to generate a comparative visual
3979 analysis among them – similar to the *ad hoc* visual representations of the CC2005 report, but with formal structural
3980 foundations.
3981

3982
3983

3984 **8.4: Future Activities**

3985

3986 The following summarize activities for curriculum-related developments that should take place over the next few
3987 years:

- 3988 ▪ Include the CC2020 report as an additional volume of the *Computing Curricula Series*.
- 3989 ▪ Establish new timetables for revision of each volume in the *Computing Curricula Series*.
- 3990 ▪ Ensure that a competency-based approach (rather than just knowledge-based learning) is part of all future
3991 computing curricular endeavors.
- 3992 ▪ Given the rapid pace of change in computing, consider more frequent revisions of computing curricular reports
3993 rather than the current approach, perhaps every six years instead of every ten to twelve years.
- 3994 ▪ Solicit improved support for more frequent updating of curricula.
- 3995 ▪ Continue processes for capturing feedback about each volume in the *Computing Curricula Series*.
- 3996 ▪ Establish new processes for ongoing evaluation of the adequacy of each curricular volume in the *Computing
3997 Curricula Series*.
- 3998 ▪ Improve current tools for visualizing computing programs.
- 3999 ▪ Develop new tools for visualizing computing programs.
- 4000 ▪ Become innovative in capturing new computing curricular areas to add to the *Computing Curricular Series*.

4001

4002 Consideration of the above projected activities should enhance computing education worldwide. The benefactors are
4003 the students who will enter those computing programs and the graduates of those programs who will find themselves
4004 as being competent professionals to enter the workplace or pursue further studies.
4005

4006

4007

4008

4009 **8.5: Digest of Chapter 8**

4010

4011 Computing and computing education is more important now than ever before. This chapter emphasizes the need for
4012 the global dissemination of this CC2020 report with the support of professional organizations and educational
4013 institutions. The project further advocates that all current and future computing curricula adopt a competency
4014 approach to better prepare the computing professionals of the future. Lastly, future activities for curriculum-related
4015 developments that should take place over the next few years are presented.
4016

4017

4018

4018
4019

Appendix A: Poster Explaining CC2005 Curricular Visuals

4020
4021
4022
4023

The Computing Field

Problem Space Scope

Computing—the goal-oriented activity that requires, benefits from, or creates computers—is a vibrant and challenging academic and professional field. The expansion and evolution of computing led to the specialization of knowledge and the emergence of several related, but quite different from each other, computing disciplines. In order to improve understanding of this family of disciplines by newcomers, but also among computing practitioners, the Association for Computing Machinery (ACM), the Association for Information Systems (AIS) and the Computer Society of the Institute for Electrical and Electronic Engineers (IEEE-CS) have sponsored a set of reports that point out the commonalities and differences between the computing disciplines. This poster provides a synthetic interpretation of those reports, highlighting the problem space scope, main knowledge areas and core performance capabilities of each of the five major computing disciplines: computer engineering, computer science, information systems, information technology, and software engineering.

**May 2014
Version 1.0**

Computer Engineering

CE

Computer Science

CS

Information Systems

IS

Information Technology

IT

Software Engineering

SE

Computer architecture & organization

Computer systems engineering

Digital logic

Programming fundamentals

Distributed systems

Circuits & systems

Electronics

Design and implement computing systems, computer-controlled equipment and communication software

Maintain computing systems that involve the integration of software and hardware devices

Software development fundamentals

Algorithms & complexity

Software engineering

Programming languages

Discrete structures

Systems fundamentals

Computer architecture & organization

Design & implement software problems

Optimize programming solutions

Prove theoretical results

Devise new ways to use computers

Foundations of information systems

Data & information management

Information systems strategy, management & acquisition

Enterprise architecture

Systems analysis & design

Information technology infrastructure

Project management

Improve organizational processes

Exploit technological innovations

Define information requirements

Design enterprise architecture

Secure data & infrastructure

Manage information systems risks

Technical support

Programming fundamentals

Information management

Information technology fundamentals

Systems integration

Mathematical fundamentals

Interpersonal communication

Train and support users

Plan, select, configure & maintain information systems infrastructure

Model, design, select, configure & manage databases

Configure & integrate business applications

Computing essentials

Software modeling & analysis

Software design

Software verification & validation

Professional practice

Mathematical & engineering fundamentals

Project management

Do small-scale & large-scale programming

Develop software systems

Manage software projects

Implement information systems

Define information systems technical requirements

Main Knowledge Areas

Core Performance Capabilities

Produced by
Filipe de Sá-Soares, PhD – fsa@isi.uminho.pt
Department of Information Systems
Centro ALGORITMI
School of Engineering
University of Minho
Guimarães, Portugal

Sources

CC (2006). Computing Curricula 2005 - The Overview Report. ACM, AIS and IEEE-CS.

CE (2004). Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering. IEEE-CS and ACM.

CS (2013). Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. ACM and IEEE.

IS (2010). Curriculum Guidelines for Undergraduate Degree Programs in Information Systems. ACM and AIS.

IT (2008). Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. ACM and IEEE-CS.

SE (2004). Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. IEEE-CS and ACM.

University of Minho
School of Engineering
CENTRO ALGORITMI

(Courtesy of Filipe de Sá-Soares, PhD - University of Minho, Portugal)

Appendix B: Computing Skills Frameworks

Different technology and informational skills presented in a report by the IEEE Computer Society [Cos1]. This *Guide to the Enterprise Information Technology Body of Knowledge* (EITBOK) [Ent1] is a compendium of high-level knowledge areas typically required for the successful delivery of computing services vital to all enterprises. EITBOK defines the key knowledge areas for the IT (computing) profession and it embodies concepts recognized as good practice in the IT domain and that are applicable to most IT efforts. The report emphasizes competence on a global scale. Frameworks enable the identification of skills and competencies required to perform duties and fulfill responsibilities in an enterprise IT workplace. Among the frameworks discussed are the Skills Framework for the Information Age (SFIA), the European Competency Framework (e-CF), and the i Competency Dictionary (iCD) of Japan. SFIA and e-CF had a major influence on the MSIS2016 report.

B.1: Skills Framework for the Information Age

Skills Framework for the Information Age (SFIA) has been used for approximately three decades and developed using a collaborative approach. The internationally represented SFIA Council oversees the direction of development for the not-for-profit SFIA Foundation, which owns and regularly updates the framework—using a well-established open process—for the benefit of the IT (computing) industry and IT professionals. The SFIA Framework is available in six languages (English, Spanish, German, Arabic, Japanese, and Chinese with more languages scheduled including French and French Canadian). It has been downloaded and used by organizations and individuals in nearly 180 countries. It can be downloaded for free at www.sfia-online.org.

The SFIA identifies 97 professional skills across IT and supporting areas and seven levels of responsibility as shown in Figure B.1. The seven levels in the SFIA are used to provide generic levels of responsibility and to reflect experience and competency. The Framework is based on demonstrated ability of applying a skill at a specific level, employing professional and behavioral skills as well as knowledge. The definitions describe the behaviors, values, knowledge and characteristics that an individual should have to be considered competent at a specific level. Underlying each SFIA Level are generic responsibilities of Autonomy, Complexity, Influence and Business Skills. These are described at each SFIA Level.

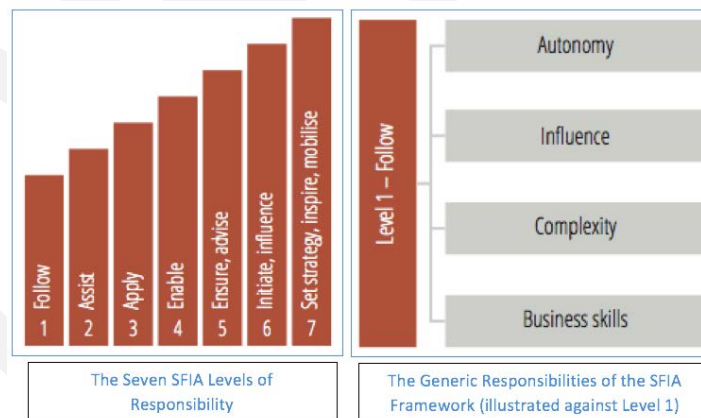


Figure B.1. SFIA Levels of Responsibility

The 97 IT skills of the Skills Framework are grouped into categories and sub-categories as in Table B.1. A skill has a name, a code, a skill description and a level description (for that skill at each level practiced).

Table B.1. SFIA Categories

Description
<p>The generic responsibility attributes delineated for all skills at all SFIA Levels:</p> <ul style="list-style-type: none"> • Autonomy • Complexity • Influence • Business Skills
The name of the skill
A description of what the skill is without reference to the levels practiced
<p>A description of the skill for each of the levels practiced, phrased to facilitate their use as professional competencies.</p> <ul style="list-style-type: none"> • Level 1 — Completes work tasks under close supervision. Seeks guidance in unexpected situations. Has an organized approach to work? Works with immediate colleagues only. • Level 2 — Uses some discretion to resolve issues or deal with enquiries. Works on a range of tasks, and proactively manages personal development. • Level 3 — Works under general direction only but has worked reviewed at regular intervals. Knows when to escalate problems / questions to a higher level. Works with suppliers and customers. May have some supervisory responsibility for less experienced staff. Performs a broad range of tasks, some complex. Plans schedules and monitors own work. • Level 4 — Has substantial personal responsibility and autonomy. Plans own work to meet objectives and execute end to end processes. Makes decisions which influence the success of projects and team objectives. Executes a broad range of complex technical or professional activities. • Level 5 — Broad direction, objective setting responsibility. Influences organization. Build effective working relationships. Performs Challenging and unpredictable work. Self-sufficient in business skills. Advises others on standards methods and tools. • Level 6 — Has authority for a significant area of work. Sets organizational objectives. Influences policy, customers, and suppliers at a senior level. Performs Highly complex and strategic work. Initiates and leads technical and organizational change. • Level 7 — At the highest organizational level, has authority over all aspects of a significant area of work, including policy formation and application. Makes decisions critical to organizational success. Inspires the organization, and influences developments within the industry at the highest levels. Develops long-term strategic relationships.

4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091

B.2: Skills and the European Competency Framework

The European e-Competence Framework (e-CF) from the European Union provides a reference of 40 competencies required for performance in the ICT workplace, using a common language for competencies, knowledge, skills, and proficiency levels that can be understood across Europe. The use of the e-CF by companies and organizations throughout Europe supports the transparency, mobility, and efficiency of ICT-sector-related human resources planning and development.

As the first sector-specific implementation of the European Qualifications Framework (EQF), the e-CF can be used by ICT service, demand and supply organizations, and by managers and human resources departments. Additionally, they are useful for educational institutions and training bodies, including higher education, professional associations, trade unions, market analysts and policy makers, and other organizations and parties in public and private sectors. The structure of the framework is based on four dimensions shown in Figure B.2.

There are five e-CF proficiency levels, e-1 to e-5, which relate to EQF learning levels 3 to 8. Table B.2 shows a description of the EQF levels [Eur3].

Dimension 1	Five e-Competence areas derived from the ICT business macro-processes PLAN – BUILD – RUN – ENABLE – MANAGE. The main aim of dimension 1 is to facilitate navigation through the framework.
Dimension 2	A set of reference e-Competences for each area, with a generic description for each competence. Forty competences identified in total provide the European generic reference definitions of the framework.
Dimension 3	Proficiency levels of each e-Competence provide European reference level specifications on e-Competence levels e-1 to e-5, which are related to EQF levels 3-8.
Dimension 4	Samples of knowledge and skills relate to e-Competences in dimension 2. They are provided to add value and context and are not intended to be exhaustive.
Figure B.2. Four dimensions of e-CF framework	

4092
4093

e-Competence Level	EQF Level
5 (highest)	8
4	7
3	6
2	4 and 5
1	3

4094
4095
4096
4097
4098
4099

As in SFIA, not all skills are subject to all five levels. Figure B.3 shows the spread of competency levels for each skill.

4100

Dimension 1 5 e-CF areas (A – E)	Dimension 2 40 e-Competences identified	Dimension 3 e-Competence proficiency levels e-1 to e-5, related to EQF levels 3–8				
		e-1	e-2	e-3	e-4	e-5
A. PLAN	A.1. IS and Business Strategy Alignment					
	A.2. Service Level Management					
	A.3. Business Plan Development					
	A.4. Product/Service Planning					
	A.5. Architecture Design					
	A.6. Application Design					
	A.7. Technology Trend Monitoring					
	A.8. Sustainable Development					
	A.9. Innovating					
B. BUILD	B.1. Application Development					
	B.2. Component Integration					
	B.3. Testing					
	B.4. Solution Deployment					
	B.5. Documentation Production					
	B.6. Systems Engineering					
C. RUN	C.1. User Support					
	C.2. Change Support					
	C.3. Service Delivery					
	C.4. Problem Management					
D. ENABLE	D.1. Information Security Strategy Development					
	D.2. ICT Quality Strategy Development					
	D.3. Education and Training Provision					
	D.4. Purchasing					
	D.5. Sales Proposal Development					
	D.6. Channel Management					
	D.7. Sales Management					
	D.8. Contract Management					
	D.9. Personnel Development					
	D.10. Information and Knowledge Management					
	D.11. Needs Identification					
	D.12. Digital Marketing					
E. MANAGE	E.1. Forecast Development					
	E.2. Project and Portfolio Management					
	E.3. Risk Management					
	E.4. Relationship Management					
	E.5. Process Improvement					
	E.6. ICT Quality Management					
	E.7. Business Change Management					
	E.8. Information Security Management					
	E.9. IS Governance					

Figure B.3. The European Competency Framework Overview

4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118

B.3: Skills and the i Competency Dictionary

The i Competency Dictionary (iCD) was developed and is maintained by the Information Technology Promotion Agency (IPA) in Japan. It consists of a comprehensive Task Dictionary and a corresponding Skill Dictionary. The Task Dictionary contains all the tasks that EIT outsourcers or EIT departments are expected to accomplish, while the corresponding Skill Dictionary provides the skills required to perform those tasks.

The diagrams in Figure B.4 through Figure B.9 show how the task and skill dictionaries are structured to be used together. The skills needed to become competent at each task are enumerated in a Task vs. Skill table. The diagrams indicate the number of tasks and skills that are included in the full iCD. It is possible to obtain the complete iCD Task Dictionary (Layers 1-4) and Skill Dictionary (Layers 1-4) from the IPA website [Ipa1].

4119
4120
4121
4122
4123
4124
4125
4126
4127

B.3.1: Task Dictionary

The Task Dictionary is intended to be used and applied by companies and organizations to determine tasks in line with their organizational strategies or organization plans. Tasks are used to define their organizational functions and the roles of personnel. The structure of the dictionary assumes a wide range of corporate activities, so that companies with any kind of business model can use and apply it. The Task Dictionary is comprised of four layers divided into three task layers plus the Task Evaluation Items layer, shown in Figure B.4.

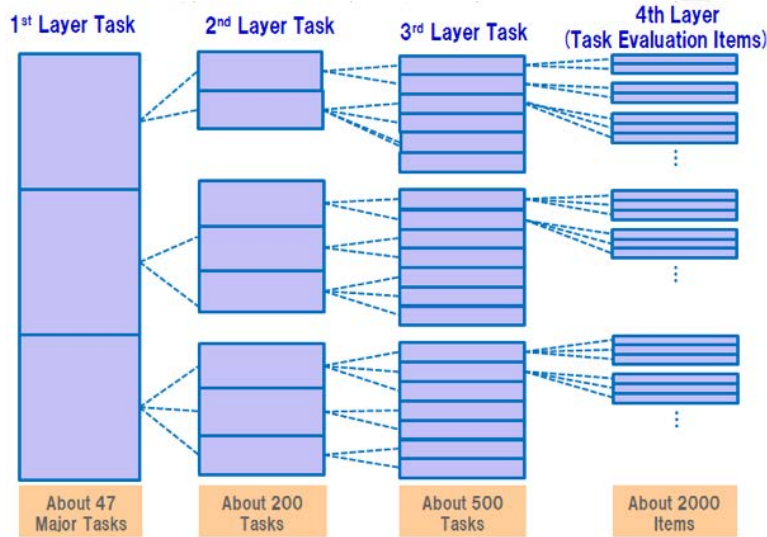


Figure B.4. The iCD Task Dictionary Structure

4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140

B.3.2: Task Dictionary Chart

The Task Dictionary Chart (Figure B.5) can be used to obtain a bird’s-eye view of the entire Task Dictionary on the 1st Layer Task level. This chart presents a task structure composed of the organization lifecycle as vertical line (Strategy, Planning, Development, Utilization, Evaluation & Improvement) and tasks associated with entire lifecycle as horizontal line (Planning & Execution, Management & Control, Promotion & Support).

4161
4162
4163
4164
4165
4166
4167
4168
4169
4170

B.3.4: Skill Dictionary

Skills are capabilities required to handle associated knowledge items to execute a task. The Skill Dictionary is comprised of four layers divided into three skill layers plus Associated Knowledge Items, shown in Figure B.7. The Skill Dictionary refers and sorts the items from the major Body of Knowledges/processes and skill standards in the world.

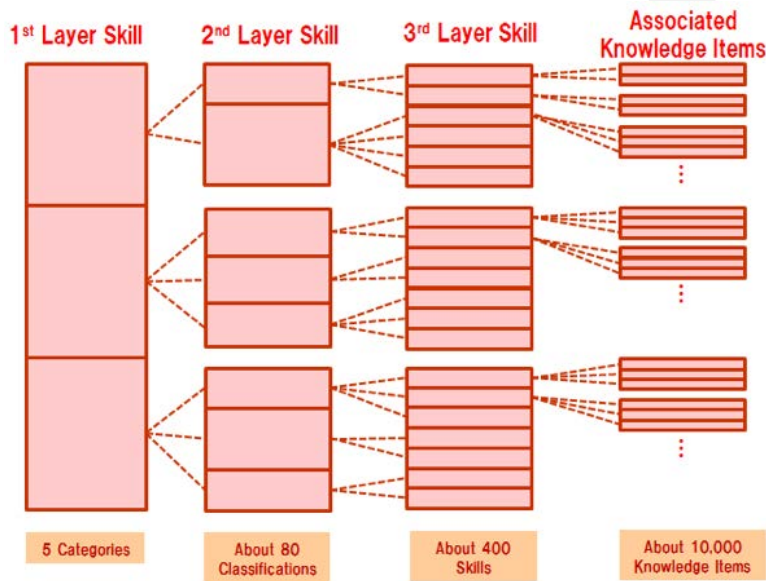


Figure B.7. The iCD Skill Dictionary Structure

4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185

B.3.5: Skill Dictionary Chart

The Skill Dictionary Chart (Figure B.8) can be used to obtain a bird's-eye view of the entire Skill Dictionary on the 1st and 2nd skill layers. The Skill Dictionary is divided into five categories based on the skill characteristics: methodology, technology, related knowledge, IT (human) skills, and specific skill (optional). This chart represents a skill structure on the perspectives of the IT orientation (Horizontal line: High-Low) and the application area (Vertical line: Wide-Narrow).

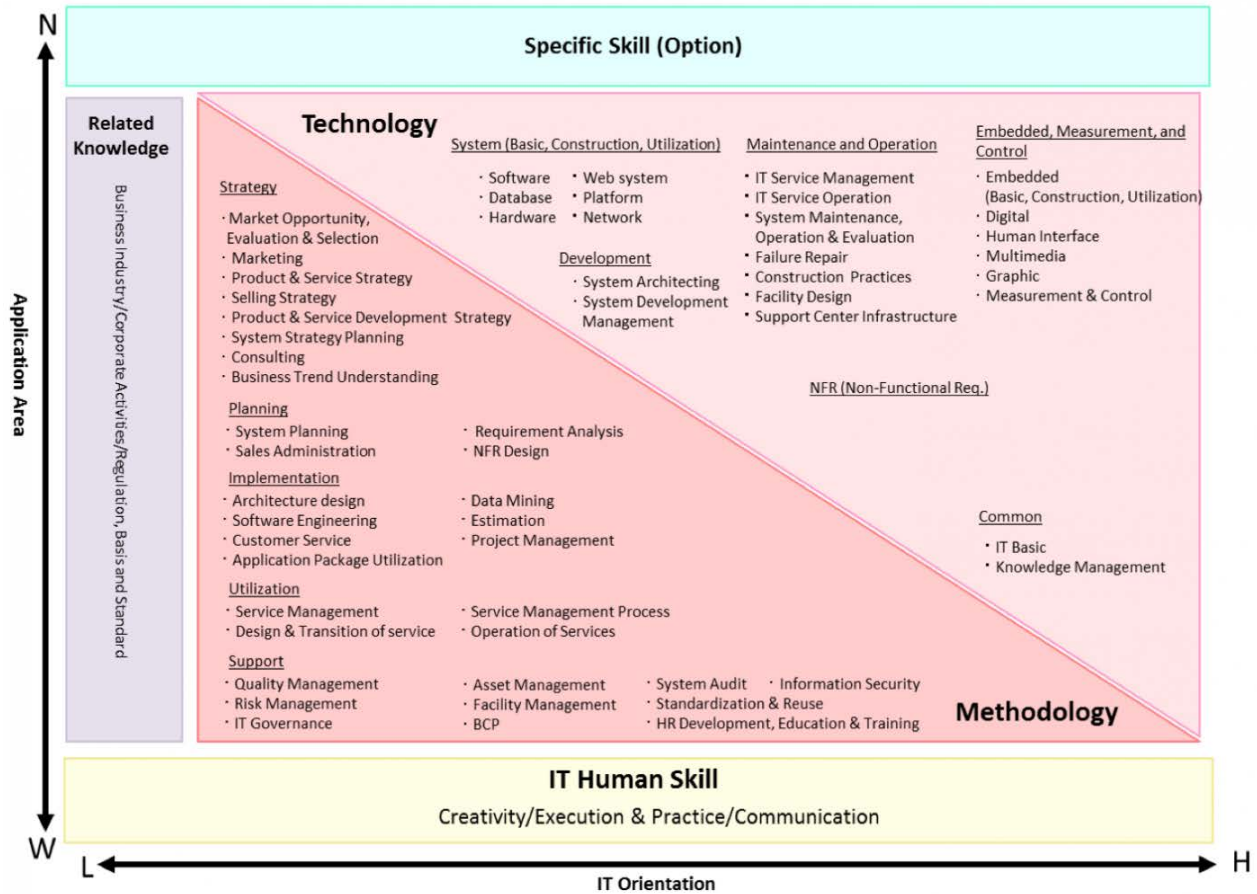


Figure B.8. The iCD Skill Dictionary Chart

B.3.6: Skill Proficiency Level

The chart in Figure B.9 measures the skill proficiency level using seven levels of skill proficiency criteria. Level 1 to 4 criteria differ according to contents of technology/methodology/related knowledge. Skill proficiency level 4 is the highest acquisition level of the skill for the task accomplishment. Level 5 to 7 criteria are defined across the categories to evaluate by social contribution degree as a professional.

Level 7	Skills at the level of an industry leader who has influence on the market		
Level 6	Skills at the level of a recognized contributor to the industry		
Level 5	Skills at the level of a recognized contributor within affiliated associations and organizations		
Level 4	Level at which one is able to produce optimal solutions that take into account non-functional requirements, step outside of established tactics, and pass the advanced information technology examinations	Has mastered and can select the most suitable methods, and can freely apply the methods according to the situation	Is able to discuss what needs to be done with senior management within the industry or business they are involved in
Level 3	Is able to create functional requirements and to work independently under limited circumstances	Is able to apply the proper method according to the problem, and has utilized the methods on-site and drawn conclusions	Has proposed solutions to the IT-related problem points in the industry and businesses they are involved in
Level 2	Has implementation experience, and is able to use and apply the technology if instructions are available	Is able to perform analysis using the method, or is able to use the methodology under guidance	Understands the IT-related problem points in the industry and businesses they are involved in
Level 1	Has knowledge, and understands lectures and presentations of technical content	Understands lectures and presentations about the method, understands and can explain what it is, and understands textbooks about it	Understands and can explain what kind of industry and business they are involved in, and understands public information such as securities reports
Category	Technology	Methodology	Related Knowledge

Figure B.9. Skill Proficiency Level

4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233

B.4: Skills via Enterprise Information Technology

The emphasis on competence has become international as Enterprise IT (EIT) and ICT in general have become indispensable across the globe. EIT and ICT derive from a growing understanding of the need for a common language for competencies, knowledge, skills and proficiency levels that can be understood across national borders. A common framework enables the identification of skills and competencies that may be required to successfully perform duties and fulfill responsibilities in an EIT workplace. They provide a common basis for the selection and recruitment of EIT staff, as well as forming the basis for employment agreements, professional development plans, and performance evaluation for ICT professionals.

Many national and regional governments have come to require certification of EIT practitioners. Accordingly, they have had to develop their own definitions of ICT competencies. Given the increasingly international composition of the EIT workforce, the EITBOK has included information from three major frameworks that are emerging as inter-regional. In general, these frameworks work towards a common understanding of competence, defined by the e-CF, for example, as “demonstrated ability to apply knowledge, skills and attitudes to achieve observable results.”

B.5: References for Appendix B

[Cos1] Computer Society of IEEE. <http://www.computer.org/>. Accessed 2019 May 6.
 [Ent1] Enterprise Information Technology Body of Knowledge (EITBOK). http://eitbokwiki.org/Main_Page. Accessed 2019Aug17.
 [Ipa1] Information-technology Promotion Agency – Japan (IPA). IT Human Resources Development: i Competency Dictionary. <https://www.ipa.go.jp/english/humandev/icd.html>. Accessed 2019Aug18.
 [Eur3] European Qualifications Framework (EQF). European Commission. Learning Opportunities and Qualifications. <https://ec.europa.eu/ploteus/en/content/descriptors-page/>. Accessed 2019Aug17.

4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268

Appendix C: Examples of Competencies

At the onset of the CC2020 project, subgroups of task force experts explored the development of competency statements for different computing disciplines. To accomplish this task, a process to generate competencies and then provide a framework for modeling competencies associated with each computing discipline was employed. The IT2017 report already specified competencies by abandoning the Ka-Ku-Lo approach and eliminating learning outcomes and topics. It published these competencies within domain clusters. The report specified competencies that were essential and supplemental.

C.1: Initial CC2020 Explorations of Competencies

For each of the established computing reports (IS2010, CS2013, SE2014, CE2016), the respective project teams had used Ka-Ku-Lo strategies rather than competency approaches. The combined output resulted in thousands of learning outcomes derived from these published reports. Members of the CC2020 task force then transformed the learning outcomes from four curricular reports into competencies. This activity was not easy because of the novelty of the meaning of competency and its use in computing educational circles. Notwithstanding, the CC2020 steering committee created a focus group and partitioned this group into four subgroups, each identified with one of the four computing disciplines mentioned above.

Each subgroup used the canonical definition of *Competency = Knowledge + Skills + Dispositions*. Over a period of six months, the subgroups prototyped competencies for their respective computing disciplines. The number of statements for each discipline varied, although three-dozen was the target number. The IT2017 report already had stated competencies so it was not part of this focus group.

The following subsections describe in part the results of the work from the competency groups for computer engineering, computer science, information systems, information technology, and software engineering. It also includes generated competencies from a Master's in information systems (MSIS2016) report.

4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286

C.1.1: Computer Engineering Draft Competencies

CC2020
Computer Engineering (24) Competencies
Derived from CE2016
2018 May 3

Notice

The material that follows are two versions of the same CE competencies. The CE subgroup had several discussions on whether it should include the dimension of “disposition” as a self-contained statement or embed dispositions within each competency statement. The left column shows the former version with (human) disposition in Item B. The right column shows the latter version with embedded dispositions. The CE task force is neutral on which is the preferred representation.

Self-contained Disposition Version	Embedded Disposition Version
<p>For each Knowledge Area:</p> <p>A. Communicate the essential elements of the history of computer engineering, including the development of tools, standards, and constraints to a technical audience. <i>[History & overview; relevant tools, standards, constraints]</i></p> <p>B. Exercise all CE competencies in a contextually appropriate manner, demonstrating proper consideration of ethics, cultures, background, and human relationships. <i>[Dispositions - the human element]</i></p> <p>CE-CAE</p> <p>1. Analyze and design circuits using electronic devices, and innovate in the context of new and existing systems using those components to create new functions on varying levels of complexity bearing in mind the tradeoffs involved. <i>[History & Overview; Tools & standards; electrical quantities, elements & circuits; electronic materials & devices; MOS transistors; data storage cells]</i></p> <p>CE-CAL</p> <p>1. Design and/or implement classic and application-specific algorithms including parallel in multi-threading ones by relevant</p>	<p>For each Knowledge Area:</p> <p>A. Communicate the essential elements of the history of computer engineering, including the development of tools, standards, and constraints to a technical audience. <i>[History & overview; relevant tools, standards, constraints]</i></p> <p>CE-CAE</p> <p>1. Analyze and design circuits for a local engineering company using electronic devices and innovate in the context of new and existing systems using those components to create new functions on varying levels of complexity bearing in mind the tradeoffs involved. <i>[History & Overview; Tools & standards; electrical quantities, elements & circuits; electronic materials & devices; MOS transistors; data storage cells]</i></p> <p>CE-CAL</p> <p>1. Design and/or implement classic and application-specific algorithms including parallel in multi-threading ones by relevant</p>

Self-contained Disposition Version	Embedded Disposition Version
<p>tools within engineering, marketing, commercial or legal constraints in the respectful and meaningful interaction with users and customers. <i>[Relevant tools; algorithms - common ones, analysis, strategies]</i></p> <p>2. Analyze correctness, efficiency, performance and complexity of the algorithms using order of complexity terms and present honestly and comprehensively the results of the analysis for either a professional or non-professional audience. <i>[Algorithmic complexity; scheduling algorithms; computability theory]</i></p>	<p>tools within engineering, marketing, commercial or legal constraints in the respectful and meaningful interaction with users and customers. <i>[Relevant tools; algorithms - common ones, analysis, strategies]</i></p> <p>2. Analyze correctness, efficiency, performance and complexity of the algorithms using order of complexity terms and present honestly and comprehensively the results of the analysis for either a professional or non-professional audience. <i>[Algorithmic complexity; scheduling algorithms; computability theory]</i></p>
<p>CE-CAO</p> <p>1. Manage the design of computer hardware components and integrate such components to provide complete hardware systems which function reliably and efficiently demonstrating sensitivity for the context of the design envelope within which they were conceived. <i>[Measuring performance; Processor organization; Distributed systems architecture; Multi/Many-core architectures; Peripheral subsystems]</i></p> <p>2. Simulate and evaluate the performance of parallel and sequential hardware solutions and tradeoffs involved in designing complex hardware systems considering the design of memory and arithmetical units as well as characterizing system performance using appropriate metrics. <i>[Processor organization; Memory system organization & architecture; Computer arithmetic; Input/Output interfacing and communication]</i></p>	<p>CE-CAO</p> <p>1. Manage the design of computer hardware components for a multidisciplinary research project and integrate such components to provide complete hardware systems which function reliably and efficiently demonstrating sensitivity for the context of the design envelope within which they were conceived. <i>[Measuring performance; Processor organization; Distributed systems architecture; Multi/Many-core architectures; Peripheral subsystems]</i></p> <p>2. Present a report that discusses the simulation and evaluation of the performance of parallel and sequential hardware solutions and tradeoffs involved in designing complex hardware systems considering the design of memory and arithmetical units as well as characterizing system performance using appropriate metrics. <i>[Processor organization; Memory system organization & architecture; Computer arithmetic; Input/Output interfacing and communication]</i></p>
<p>CE-DIG</p> <p>1. Using appropriate tools, design digital circuits including the basic building blocks of Boolean algebra, computer numbering systems, data encoding, combinatorial and sequential elements. <i>[Tools & standards; numbering systems & data encoding; Boolean algebra; digital logic, combinatorial & sequential]</i></p>	<p>CE-DIG</p> <p>1. Manage the design of a computer system for a manufacturer using appropriate tools, design digital circuits including the basic building blocks of Boolean algebra, computer numbering systems, data encoding, combinatorial and sequential elements. <i>[Tools & standards; numbering systems & data encoding; Boolean algebra; digital logic, combinatorial & sequential]</i></p>

Self-contained Disposition Version	Embedded Disposition Version
<p>2. Design a control or datapath circuit using programmable logic and considering relevant system design constraints and testability concerns. <i>[Control & datapaths; programmable logic; system constraints; fault models & testing]</i></p>	<p>2. Design a control or datapath circuit for a small company using programmable logic and considering relevant system design constraints and testability concerns. <i>[Control & datapaths; programmable logic; system constraints; fault models & testing]</i></p>
<p>CE-ESY</p> <p>1. Design and/or implement basic and advanced I/O techniques, both synchronous and asynchronous and serial/parallel, including interrupts and time considerations. <i>[Parallel/ serial I/O; synchronous/asynchronous I/O; interrupts and timing]</i></p> <p>2. Design and implement an example of an embedded system in a non-electronic device, including sensor feedback, low-power, and mobility. <i>[Data acquisition & sensors; embedded systems characteristics; low-power operation]</i></p>	<p>CE-ESY</p> <p>1. Present to a group of peers the design and implementation of basic and advanced I/O techniques, both synchronous and asynchronous and serial/parallel, including interrupts and time considerations. <i>[Parallel/ serial I/O; synchronous/asynchronous I/O; interrupts and timing]</i></p> <p>2. Design and implement for a professional seminar an example of an embedded system in a non-electronic device, including sensor feedback, low-power, and mobility. <i>[Data acquisition & sensors; embedded systems characteristics; low-power operation]</i></p>
<p>CE-NWK</p> <p>1. Develop, deploy, maintain and evaluate the performance of wireless and wired networking solutions in the context of relevant standards and the needs of stakeholder groups and demonstrating awareness of the foundations and history of the area. <i>[History and overview; Relevant tools, standards]</i></p> <p>2. Relate general networking competence to integrated solutions in the Internet of Things considering security and privacy aspects and the impact of solutions on citizens and society. <i>[Network architecture; Local and wide-area networks; Network protocols; Network applications; Network management; Data communications; Performance evaluation; Wireless sensor networks]</i></p>	<p>CE-NWK</p> <p>1. Develop, deploy, maintain and evaluate the performance of wireless and wired networking solutions for a manufacturer in the context of relevant standards and the needs of stakeholder groups and demonstrating awareness of the foundations and history of the area. <i>[History and overview; Relevant tools, standards]</i></p> <p>2. Relate general networking competence to integrated solutions in the Internet of Things considering security and privacy aspects and the impact of solutions on citizens and society. <i>[Network architecture; Local and wide-area networks; Network protocols; Network applications; Network management; Data communications; Performance evaluation; Wireless sensor networks]</i></p>
<p>CE-PPP</p> <p>1. Analyze the importance of communication skills in a team environment and within a</p>	<p>CE-PPP</p> <p>1. Analyze the importance of communication skills in a team environment and within a</p>

Self-contained Disposition Version	Embedded Disposition Version
<p>computer engineering group setting, discuss and determine how these skills contribute to the optimization of organization goals. <i>[Communication and teamwork]</i></p> <p>2. Evaluate the philosophical and cultural attributes necessary for maintaining a global relationship in solving a computer engineering problem that involves a system development in a political context. <i>[Philosophical, cultural and societal issues]</i></p> <p>3. Develop hardware policies that include professional, legal, and ethical considerations as they relate to a global engineering company. <i>[Professional, ethical, and legal issues]</i></p> <p>4. Evaluate contemporary issues facing a computer engineering project and develop an effective project plan using business acumen and cost/benefit analysis. <i>[Contemporary, business, and management issues]</i></p>	<p>computer engineering group setting, discuss and determine how these skills contribute to the optimization of organization goals. <i>[Communication and teamwork]</i></p> <p>2. Evaluate the philosophical and cultural attributes necessary for maintaining a global relationship in solving a computer engineering problem that involves a system development in a political context. <i>[Philosophical, cultural and societal issues]</i></p> <p>3. Develop hardware policies that include professional, legal, and ethical considerations as they relate to a global engineering company. <i>[Professional, ethical, and legal issues]</i></p> <p>4. Evaluate contemporary issues facing a computer engineering project and develop an effective project plan using business acumen and cost/benefit analysis. <i>[Contemporary, business, and management issues]</i></p>
<p>CE-SEC</p> <p>1. Evaluate the current cybersecurity tools for their effectiveness in providing data security, side-channel attacks, and integrity while avoiding vulnerabilities, both technical and human-factor caused. <i>[Data security and integrity; Vulnerabilities; Network and web security; Side-channel attacks]</i></p> <p>2. Design a cybersecurity solution which provides resource protection, public and private key cryptography, authentication, network and web security, and trusted computing. <i>[Resource protection models; Secret and public key cryptography; Message authentication codes; Authentication; Trusted computing]</i></p>	<p>CE-SEC</p> <p>1. Write a report on the evaluation of the current cybersecurity tools for their effectiveness in providing data security, side-channel attacks, and integrity while avoiding vulnerabilities, both technical and human-factor caused. <i>[Data security and integrity; Vulnerabilities; Network and web security; Side-channel attacks]</i></p> <p>2. Design a cybersecurity solution for a network company that provides resource protection, public and private key cryptography, authentication, network and web security, and trusted computing. <i>[Resource protection models; Secret and public key cryptography; Message authentication codes; Authentication; Trusted computing]</i></p>
<p>CE-SGP</p> <p>1. Design signal processing systems applying knowledge of sampling and quantization to bridge the analog and digital domains. <i>[Transform analysis; frequency response; sampling & aliasing; spectra]</i></p>	<p>CE-SGP</p> <p>1. Design signal processing systems with an engineering team by applying knowledge of sampling and quantization to bridge the analog and digital domains. <i>[Transform analysis; frequency response; sampling & aliasing; spectra]</i></p>

Self-contained Disposition Version	Embedded Disposition Version
<p>2. Evaluate signal processing challenges (e.g., detection, denoising, interference removal) to support the selection and implementation of appropriate algorithmic solutions including non-recursive and recursive filters, time-frequency transformations, and window functions. <i>[Relevant tools, standards & constraints; convolution; window functions; multimedia processing; control systems]</i></p>	<p>2. Evaluate signal processing challenges (e.g., detection, denoising, interference removal) to support the selection and implementation of appropriate algorithmic solutions including non-recursive and recursive filters, time-frequency transformations, and window functions and present the results to an electrical engineering team. <i>[Relevant tools, standards & constraints; convolution; window functions; multimedia processing; control systems]</i></p>
<p>CE-SPE</p> <p>1. Manage a project that requires the analysis of a system (hardware and software), including system requirements, both technical (including functional and performance requirements) and in terms of suitability, usability and inclusiveness, taking an holistic perspective to crafting specifications and evaluating reliability. <i>[Project management principles; User experience; Risk, dependability, safety & fault tolerance; Requirements analysis and elicitation; Hardware and software processes; System specifications; System architecture design and evaluation; Concurrent hardware and software design; System integration, testing and validation; Maintainability, sustainability, manufacturability]</i></p>	<p>CE-SPE</p> <p>1. Manage a project for an organization that requires the analysis of a system (hardware and software), including system requirements, both technical (including functional and performance requirements) and in terms of suitability, usability and inclusiveness, taking an holistic perspective to crafting specifications and evaluating reliability. <i>[Project management principles; User experience; Risk, dependability, safety & fault tolerance; Requirements analysis and elicitation; Hardware and software processes; System specifications; System architecture design and evaluation; Concurrent hardware and software design; System integration, testing and validation; Maintainability, sustainability, manufacturability]</i></p>
<p>CE-SRM</p> <p>1. Analyze the role of single user, mobile, networked, client-server, distributed, and embedded operating systems, interrupts, and real-time support in managing system resources and interfacing between hardware and software elements considering economic, environmental and legal limitations. <i>[History and overview of operating systems, Managing system resources, Operating systems for mobile devices, Support for concurrent processing]</i></p> <p>2. Design and implement an appropriate performance monitoring procedure for standard and virtual systems. <i>[Real-time operating system design, System performance evaluation; Support for virtualization]</i></p>	<p>CE-SRM</p> <p>1. Analyze the role of single user, mobile, networked, client-server, distributed, and embedded operating systems, interrupts, and real-time support in managing system resources and interfacing between hardware and software elements considering economic, environmental and legal limitations. <i>[History and overview of operating systems, Managing system resources, Operating systems for mobile devices, Support for concurrent processing]</i></p> <p>2. Present to an organization the design and implementation of appropriate performance monitoring procedures for standard and virtual systems. <i>[Real-time operating system design, System performance evaluation; Support for virtualization]</i></p>
<p>CE-SWD</p>	<p>CE-SWD</p>

Self-contained Disposition Version	Embedded Disposition Version
<p>1. Evaluate and apply programming paradigms and languages to solve a wide variety of software design problems being mindful of trade-offs including maintainability, efficiency, and intellectual property constraints. <i>[Programming constructs & paradigms; problem solving; history & overview; relevant tools, standards, constraints]</i></p> <p>2. Design software tests for evaluating a wide variety of performance criteria on subsystems (including usability, correctness, graceful failure, and efficiency) within the context of a complete hardware-software system. <i>[Software testing & quality]</i></p>	<p>1. Write a report for a manufacturer regarding the evaluation and application of programming paradigms and languages to solve a wide variety of software design problems being mindful of trade-offs including maintainability, efficiency, and intellectual property constraints. <i>[Programming constructs & paradigms; problem solving; history & overview; relevant tools, standards, constraints]</i></p> <p>2. Design software testing procedures for an engineering team that evaluates a wide variety of performance criteria on subsystems (including usability, correctness, graceful failure, and efficiency) within the context of a complete hardware-software system. <i>[Software testing & quality]</i></p>

4287
4288
4289

Legend

CE-CAE	Circuits and Electronics
CE-CAL	Computing Algorithms
CE-CAO	Computer Architecture & Organization
CE-SPE	Systems and Project Engineering
CE-DIG	Digital Design
CE-ESY	Embedded Systems
CE-NWK	Computer Networks
CE-PPP	Preparation for Professional Practice
CE-SEC	Information Security
CE-SGP	Signal Processing
CE-SRM	Systems Resource Management
CE-SWD	Software Design

4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301

Task force Members on the CE Subgroup

Barry Lunt (Leader)
Olga Bogyavlenskaya
Eric Durant
John Impagliazzo
Arnold Neville Pears

4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356

C.1.2: Computer Science Draft Competencies

CC2020
Computer Science (84) Competencies
Derived from CS2013
2018 May 30

AL-Algorithms and Complexity

- A. Present to a group of peers the data characteristics of conditions or assumptions that can lead to different behaviors of specific algorithms and from the analysis, illustrate empirical studies to validate hypotheses about runtime measures.
- B. Illustrate informally the time and space complexity of algorithms and use big-O notation formally to show asymptotic upper bounds and expected case bounds on time and space complexity, respectively.
- C. Use recurrence relations to determine the time complexity of recursively defined algorithms by solve elementary recurrence relations and present the results to a group of scholars.
- D. Determine an appropriate algorithmic approach to an industry problem and use appropriate techniques (e.g., greedy approach, divide-and-conquer algorithm, recursive backtracking, dynamic programming or heuristic approach) that considers the trade-offs between the brute force to solve a problem.
- E. Implement basic numerical algorithm methods (e.g., search algorithms, common quadratic and $O(N \log N)$ sorting algorithms, fundamental graph algorithms, string-matching algorithm) to solve an industry problem and select the appreciate algorithm for a particular context.
- F. Design a deterministic finite state machine for a local engineering firm that accepts a specified language and generates a regular expression to represent the language.

AR-Architecture and Organization

- A. Use CAD tools for capture, synthesis, and simulation to evaluate simple building blocks
- B. of a simple computer design for a local engineering company.
- C. Evaluate the timing diagram behavior of a simple processor implemented at the logic circuit level and develop a report expressing the findings.
- D. Write a simple program at the assembly/machine level for string processing and manipulation and for converting numerical data into hexadecimal form.
- E. Implement a fundamental high-level construct in both machine and assembly languages and present the results to a group of peers.
- F. Calculate the average memory access time under a variety of cache and memory configurations and develop a short report of the findings.

CN-Computational Science

- A. Create a simple, formal mathematical model of a real-world situation and use that model in a simulation for a local technology company.

DS-Discrete Structures

- A. Present to a peer group some practical examples of an appropriate set, function, or relation model, and interpret the associated operations and terminology in context.
- B. Use symbolic propositional and predicate logic to model a real-life industry application by applying formal methods (e.g., calculating validity of formulae and computing normal forms to the symbolic logic).
- C. Apply rules of inference to construct proofs and present results to a group of professionals, appropriate proofs or logical reasoning to solve a strategic problem.
- D. Map real-world applications to appropriate counting formalisms and apply basic counting theories (e.g., counting arguments, the pigeonhole principle, modular arithmetic as well as compute permutations and combinations of a set) to solve an industry problem.
- E. Analyze an industry problem to determine underlying recurrence relations and present the solution to professionals by using a variety of basic recurrence relations.

- 4357 F. Model a real-world problem using appropriate graphing strategies (e.g., trees, traversal methods for graphs and
4358 trees, spanning trees of graph) and determine whether two graph approaches are isomorphic.
4359 G. Calculate different probabilities of dependent or independent events and expectations of random variables to solve
4360 a problem and present to a group of peers the ways to compute the variance for a given probability distribution.
4361
4362

GV-Graphics and Visualization

- 4364 A. Design and develop a user interface using a standard API and that incorporates visual and audio techniques usable
4365 for a local organization
4366

HCI-Human-Computer Interaction

- 4368 A. Design an interactive application, applying a user-centered design cycle with related tools and techniques (modes,
4369 navigation, visual design), to optimize usability and user experience within a corporate environment.
4370 B. Analyze and evaluate a user interface that considers context of use, stakeholder needs, state-of-the-art response
4371 interaction times, design modalities taking into consideration universal access, inclusiveness, assistive
4372 technologies, and culture sensitive design.
4373 C. Design and develop an interactive application for a local charity, applying a user-centered design cycle with
4374 related vocabulary, tools and techniques that optimizes usability and user experience.
4375 D. Create and conduct a simple usability test to analyze and evaluate a user interface that considers context of use
4376 taking into consideration universal access and culturally sensitive design.
4377 E. Create a simple application, together with help and documentation, that supports a graphical user interface for an
4378 enterprise and conduct a quantitative evaluation and report the results.
4379

IAS-Information Assurance and Security

- 4380
4381 A. Write correct input validation code for a cybersecurity company after classifying common input validation errors.
4382 B. Demonstrate to a group of security professionals ways to prevent a race condition from occurring and ways to
4383 handle exceptions.
4384

IM-Information Management

- 4385
4386 A. Contrast information with data and knowledge and describe to a group of professionals the advantages and
4387 disadvantages of centralized data control.
4388 B. Demonstrate to a group of peers a declarative query language to elicit information from a database.
4389 C. Contrast appropriate data models, including internal structures, for different types of data and present an application
4390 to a group of professionals the use of modeling concepts and notation of the relational data model.
4391

IS-Intelligent Systems

- 4392
4393 A. Determine the characteristics of a given problem that an intelligent system must solve and present the results to a
4394 project team.
4395 B. Formulate an industry problem specified in a natural language (e.g., English) as a constraint satisfaction problem
4396 and implement it using an appropriate technique (e.g., chronological backtracking algorithm or stochastic local
4397 search).
4398 C. Implement an appropriate uninformed or informed search algorithm for an industry problem by characterizing
4399 time and space complexities of informed algorithm or designing the necessary heuristic evaluation function for
4400 uninformed search algorithm to guarantee an optimal solution, respectively.
4401 D. Translate a natural language (e.g., English) sentence for a corporate query system into a predicate logic statement
4402 by converting a logic statement into clause form and applying resolution to a set of logic statements to answer a
4403 query.
4404 E. Make a probabilistic inference in a real-world industry problem using Bayes' theorem to determine the probability
4405 of a hypothesis given evidence.
4406

NC-Networking and Communication

- 4407
4408 A. Design and develop for a corporate customer a simple client-server socket-based application.
4409 B. Design and implement a simple reliable protocol for an industry network by considering factors that affect the
4410 network's performance.
4411 C. Contrast fixed and dynamic allocation techniques as well as current approaches to congestion and present the
4412 results to company executives.

4413

4414 **OS-Operating Systems**

- 4415 A. Apply knowledge of computing theory and mathematics to solve problems and present comprehensively the results
4416 and methods of the solution for either a professional or non-professional audience.
4417 B. Implement software solutions within system constraints of a target system considering its abilities and constraints,
4418 and document and explain the implementation to both technical and non-technical audiences
4419 C. Predict the behavior of systems under random events using knowledge of probability and expectation and inform
4420 users of its potential behavior.
4421 D. Assess the security of a system using the knowledge of confidentiality, availability, and integrity with an
4422 understanding of risks, threats, vulnerabilities and attack vectors, and relate its societal and ethical impact to the
4423 system's constituents.

4424

4425 **PBD-Platform-based Development**

- 4426 A. Design for a client a responsive web application utilizing a web framework and presentation technologies in
4427 support of a diverse online community.
4428 B. Develop a mobile app for a company that is usable, efficient, and secure on more than one device.
4429 C. Simulate for a company an industry platform.
4430 D. Develop and implement programming tasks via platform-specific APIs and present the results to a group of peers.
4431 E. Present the analysis of a mobile industrial system and illustrate correct security vulnerabilities.

4432

4433 **PD-Parallel and Distributed Computing**

- 4434 A. Design a scalable parallel algorithm for a computer firm by applying task-based decomposition or data-parallel
4435 decomposition.
4436 B. Write a program for a client that correctly terminates when all concurrent tasks terminate by considering actors
4437 and/or reactive processes, deadlocks, and properly synchronized queues.
4438 C. Write a test program for a company that reveals a concurrent programming error (e.g., missing an update when
4439 two activities both try to increment a variable).
4440 D. Present computational results of the work and span in a program by identifying independent tasks that may be
4441 parallelized and determining the critical path with respect to a parallel execution diagram.
4442 E. Implement a parallel divide-and-conquer (and/or graph algorithm) for a client by mapping and reducing
4443 operations for the real industry problem and empirically measure its performance relative to its sequential analog.

4444

4445 **PL-Programming Languages**

- 4446 A. Present the design and implementation of a class considering object-oriented encapsulation mechanisms (e.g.,
4447 class hierarchies, interfaces and private members).
4448 B. Produce a brief report on the implementation of a basic algorithm considering control flow in a program using
4449 dynamic dispatch that avoids assigning to a mutable state (or considering reference equality) for two different
4450 languages.
4451 C. Present the implementation of a useful function that takes and returns other functions considering variables and
4452 lexical scope in a program as well as functional encapsulation mechanisms.
4453 D. Use iterators and other operations on aggregates (including operations that take functions as arguments in two
4454 programming languages and present to a group of professionals ways of selecting the most natural idioms for
4455 each language.
4456 E. Contrast and present to peers (1) the procedural/functional approach (defining a function for each operation with
4457 the function body providing a case for each data variant) and (2) the object-oriented approach (defining a class
4458 for each data variant with the class definition providing a method for each operation).
4459 F. Write event handlers for a web developer for use in reactive systems such as GUIs.
4460 G. Demonstrate program pieces (such as functions, classes, methods) that use generic or compound types, including
4461 for collections to write programs.
4462 H. Write a program for a client to process a representation of code that illustrates the incorporation of an interpreter,
4463 an expression optimizer, and a documentation generator.
4464 I. Use type-error messages, memory leaks and dangling-pointer to debug a program for an engineering firm.

4465

4466 **SDF-Software Development Fundamentals**

- 4467 A. Create an appropriate algorithm to illustrate iterative, recursive functions, as well as divide-and-conquer
4468 techniques and use a programming language to implement, test, and debug the algorithm for solving a simple
4469 industry problem.
- 4470 B. Decompose a program for a client that identifies the data components and behaviors of multiple abstract data
4471 types and implementing a coherent abstract data type, with loose coupling between components and behaviors.
- 4472 C. Design, implement, test, and debug an industry program that uses fundamental programming constructs including
4473 basic computation, simple and file I/O, standard conditional and iterative structures, the definition of functions,
4474 and parameter passing.
- 4475 D. Present the costs and benefits of dynamic and static data structure implementations, choosing the appropriate data
4476 structure for modeling a given engineering problem.
- 4477 E. Apply consistent documentation and program style standards for a software engineering company that contribute
4478 to the readability and maintainability of software, conducting a personal and small-team code review on program
4479 component using a provided checklist.
- 4480 F. Demonstrate common coding errors, constructing and debugging programs using the standard libraries available
4481 with a chosen programming language.
- 4482 G. Refactor an industry program by identifying opportunities to apply procedural abstraction.
- 4483

SE-Software Engineering

- 4485 A. Conduct a review of a set of software requirements for a local project, distinguishing between functional and non-
4486 functional requirements, and evaluate the extent to which the set exhibits the characteristics of good requirements.
- 4487 B. Present to a client the design of a simple software system using a modeling notation (such as UML), including an
4488 explanation of how the design incorporated system design principles.
- 4489

SF-Systems Fundamentals

- 4491 A. Design a simple sequential problem and a parallel version of the same problem using fundamental building blocks
4492 of logic design and use appropriate tools to evaluate the design for a commercial organization and evaluate both
4493 problem versions.
- 4494 B. Develop a program for a local organization that incorporated error detection and recovery that incorporates
4495 appropriate tools for program tracing and debugging.
- 4496 C. Design a simple parallel program for a corporation that manages shared resources through synchronization
4497 primitives and use tools to evaluate program performance.
- 4498 D. Design and conduct a performance-oriented, pattern recognition experiment incorporating state machine
4499 descriptors and simple schedule algorithms for exploiting redundant information and data correction that is usable
4500 for a local engineering company and use appropriate tools to measure program performance.
- 4501 E. Calculate average memory access time and describe the tradeoffs in memory hierarchy performance in terms of
4502 capacity, miss/hit rate, and access time for a local engineering company.
- 4503 F. Measure the performance of two application instances running on separate virtual machines at a local engineering
4504 company and determine the effect of performance isolation.
- 4505

SP-Social Issues and Professional Practice

- 4507 A. Perform a system analysis for a local organization and present the results to them in a non-technical way.
- 4508 B. Integrate interdisciplinary knowledge to develop a program for a local organization.
- 4509 C. Document industry trends, innovations and new technologies and produce a report to influence a targeted
4510 workspace.
- 4511 D. Present to a group of professionals an innovative computer system by using audience-specific language and
4512 examples to illustrate the group's needs.
- 4513 E. Produce a document that is helpful to others that addresses the effect of societal change due to technology.
- 4514 F. Adopt processes to track customer requests, needs and their satisfaction
- 4515 G. Compare different error detection and correction methods for their data overhead, implementation complexity, and
4516 relative execution time for encoding, detecting, and correcting errors and ensure that any error does not affect
4517 humans adversely.
- 4518
- 4519
- 4520

4521 **Task force Members on the CS Subgroup**

4522

4523 Bruce McMillin (Leader)

4524 John Impagliazzo

4525 Richard LeBlanc

4526 Ariel Sabiguero Yawelak

4527 Ming Zhang

4528

4529

4530

4531

DRAFT

4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587

C.1.3: Information Systems Draft Competencies

CC2020
Information Systems (88) Competencies
Derived from IS2010
2018 June 07

Information Systems Specific Competencies

Identifying and designing opportunities for IT-enabled organizational improvement

1. Analyze the current fit between IT strategy and organizational strategy and take corrective action to align the two, when necessary.
2. Understand General Systems theory, including its key principles and applications
3. Model organizational processes with at least one modern business process modeling language.
4. Extract information systems requirements from future state process models.
5. Building on the foundation of risk-based management theory, apply risk analysis to real organizations.
6. Determine information systems requirements based on demonstrated needs for organizational controls.
7. Identify process performance indicators and monitors, applying industry recommendations like ITIL
8. Understand emerging technologies to identify innovative business opportunities based on these technologies.
9. Develop business proposals based on use of emerging technologies in an organization.
10. Apply entrepreneurial and creative thinking to transform organizations using emerging technologies
11. Analyze and document various business stakeholders' information requirements for a proposed system.
12. Apply modern industrial practices and techniques on system documentation and user interviewing (i.e. ITIL and PMBOK).
13. Apply foundational knowledge of human-computer interaction principles to systems and user interface design.
14. Apply knowledge on data visualization and representation their application to analytics and complex data representation.

Analyzing trade-offs

15. Identify and design the technology alternatives and manage risk across various options within an information systems project to select the most appropriate options based on the organizational needs and implement a solution that solves key business problems.
16. Justify an information systems project in terms of technical feasibility, business viability and cost-effectiveness to demonstrate the project's feasibility.
17. Analyze and compare solution options according to a variety of criteria and policies to evaluate the different possible solutions according to how well they promote the organizational needs.
18. Create a budget for IT-based solution and sourcing options to enable the organization to determine the financial impact of each option.
19. Analyze the cultural differences that affect a global business environment in order to show how cultural standards and expectations can have a positive impact on business success to support the process of selecting between options.

Designing and implementing information systems solutions

20. Design an enterprise architecture (EA) using formal approaches by identifying EA change needs and by addressing domain requirements and technology development.
21. Apply a systematic methodology for specifying system solution options based on the requirements for the information systems solution, considering in-house development, development from third-party providers, or purchased commercial-off-the-shelf (COTS) packages.
22. Design and implement a high-quality UX (user experience) for target users to enable effective support for the users' goals in their environment.
23. Design principles of information technology security and data infrastructure at the organizational level that enable them to plan, develop, and perform security tasks and apply them to organizational systems and databases.

- 4588 24. Design and implement an IT application that satisfies user needs in the context of processes that integrate analysis,
4589 design, implementation, and operations.
4590 25. Identify data and information management alternatives and suggest the most appropriate options based on the
4591 organizational information needs.
4592 26. Design data and information models aligned with organizational processes and compatible with data and
4593 information security management criteria.
4594 27. Select the suitable outsourcing contractors based on the external procurement selection criteria and manage people
4595 in development teams including selected contractors in multiple projects complex situation.
4596 28. Understand the processes, methods, techniques and tools that organizations use to manage information systems
4597 projects.
4598 29. Implement modern project management approaches to information systems project, demonstrating an
4599 understanding of a complex team- based activities are an inherent part of the project management.
4600

Managing ongoing information technology operations

- 4601
4602 30. Develop and implement plans of action for optimizing the use of enterprise technology resources.
4603 31. Develop indicators to assess the application performance and scalability.
4604 32. Monitor application performance indicators and implement corrective actions.
4605 33. Establish practices for optimized use of information systems and plan for a long term IS viability.
4606 34. Monitor and control IS to track performance and fit with organizational needs.
4607 35. Implement corrective actions by modifying the system as necessary.
4608 36. Negotiate and enforce contracts with providers of technology service in order to maintain the operational integrity
4609 of the technologies and services provided and be compliant with the roles and responsibilities of all parties
4610 involved.
4611 37. Develop, implement and monitor a security plan strategy based on a risk management model.
4612 38. Implement corrective security actions as necessary.
4613 39. Plan and implement procedures, operations and technologies for managing security and safety ensuring business
4614 continuity and information assurance from a disaster recovery situation.
4615
4616
4617

Foundational Competencies

Leadership and collaboration

- 4618
4619
4620
4621 40. Manage interpersonal relationships in a cross-cultural, cross-functional team.
4622 41. Provide a clearly articulated vision for the team so that it will be able to work towards a common goal.
4623 42. Support each individual member of the team in their effort to achieve their best possible level of individual
4624 performance.
4625 43. Specify sufficiently challenging goals for the team.
4626 44. Create a work breakdown structure, a task dependency model, and project schedule for a globally distributed
4627 project.
4628 45. Ensure that the project has sufficient resources and manage those resources in a context-appropriate way.
4629 46. Allocate project tasks to project resources in an equitable and achievable way.
4630 47. Monitor the progress the project is making.
4631 48. Respect different viewpoints between team members.
4632 49. View differences between team members as richness and a resource.
4633 50. Listen and consider carefully to the viewpoints of all team members.
4634 51. Establish and support decision structures that ensure equal opportunity to participate for all team members.
4635 52. Align the structure of an organization so that it supports the achievement of its goals.
4636 53. Select the organizational form based on criteria known to be effective.
4637 54. Execute the transformation of an organization's structure so that it does not unnecessarily disrupt its work.
4638 55. Monitor the effectiveness of an organizational structure continuously.
4639

Communication

- 4640
4641 56. Acquire facts and opinions regarding the domain of interest from various stakeholders in relevant organizational
4642 contexts using appropriate communication methods.
4643 57. Extract information from digital archives using modern data retrieval tools.
4644

- 4644 58. Communicate effectively in writing in a broad range of organizational contexts.
4645 59. Select the appropriate form of written communication for a specific organizational situation.
4646 60. Use state of the art virtual collaboration tools (such as wikis, blogs, and shared collaboration spaces) effectively
4647 in a variety of organizational situations.
4648 61. Communicate effectively orally with different audiences and using different channels in a variety of
4649 organizational situations.
4650 62. Identify and articulate key elements of a persuasive presentation to support a specific viewpoint.

4651

4652 **Negotiation**

- 4653 63. Apply a detailed problem analysis to determine the interests of each party in the negotiation in order to provide a
4654 clear proposal of the funding, time and staff required.
4655 64. Articulate and justify service levels for an IT service in terms of metrics that guarantee a description of the service
4656 being provided, reliability, responsiveness, procedure for reporting problems, monitoring and reporting service
4657 level, consequences for not meeting service obligations, and escape clauses or constraints.
4658 65. Demonstrate the specification and measurements for each area in the level of service definitions to allow the
4659 quality of service to be benchmarked.
4660 66. Identify and apply a more positive and confident approach to negotiating for each provider to support the quality
4661 enhancement of the project design as well as to ensure quality project preparation and implementation.
4662 67. Classify the key decision points, identify who is involved in making those decisions, and understand the actions
4663 and information that will be required for such decisions to be made within an information systems team in the
4664 context of competing internal interests.

4665

4666 **Analytical and critical thinking, including creativity and ethical analysis**

- 4667 68. Interpret and comply with legislative and regulatory requirements governing IT practices as well as industry
4668 standards for IT practices. Understand how culture and ethics shape compliance behavior.
4669 69. Analyze privacy and integrity guide for all IT practices.
4670 70. Identify complex situations and analyze the practices guide to ensure the ethical and legal corporate requirements
4671 are met.
4672 71. Identify the value of the systems.
4673 72. Identify the systems vulnerabilities.
4674 73. Identify the occurrence of a threat that may exploit a system vulnerabilities(s) aimed at compromising the systems.
4675 74. Identify a complex problem in, but separate from, its environment.
4676 75. Apply knowledge and understanding to solve the identified problem.
4677 76. Apply creative problem solving to technology-related issues.
4678 77. Select appropriate data collection methods and techniques for the investigation of domain activities.
4679 78. Capture and structure data and information requirements using appropriate conceptual modeling techniques.
4680 79. Reason effectively with a learned audience based on the results of quantitative analyses.
4681 80. Apply adequate quantitative analysis techniques according to the data analysis goal.
4682 81. 81 Develop innovative and creative models that rely on new uses of existing technology or new technologies
4683 themselves.
4684 82. Develop a plan to exploit new and emerging methods and technologies for new purposes within an organization.
4685 83. Devise new ways of structuring and performing domain activities at different levels (individual, team, process
4686 and organization) while considering the enabling and enhancing effects of information technology applications.
4687 84. Estimate the benefits of the new designs, assess the consequences of their implementation, and anticipate potential
4688 adverse consequences.

4689

4690 **Mathematical foundations**

- 4691 85. Identify those domain of interest problems that can be addressed mathematically and find a mathematical
4692 formulation for those problems.
4693 86. Use logical thought processes to divide a problem into smaller components and make inferences based on problem
4694 components.
4695 87. Select and implement an effective mathematical strategy.
4696 88. Communicate mathematical results effectively to a variety of stakeholders. (See, for example, Turner 2010 at
4697 <https://research.acer.edu.au/cgi/viewcontent.cgi?article=1083&context=resdev>)
4698

4699

4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716

Task force Members on the Information Systems Subgroup

Hala Alrumaih	hala.alrumaih@gmail.com
Teresa Pereira	tpereira@esce.ipv.pt
Ariel Sabiguero	asabigue@fing.edu.uy
Eiji Hayashiguchi	eiji_hayashiguchi@biscuit.ocn.ne.jp
Heikki Topi*	htopi@bentley.edu
John Impagliazzo* (ex officio)	john.impagliazzo@hofstra.edu

* = Steering committee (SC) member

DRAFT

4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772

C.1.4: Information Technology Competencies

CC2020 Information Technology Competencies [Duplicated from the IT2017 Report]

ITE-CSP Cybersecurity Principles [6%]

- A. Evaluate the purpose and function of cybersecurity technology identifying the tools and systems that reduce the risk of data breaches while enabling vital organization practices. (*Cybersecurity functions*)
- B. Implement systems, apply tools, and use concepts to minimize the risk to an organization’s cyberspace to address cybersecurity threats. (*Tools and threats*)
- C. Use a risk management approach for responding to and recovering from a cyber-attack on system that contains high value information and assets such as an email system. (*Response and risks*)
- D. Develop policies and procedures needed to respond and remediate a cyber-attack on a credit card system and describe plan to restore functionality to the infrastructure. (*Policies and procedures*)

ITE-GPP Global Professional Practice [3%]

- A. Analyze the importance of communication skills in a team environment and determine how these skills contribute to the optimization of organization goals. (*Communication and teamwork*)
- B. Evaluate the specific skills necessary for maintaining continued employment in an IT career that involves system development in an environmental context. (*Employability*)
- C. Develop IT policies within an organization that include privacy, legal, and ethical considerations as they relate to a corporate setting. (*Legal and ethical*)
- D. Evaluate related issues facing an IT project and develop a project plan using a cost/benefit analysis including risk considerations in creating an effective project plan from its start to its completion. (*Project management*)

ITE-IMA Information Management [6%]

- A. Express how the growth of the internet and demands for information have changed data handling and transactional and analytical processing, and led to the creation of special purpose databases. (*Requirements*)
- B. Design and implement a physical model based on appropriate organization rules for a given scenario including the impact of normalization and indexes. (*Requirements and development*)
- C. Create working SQL statements for simple and intermediate queries to create and modify data and database objects to store, manipulate and analyze enterprise data. (*Testing and performance*)
- D. Analyze ways data fragmentation, replication, and allocation affect database performance in an enterprise environment. (*Integration and evaluation*)
- E. Perform major database administration tasks such as create and manage database users, roles and privileges, backup, and restore database objects to ensure organizational efficiency, continuity, and information security. (*Testing and performance*)

ITE-IST Integrated Systems Technology [3%]

- A. Illustrate how to code and store characters, images and other forms of data in computers and show why data conversion is often a necessity when merging disparate computing systems together. (*Data mapping and exchange*)
- B. Show how a commonly used intersystem communication protocol works, including its advantages and disadvantages. (*Intersystem communication protocols*)
- C. Design, debug and test a script that includes selection, repetition and parameter passing. (*Integrative programming and scripting*)
- D. Illustrate the goals of secure coding, and show how to use these goals as guideposts in dealing with preventing buffer overflow, wrapper code, and securing method access. (*Defensible integration*)

4773

4774 **ITE-NET Networking [5%]**

- 4775 A. Analyze and compare the characteristics of various communication protocols and how they support application requirements within a telecommunication system. (*Requirements and Technologies*)
- 4776 B. Analyze and compare several networking topologies in terms of robustness, expandability, and throughput used within a cloud enterprise. (*Technologies*)
- 4777 C. Describe different network standards, components, and requirements of network protocols within a distributed computing setting. (*Network protocol technologies*)
- 4778 D. Produce managerial policies to address server breakdown issues within a banking system. (*Risk Management*)
- 4779 E. Explain different main issues related to network management. (*Network Management*)

4783

4784 **ITE-PFT Platform Technologies [1%]**

- 4785 A. Describe how the historical development of hardware and operating system computing platforms produced the computing systems we have today. (*Computing systems*)
- 4786 B. Show how to choose among operating system options, and install at least an operating system on a computer device. (*Operating systems*)
- 4787 C. Justify the need for power and heat budgets within an IT environment, and document the factors needed when considering power and heat in a computing system. (*Computing infrastructure*)
- 4788 D. Produce a block diagram, including interconnections, of the main parts of a computer, and illustrate methods used on a computer for storing and retrieving data. (*Architecture and organization*)

4792

4793 **ITE-SPA System Paradigms [6%]**

- 4794 A. Justify the way IT systems within an organization can represent stakeholders using different architectures and the ways these architectures relate to a system lifecycle. (*Requirements and development*)
- 4795 B. Demonstrate a procurement process for software and hardware acquisition and explain the procedures one might use for testing the critical issues that could affect IT system performance. (*Testing and performance*)
- 4796 C. Evaluate integration choices for middleware platforms and demonstrate how these choices affect testing and evaluation within the development of an IT system. (*Integration and evaluation*)
- 4797 D. Use knowledge of information technology and sensitivity to the goals and constraints of the organization to develop and monitor effective and appropriate system administration policies within a government environment. (*System governance*)
- 4798 E. Develop and implement procedures and employ technologies to achieve administrative policies within a corporate environment. (*Operational activities*)
- 4799 F. Organize personnel and information technology resources into appropriate administrative domains in a technical center. (*Operational domains*)
- 4800 G. Use appropriate and emerging technologies to improve performance of systems and discover the cause of performance problems in a system. (*Performance analysis*)

4809

4810 **ITE-SWF Software Fundamentals [4%]**

- 4811 A. Use multiple levels of abstraction and select appropriate data structures to create a new program that is socially relevant and requires teamwork. (*Program development*)
- 4812 B. Evaluate how to write a program in terms of program style, intended behavior on specific inputs, correctness of program components, and descriptions of program functionality. (*App development practices*)
- 4813 C. Develop algorithms to solve a computational problem and explain how programs implement algorithms in terms of instruction processing, program execution, and running processes. (*Algorithm development*)
- 4814 D. Collaborate in the creation of an interesting and relevant app (mobile or web) based on user experience design, functionality, and security analysis and build the app's program using standard libraries, unit testing tools, and collaborative version control. (*App development practices*)

4821

4822 **ITE-UXD User Experience Design [3%]**

- 4823 A. Design an interactive application, applying a user-centered design cycle and related tools and techniques (e.g., prototyping), aiming at usability and relevant user experience within a corporate environment. (*Design tools and techniques*)
- 4824 B. For a case of user centered design, analyze and evaluate the context of use, stakeholder needs, state-of-the-art interaction opportunities, and envisioned solutions, considering user attitude and applying relevant tools and
- 4825
- 4826
- 4827

- 4828 techniques (e.g., heuristic evaluation), aiming at universal access and inclusiveness, and showing a responsive
4829 design attitude, considering assistive technologies and culture sensitive design. (*Stakeholder needs*)
4830 C. For evaluation of user-centered design, articulate evaluation criteria and compliance to relevant standards
4831 (*Benchmarks and standards*)
4832 D. In design and analysis, apply knowledge from related disciplines including human information processing,
4833 anthropology and ethnography, and ergonomics/human factors. (*Integrative design*)
4834 E. Apply experience design for a service domain related to several disciplines, focusing on multiple stakeholders
4835 and collaborating in an interdisciplinary design team. (*Application design*)
4836

4837 **ITE-WMS Web and Mobile Systems [3%]**

- 4838 A. Design a responsive web application utilizing a web framework and presentation technologies in support of a
4839 diverse online community. (*Web application development*)
4840 B. Develop a mobile app that is usable, efficient, and secure on more than one device. (*Mobile app development*)
4841 C. Analyze a web or mobile system and correct security vulnerabilities. (*Web and mobile security*)
4842 D. Implement storage, transfer, and retrieval of digital media in a web application with appropriate file, database, or
4843 streaming formats. (*Digital media storage and transfer*)
4844 E. Describe the major components of a web system and how they function together, including the web server,
4845 database, analytics, and front end. (*Web system infrastructure*)
4846
4847
4848
4849
4850
4851

4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906

C.1.5: Software Engineering Draft Competencies

CC2020
Software Engineering
Draft Competencies
May 1, 2018

Software Requirements

1. Identify and document software requirements by applying a known requirements elicitation technique in work sessions with stakeholders, using facilitative skills, as a contributing member of a requirements team.
2. Analyze software requirements for consistency, completeness, and feasibility, and recommend improved requirements documentation, as a contributing member of a requirements team.
3. Specify software requirements using standard specification formats and languages that have been selected for the project, and be able to describe the requirements in an understandable way to non-experts such as end users, other stakeholders, or administrative managers, as a contributing member of a requirements team.
4. Verify and validate the requirements using standard techniques, including inspection, modeling, prototyping, and test case development, as a contributing member of a requirements team.
5. Follow process and product management procedures that have been identified for the project, as a contributing member of the requirements engineering team.

Software Design

1. Present to business decision makers architecturally significant requirements from a software requirements specification document.
2. Evaluate and compare tradeoffs from alternative design possibilities for satisfying functional and non-functional requirements and write a brief proposal summarizing key conclusions for a client.
3. Produce a high-level design of specific subsystems that is presentable to a non-computing audience by considering architectural and design patterns.
4. Produce detailed designs for a client for specific subsystem high-level designs by using design principles and cross-cutting aspects to satisfy functional and non-functional requirements.
5. Evaluate software testing consideration of quality attributes in the design of subsystems and modules for a developer/manufacturer.
6. Create software design documents which communicate effectively to software design clients such as analysts, implementers, test planners, or maintainers.

Software Construction

1. Design and implement an API using an object-oriented language and extended libraries, including parameterization and generics on a small project.
2. Evaluate a software system against modern software practices such as defensive programming, error and exception handling, accepted fault tolerances, in a runtime mode that considers state-based table-driven constructions on a large project, as a member of a project team.
3. Develop a distributed cloud-based system that incorporates grammar-based inputs and concurrency primitives for a medium-size project and then conduct a performance analysis to fine-tune the system, as a member of a project team.

Software Testing

1. Perform an integrative test and analysis of software components by using black-box and use case techniques in collaboration the clients.
2. Conduct a regressive test of software components for a client that considers operational profiles and quality attributes specific to the application in accordance with empirical data and the intended usages.
3. Conduct a test utilizing appropriate testing tools focused on desirable quality attributes specified by the quality control team and the client.

- 4907 4. Plan and conduct process to design test cases for an organization using both clear- and black-box techniques to
4908 measure quality metrics in terms of coverage and performance.
4909

4910 Software Sustainment

- 4911 1. Describe the criteria for transition into a sustainment status and assist in identifying applicable systems and
4912 software operational standards.
4913 2. Relate to the needs of operational support personnel for documentation and training, and help develop software
4914 transition documentation and operational support training materials.
4915 3. Help in determining the impacts of software changes on the operational environment.
4916 4. Describe the elements of software support activities, such as configuration management, operational software
4917 assurance, help desk activities, operational data analysis, and software retirement.
4918 5. Perform software support activities; and interact effectively with other software support personnel.
4919 6. Assist in implementing software maintenance processes and plans, and make changes to software to implement
4920 maintenance needs and requests.
4921

4922 Software Process and Life Cycle

- 4923 1. Engage with a team to translate a software development process into individual areas of responsibility.
4924 2. Commit to and perform tasks related to assigned or agreed upon areas of responsibility. [MEE: Would it make
4925 sense to designate "on time, or "with a reasonable explanation for and plan for addressing delays"]
4926 3. Propose and provide a justification for software lifecycle process improvements based on team capacity, project
4927 progress data, and quality analysis as part of a software development team's retrospective activities.
4928

4929 Software Systems Engineering

- 4930 1. Provide a description of system engineering concepts and activities to identify problems or opportunities, explore
4931 alternatives, create models and test them.
4932 2. Develop the big picture of a system in its context and environment in order to simplify and improve system
4933 architectures for supporting system designers.
4934 3. Develop interfaces, which interact with other subsystems. Use information hiding to isolate the contents and
4935 collaborations within subsystems, so that clients of the subsystem need not be aware of the internal design of
4936 subsystems.
4937 4. Work effectively with engineers and developers from other disciplines to ensure effective interaction.
4938

4939 Software Quality

- 4940 1. Distinguish quality attributes that are discernable at run-time (performance, security, availability, functionality,
4941 usability), from those not discernable at run-time (modifiability, portability, reusability, integrability, and
4942 testability), and those related to the intrinsic qualities of architecture and detailed design (conceptual integrity,
4943 correctness, and completeness). [Based on SWEBOK 2014]
4944 2. Design, coordinate, and execute, within a project team, software quality assurance plans for small software
4945 subsystems and modules, considering the way in which quality attributes are discernable. Correspondingly,
4946 measure, document, and communicate appropriately the results.
4947 3. Perform peer code reviews for evaluating quality attributes that are not discernable at run-time.
4948 4. Explain the statistical nature of quality evaluation when performed on software execution; develop, deploy and
4949 implement approaches to collect statistical usage and testing outcome data; compute and analyze statistics on
4950 outcome data.
4951 5. Interact with external entities including clients, users, and auditing agencies in conveying quality goals for
4952 processes and products.
4953

4954 Software Security

- 4955 1. Apply the project's selected security lifecycle model (e.g. Microsoft SDL), as a contributing member of a project
4956 team.
4957 2. Identify security requirements by applying the selected security requirements method, as a contributing member
4958 of a software project team.
4959 3. Incorporate security requirements into architecture, high-level, and detailed design, as a contributing member of
4960 a software project team.
4961 4. Develop software using secure coding standards.
4962 5. Execute test cases that are specific to security.

- 4963 6. Adhere to the project's software development process, as a contributing member of a software project team.
4964 7. Develop software that supports the project's quality goals and adheres to quality requirements.
4965

4966 **Software Safety**

- 4967 1. Describe the principal activities with the development of software systems, which involve safety concerns
4968 (activities related to requirements, design, construction, and quality);
4969 2. Create and verify preliminary hazard lists; perform hazard and risk analyses, identify safety requirements;
4970 3. Implement and verify design solutions, using safe design and coding practices, to assure that the hazards are
4971 mitigated and the safety requirements are met;
4972 4. Be aware of the consequences of the development of unsafe software, that is, the negative affect on those who
4973 use or receive services from the software.
4974

4975 **Software Configuration Management**

4976 [None]
4977

4978 **Software Measurement**

- 4979 1. Develop and implement plans for measurement of software processes and work products using appropriate
4980 methods, tools, and abilities.
4981

4982 **Human-Computer Interaction**

4983 [None]
4984

4985 **Project Management**

- 4986 1. Explain the principal elements of management for a small project team;
4987 2. Assist in the managerial aspects of a small project team, including software estimation, project planning and
4988 tracking, staffing and resource allocation, and risk management;
4989 3. Develop and implement plans for measurement of software processes and work products using appropriate
4990 methods and tools.
4991 4. Work effectively with other team members in project management activities.
4992

4993 **Behavioral Attributes**

- 4994 1. Engage with team members to collaborate in solving a problem, effectively applying oral and/or written
4995 communication skills. Work done towards team effort is accomplished on time; it is in compliance with the role
4996 played in the team: it uses established quality procedures; and it advances the team effort.
4997 2. Assist in the analysis and presentation of a complex problem, taking into account the needs of stakeholders from
4998 diverse cultures, needs, and/or geographic locations. Help in developing a solution for the problem and presenting
4999 it to stakeholders, explaining the economic, social and/or environmental impact of the proposed solution. Identify
5000 areas of uncertainty or ambiguity, and explain how these have been managed.
5001 3. Analyze software employment contracts from various social and legal perspectives, ensuring that the final product
5002 conforms to professional and ethical expectations, and follows standard licensing practices.
5003 4. Locate and make sense of learning resources, and use these to expand knowledge, skills, and dispositions. Reflect
5004 upon one's own learning and how it provides a foundation for future growth.
5005
5006

5007 **Software Engineering Subgroup Members who are Task Force Members**

5008
5009 Nancy Mead (Leader)
5010 Hala Alrumaih
5011 Marisa Exter
5012 Rich LeBlanc
5013 John Impagliazzo
5014 Barbara Viola
5015
5016

5017 **Software Engineering Subgroup Members who are not Task Force Members (Contributors)**

5018

5019 Kai H. Chang, Auburn University

5020 Dick Fairley, Software and Systems Engineering Associates

5021 Kevin Gary, Arizona State University

5022 Thomas Hilburn, Embry-Riddle Aeronautical University

5023 Gabriel Tamura, Universidad Icesi, Colombia

5024 Chris Taylor, Milwaukee School of Engineering

5025 Jim Vallino, Rochester Institute of Technology

5026 Norha M. Villegas, Universidad Icesi, Colombia

5027

5028

DRAFT

5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5080
5081
5082
5083
5084

C.1.6: Master's in Information Systems Draft Competencies

Development of Information Systems MSIS Competencies (Knowledge + Skill + Disposition)

Derived from the MSIS2016 Report

Business Continuity and Information Assurance (BCIA)

From MSIS2016 Report, Page 16

Competencies in the area of Business Continuity and Information Assurance (BCIA)

Area description: *the Business Continuity and Information Assurance competency area mainly concerns the continuity, auditing, and assurance of information systems. It generally covers areas such as risk avoidance, security management, and quality auditing. The challenging issues related to business continuity and information assurance span from tactical and strategic to technical and operational levels. They often involve a range of processes from management, such as policy and standard setting, to hands-on skills, such as system contingency and recovery planning.*

High-level area dimensions: *Graduates will be able to:*

- 1. Create policies and standards for business continuity and information assurance.*
- 2. Plan and implement procedures, operations, and technologies for managing risk and trust, security, and safety and for business continuity and disaster recovery.*
- 3. Monitor, control, and institutionalize the protection and growth of the hardware, software, and information assets in information systems.*

IS Competencies = (Knowledge + Skill + Disposition)

Business Continuity and Information Assurance [BCIA]

- Analyze policies and standards for business continuity and information assurance and present the findings to a group of peers.
[Bcia1]
- Plan procedures, operations, and technologies for managing security and safety in a disaster recovery situation.
[Bcia2]
- Monitor the protection and growth of hardware and software within an information system for a small company.
[Bcia3]

Data, Information, and Content management (DATA)

From MSIS2016 Report, Page 16

Competencies in the area of Data, Information, and Content management (DATA)

Area description: *the Data, Information, and Content management area covers competencies that enable graduates to be effective contributors in processes that improve the domain's ability to achieve its goals using structured and unstructured data and information effectively.*

High-level area dimensions: *Graduates will be able to:*

- 5085 1. *Identify data and information management technology alternatives, select the most appropriate options based on*
5086 *the organizational information needs, and manage the implementation of the selected options.*
5087 2. *Identify, create, and manage organizational policies and processes related to data and information management*
5088 *by balancing multidimensional requirements, such as legal and regulatory requirements, ethical considerations*
5089 *and implications of technology decisions, organizational business requirements, data quality issues, and*
5090 *requirements of operating in an international environment.*
5091 3. *Analyze the needs of a domain and determine how those needs can best be addressed with data, information, and*
5092 *content management solutions.*

5093
5094 **IS Competencies = (Knowledge + Skill + Disposition)**

5095 **Data, Information, and Content management [DATA]**

- 5096
5097 D. Identify and report data and information management technology alternatives for a small organization and suggest
5098 to management the most appropriate options based on the organizational information needs.
5099 [Data1]
5100 E. Identify organizational policies and processes related to data and information management within a team
5101 environment and how to address information and content management solutions for policy infringement.
5102 [Data2, Data3]

5103
5104
5105 **Enterprise Architecture (EARC)**

5106
5107 From MSIS2016 Report, Page 17

5108
5109 **Competencies in the area of Enterprise Architecture (EARC)**

5110
5111 ***Area description:** Enterprise architecture has two aims: managing the complexity of information systems and*
5112 *technologies and aligning these systems/technologies with the organization's strategy. The area covers competences*
5113 *that enable graduates to participate in planning, building, using, maintaining, and evaluating architectures.*

5114
5115 ***High-level area dimensions:** Graduates will be able to:*

- 5116 1. *Design an enterprise architecture (EA). This involves identifying and applying a formal approach to EA*
5117 *development, performing the multistage process of developing an EA, identifying the EA change needs, and*
5118 *applying them to the EA. Graduates are able to incorporate information, domain activity processes, technology*
5119 *platforms, applications, software, and hardware in the EA while considering the alignment between domain*
5120 *requirements and technology development.*
5121 2. *Deploy and maintain an EA. This involves conveying the architecture to domain process owners, software*
5122 *development and maintenance projects, and infrastructure planners and taking into account the current status of*
5123 *the projects and infrastructure. This also includes gathering input from the enterprise and from technology*
5124 *developments for maintaining the architecture and performing architecture maintenance.*

5125
5126 **IS Competencies = (Knowledge + Skill + Disposition)**

5127 **Enterprise Architecture [EARC]**

- 5128
5129 F. Evaluate an enterprise architecture (EA) using formal approaches by identifying the EA change needs and by
5130 addressing domain requirements and technology development through a written report.
5131 [Earc1]
5132 G. Describe to a group of managers an enterprise architecture (EA) highlighting software development and
5133 maintenance by gathering input from the enterprise to evaluate the level of maintenance involved.
5134 [Earc2]

5135
5136
5137
5138 **Ethics, Impacts, and Sustainability (ETIS)**

5139
5140 From MSIS2016 Report, Page 17

5141
5142
5143
5144
5145
5146
5147
5148
5149
5150
5151
5152
5153
5154
5155
5156
5157
5158
5159
5160
5161
5162
5163
5164
5165
5166
5167
5168
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196

Competencies in the area of Ethics, Impacts and Sustainability (ETIS)

Area description: *The Ethics, Impacts, and Sustainability competency area covers the conceptualization and implementation of environmentally and socially sustainable IT solutions that are aligned with the responsibilities of organizations and in compliance with legislative and regulatory requirements and industry standards. This competency area addresses key questions such as environmental and social sustainability, safety and health, privacy, and integrity. It also covers the impact of IT on the nature of work and workplaces and explores how culture and ethics (internal pertaining to organizations and external pertaining to stakeholders) shape behavior. These areas tend to be aligned with a strategic or a tactical level of organizational decision making.*

High-level area dimensions: *Graduates will be able to:*

- 1. Apply sustainable approaches for IT solutions development, IT procurement, IT operations, IT resources management, and other IT practices.*
- 2. Ensure safety and avoid health hazards for contract arrangements with external parties and internal systems development, maintenance, and reuse.*
- 3. Ensure that privacy and integrity guide all IT practices.*
- 4. Interpret and comply with legislative and regulatory requirements governing IT practices as well as industry standards for IT practices. Understand how culture and ethics shape compliance behavior.*

IS Competencies = (Knowledge + Skill + Disposition)

Ethics, Impacts, and Sustainability [ETIS]

- H. Apply sustainable system approaches by incorporating multiple IT practices for a corporate environment in a manner that ensures personnel privacy and integrity.
[Etis1, Etis3]
- I. Develop a policy concerning contracts usable within an enterprise or government that ensures safety and health standards in compliance with regulatory statutes and requirements for mutual benefit irrespective of cultural and personal characteristics.
[Etis2, Etis4]

Innovation, Organizational Change, and Entrepreneurship (IOCE)

From MSIS2016 Report, Page 18

Competencies in the area of Innovation, Organizational Change, and Entrepreneurship (IOCE)

Area description: *the Innovation, Organizational Change, and Entrepreneurship area covers the capability to recognize and exploit the potential afforded by current and upcoming technologies to address existing and new business opportunities. This area also includes competencies required to understand and to intervene in different forms of domain activities (e.g., work units, work teams, processes, organizations, markets, society setting) in order to use information technologies to improve the way those business activities are structured and performed.*

High-level area dimensions: *Graduates will be able to:*

- 1. Monitor the environment in order to identify and evaluate new IS methods and trends in terms of their appropriateness for an organization.*
- 2. Develop innovative domain activity models that rely on new uses of existing technology or new technologies themselves.*
- 3. Develop a plan to exploit new and emerging methods and technologies for new purposes in an organization.*
- 4. Devise new ways of structuring and performing domain activities at different levels (individual, team, process, and organization) while considering the enabling and enhancing effects of information technology applications.*
- 5. Estimate the benefits of the new designs, assess the consequences of their implementation, and anticipate potential adverse consequences.*

IS Competencies = (Knowledge + Skill + Disposition)

5197 **Innovation, Organizational Change, and Entrepreneurship [IOCE]**

5198

5199 J. Report to the management of an organization new IS methods and trends and suggest innovative activity models
5200 that rely on new uses of existing technologies.

5201 [Ioce1, Ioce2]

5202 K. Explain ways of exploiting emerging technologies at different levels (individual, team, process, and organization)
5203 and address the enabling or enhancing effects of information technology applications.

5204 [Ioce3, Ioce4]

5205 L. Report to peers the benefits of a new information system design and highlight potential adverse consequences of
5206 the system.

5207 [Ioce5]

5208

5209

5210 **IS Management and Operations (ISMO)**

5211

5212 From MSIS2016 Report, Page 18

5213

5214 **Competencies in the area of IS Management and Operations (ISMO)**

5215

5216 *Area description: the IS Management and Operations area covers the capability to develop, maintain, and*
5217 *consistently improve domain performance while providing appropriate information systems, services, and*
5218 *infrastructure. The capability focuses externally on creating value for the domain and internally on IS staff motivation,*
5219 *performance, and accountability.*

5220

5221 **High-level area dimensions:** Graduates will be able to:

5222 1. Apply professional management skills to the design and management of an effective IS organization.

5223 2. Ensure operational efficiency and effectiveness in service delivery.

5224 3. Govern IS project management principles and support their use in the organization.

5225 4. Manage information systems use.

5226 5. Manage information resources together with line management.

5227

5228 **IS Competencies = (Knowledge + Skill + Disposition)**

5229 **IS Management and Operations [ISMO]**

5230

5231 M. Identify the professional management skills needed to the design and management an effective IS organization
5232 that ensures operational efficiency n service delivery.

5233 [Ismo1, Ismo2]

5234 N. Analyze and report IS project management principles that support their use in the organization.

5235 [Ismo3]

5236 O. Evaluate the use of information systems and resources and present the finding to the management of an
5237 organization.

5238 [Ismo4, Ismo5]

5239

5240

5241 **IS Strategy and Governance (ISSG)**

5242

5243 From MSIS2016 Report, Page 18

5244

5245 **Competencies in the area of IS Strategy and Governance (ISSG)**

5246

5247 *Area description: the IS Strategy and Governance area covers the creation and implementation of long-term plans*
5248 *for designing, delivering, and using organizational information systems to achieve strategic domain goals and*
5249 *objectives. This area also covers monitoring and controlling organizational IS resources to ensure alignment with*
5250 *and achievement of strategies, goals, and objectives.*

5251

5252 **High-level area dimensions:** Graduates will be able to:

- 5253 1. *Analyze the effect and impact of IS on industries, firms, and institutions; develop and implement plans of action*
5254 *for maximizing firm benefits associated with IS design, delivery, and use; and manage IS resources financially.*
5255 2. *Create and manage the oversight mechanisms by which an organization evaluates, directs, and monitors*
5256 *organizational IT. These mechanisms may leverage one or more governance frameworks; hence, understanding*
5257 *the process of applying and analyzing a framework is a critical competency. Distribution of decision rights and*
5258 *organizational decision-making practices are other key components of this competency area.*
5259 3. *Establish practices for minimizing environmental impacts and planning for long-term firm viability.*

5260

5261 **IS Competencies = (Knowledge + Skill + Disposition)**

5262 **IS Strategy and Governance [ISSG]**

5263

5264 P. Identify the effect of IS on industries, firms, and institutions and suggest to organizational managers plans for
5265 maximizing firm benefits associated with IS design, delivery, and use.

5266 [Issg1]

5267 Q. Report to peers some oversight mechanisms by which an organization evaluates, directs, and monitors
5268 organizational IT by leveraging one or more governance frameworks and organizational decision-making
5269 practices.

5270 [Issg2]

5271 R. Recommend to organizational managers some practices for minimizing environmental effects and suggest ways
5272 for long-term organizational viability.

5273 [Issg3]

5274

5275

5276 **IT Infrastructure (INFR)**

5277

5278 From MSIS2016 Report, Page 19

5279

5280 **Competencies in the area of IT Infrastructure (INFR)**

5281

5282 *Area description: the IT Infrastructure area covers competencies that allow graduates to contribute to needs analysis*
5283 *for and design and implementation of effective, technically correct IT infrastructure solutions.*

5284

5285 **High-level area dimensions:** Graduates will be able to:

5286 1. *Design integrated communication networks for small- and medium-size organizations, including local area*
5287 *networks and the use of wide-area network technologies to connect the local networks.*

5288 2. *Specify requirements for large-scale network solutions.*

5289 3. *Design an implementation architecture for organizational data processing and system solutions, using both*
5290 *internal hardware resources and external service solutions.*

5291 4. *Negotiate and enforce contracts with providers of IT infrastructure services.*

5292

5293 **IS Competencies = (Knowledge + Skill + Disposition)**

5294 **IT Infrastructure [INFR]**

5295

5296 S. Evaluate an integrated communication network for a medium-size organization that includes local-area and wide-
5297 area network technologies and specify requirements for a large-scale network expansion.

5298 [Infr1, Infr2]

5299 T. Analyze and provide a written report of an implementation architecture for organizational data processing system
5300 that uses both internal hardware resources.

5301 [Infr3]

5302 U. Enhance the financial aspects of a contract that involves providers of several IT infrastructure services.

5303 [Infr4]

5304

5305

5306

5307

5308

5309 **Systems Development and Deployment (SDAD)**

5310

5311 From MSIS2016 Report, Page 19

5312

5313 ***Competencies in the area of Systems Development and Deployment (SDAD)***

5314

5315 ***Area description:*** *the Systems Development and Deployment area covers the design of information systems and*
5316 *services, including the design of how humans interact with and how they experience IT artifacts. It also includes*
5317 *competencies related to systems implementation and the deployment of systems to organizational use.*

5318

5319 ***High-level area dimensions:*** *Graduates will be able to:*

5320

5321 1. *Analyze and specify requirements for IT artifacts through studying and documenting the whole or part of some*
5322 *form of domain activities (e.g., work unit, work team, process, organization, market, society setting) in terms of*
5323 *the actions they involve and the information they deal with. Define requirements for IT artifacts that can enhance*
5324 *the way existing domain activities are structured and performed or enabling new forms of domain activities*

5325

5326 2. *Design and document IT artifacts that meet specified requirements taking into account non-functional*
5327 *requirements (including user experience design) and organizational, technical, infrastructural, and other*
5328 *constraints.*

5329

5330 3. *In the context of iterative processes that integrate analysis, design, implementation, and operations, develop and*
5331 *deploy IT applications that satisfy user needs.*

5332

5333 **IS Competencies = (Knowledge + Skill + Disposition)**

5334 **Systems Development and Deployment [SDAD]**

5335

5336 V. Describe to an audience the requirements for an IT artifact that enhances the way existing domain activities are
5337 structured and performed.

5338 [Sdad1]

5339 W. Report on an IT artifact that meets specified requirements considering non-functional requirements and
5340 organizational constraints.

5341 [Sdad2]

5342 X. Deploy an IT application that satisfies user needs in the context of processes that integrate analysis, design,
5343 implementation, and operations.

5344 [Sdad3]

5345

5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365

C.2: References for Appendix C

- [Acm03] ACM (2010) *IS 2010 Curriculum Guidelines for Undergraduate Degree Programs in Information Systems*, Association for Computing Machinery (ACM) and Association for Information Systems (AIS).
- [Acm04] ACM (2013). *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, Association for Computing Machinery and IEEE Computer Society.
- [Acm05] ACM (2014). *Software Engineering Curricula 2014 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*, Association for Computing Machinery and IEEE Computer Society.
- [Acm06] ACM (2016). *Computer Engineering Curricula 2016 Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering*, Association for Computing Machinery (ACM) and IEEE Computer Society, ISBN: 978-1-4503-4875-1, DOI: 10.1145/3025098, Web link: <https://dx.doi.org/10.1145/3025098>.
- [Acm07] ACM (2017). *Information Technology Curricula 2017 Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology*, Association for Computing Machinery (ACM) IEEE Computer Society (IEEE-CS), ISBN: 978-1-4503-6416-4, DOI: 10.1145/3173161, Web link: <https://dl.acm.org/citation.cfm?id=3173161>.
- [Acm11] ACM (2017) Topi, H., Karsten, H., Brown, S. A., Carvalho, J. A., Donnellan, B., Shen, J., et al. (2017). MSIS 2016: Global Competency Model for Graduate Degree Programs in Information Systems, 40(1). Retrieved from <http://aisel.aisnet.org/cais/vol40/iss1/186>

5366

5367 **Appendix D: Competency-Based Computing Curricula**

5368

5369

5370

5371

5372

5373

5374

Computing curricula are the educational matter that define the course of study in baccalaureate programs. CC2020 sees a strategic imperative to shape the future of computing education by reshaping the language used for defining curricula. Within the broader context of industry, professions, and society as a whole a curriculum description centered on competency focuses on the individual's capability to apply their computing education in the practical service to society.

5375

5376

5377

5378

5379

5380

A whole curriculum centered on competency informs pedagogy, academic and professional assessment, ethical conduct, relevance to industry, and society. Effective computing education must envelop the individual's *knowing what*, *knowing how*, and *knowing why* to engage their computing education. To better effect these ends, CC2020 proposes a philosophical shift in the format and emphasis of computing curricula through the adoption of a competency model for curriculum specification.

5381

5382

5383

5384

5385

5386

5387

5388

Adopting a coherent competency model to define computing curricula will more clearly promote describing the practical benefits of computing programs to students, benefactors, faculty, administrators, employers, accreditors, lawmakers, and society as a whole. Describing a computing competence in a practical context shifts the focus of curricula away from describing a body of knowledge in relation to a disciplinary area toward pragmatic student accomplishment. Descriptions of what graduates can do in practical situations replace descriptions of content and classroom time. Competency more effectively describes outcome expectations: challenge educators to develop more proficient computing professionals and lead society to recognize the purpose and benefits of computing education.

5389

5390

5391

5392

5393

5394

5395

5396

5397

5398

5399

This appendix presents CC2020's definition of competency and a template for specifying the subject matter of baccalaureate computing education. The competency template is composed of a prose statement of *task* and a structured list of components: *knowledge*, *skills*, and *dispositions*. The model structure is elaborated by examining each of the components in relation to the others and as a whole in contrast to the time-honored definitions of *knowledge area*, *knowledge unit*, *learning outcome* (Ka-Ku-Lo). We will address how competency can be leveraged by different stakeholder groups and at different levels in undergraduate/first-cycle computing programs. The chapter concludes acknowledging the most recent efforts to incorporate competency in computing curriculum guidelines that informed and inspired CC2020's adoption of competency [1]–[4].

5400

D.1: Competency in Computing Baccalaureate Education

5401

5402

5403

5404

5405

5406

5407

5408

5409

5410

5411

5412

5413

The genesis of CC2020's commitment to a competency-based orientation for baccalaureate education is rooted in the specific subject matter of computing. But computing knowledge alone has never been the limit of the preparation appropriate for computing graduates. It is only one part, a significant and crucial part, but not the whole of the competency relevant to educated, productive, computing professionals. Whether the terminus of formal education, the conduit to higher academic degrees, or as is the case most often, a gateway into the workforce of computing professionals a baccalaureate education must address the wider world of competency that interconnects with the practice of computing in society. The fundamental tradition of published, computing curriculum guidelines has focused almost exclusively on the subject matter of computing [2]. This is the case even though most if not all baccalaureate computing programs profess to develop practicing professionals who will apply their computing capabilities in a wide variety of workplaces [6]–[8]. To that end, the scope of competency encompassed by computing curricula cannot ignore capabilities that extend far beyond an emphasis on technical computing knowledge.

5414

5415

5416

5417

5418

5419

5420

When leveraged like learning objectives, well-modeled competencies provide a richer language for expressing the goals of a learning experience. Competency modeling provides the ability to more clearly articulate the connection of learning experiences in language that is better connected to both the expectations of graduates and the broader goals of their education [1], [2]. Other advantages of competency modeling include that it connects knowledge and skills as they are expected to be observed in practical tasks. Another is the opportunity to enfold non-technical knowledge and/or skills as the goal or outcome of an educational experience, and that like learning outcomes, competencies should be observed at some level of skill preferably with a relevant performance of task.

5421
5422 This appendix presents the theory and structure of the CC2020’s model of competency that we believe forms a
5423 foundation for adoption across the computing disciplines and their foundational educational enterprise. While
5424 CC2020’s mission ends at conceptually and structurally defining competency in service to computing, it is our
5425 conclusion that this model of competency should underpin other areas of professional capability that are inexorably
5426 integral to educating computing professionals. While we shall not attempt to demonstrate competencies in this alliance,
5427 it is appropriate to scan the knowledge among these that is worth curriculum designers’ commitment. The following
5428 attempts an illustrative scan of the knowledge areas appropriate in a baccalaureate program that results in an educated
5429 computing professional.

5430
5431
5432

5433 **D.2: The CC2020 Definition of Competency**

5434
5435 CC2020’s definition of competency has evolved from numerous models for competency developed and applied in
5436 different educational frameworks. A useful overview of competency occurs in the Harvard University Competency
5437 Dictionary [Har2] which offers the following explanation:

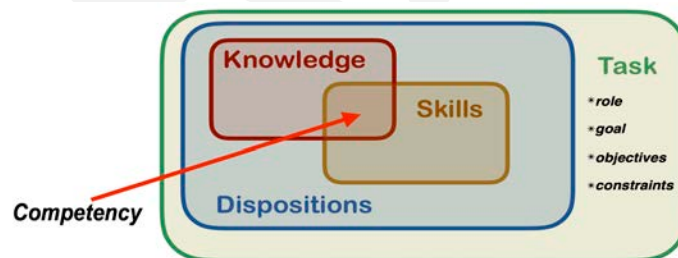
5438 “Competencies, in the most general terms, are “things” that an individual must demonstrate to be effective in a job, role,
5439 function, task, or duty. These “things” include job-relevant behavior (what a person says or does that results in good or
5440 poor performance), motivation (how a person feels about a job, organization, or geographic location), and technical
5441 knowledge/skills (what a person knows/demonstrates regarding facts, technologies, a profession, procedures, a job, an
5442 organization, etc.)”

5443
5444 CC2020 articulates a notion of competency as a practical educational goal [1]–[4] that refines the Knowledge-Skill-
5445 Disposition (K-S-D) framework popularized in the IT 2017 Curriculum Report [5]. While the knowledge dimensions
5446 of computing have been extensively explored in the various computing curricula, what is meant by skill and disposition
5447 have had significantly less focus. Extending previous work, we specify competency as composed of K-S-D dimensions
5448 observed within the performance of a task, T.

5449
5450
$$\text{Competency} = [\text{Knowledge} + \text{Skills} + \text{Dispositions}] \text{ in Task}$$

5451
5452 A competency specification enumerates knowledge, skills, and dispositions that are observable in the accomplishment
5453 of a task, a task that prescribes purpose within a work context [1]. Figure D.1 provides a diagrammatic illustration of
5454 the conceptual structure of competency.

5455



5456
5457
5458 Figure D.1. Conceptual Structure of Competency

5459 These four components that structure the competency specification have the following meanings.

5460
5461 **Knowledge** is the “know-what” dimension of competency that can be understood as factual. An element of knowledge
5462 designates a core concept essential to a competency. This dimension reflects the enumerated subject matter that
5463 teachers catalog as topics in their syllabi, departments distribute and balance among the courses they develop in an
5464 academic program, accreditation organizations stipulate in their accreditation criteria, and employers identify in job
5465 descriptions of their workers. Traditionally, curriculum guidelines for computing education have been predominated
5466 by the designation of knowledge elements composed of facts based upon scientific derivation or proof.

5467
5468 **Skills** refer to the capability and strategy for applying “know-what” to perform a task in context. Competency is
5469 realized when “know-what” knowledge is applied in action to accomplish a task, hence in application, skills express

5470 “know-how.” Skills are most often developed over time and with practice. Consequently, skill development often
5471 requires engagement in a progressive hierarchy of higher-order cognitive process. CC2020’s definition of competency
5472 has adopted Bloom’s levels of cognitive process [14] to specify the degree of skill expected in successful task
5473 accomplishment.
5474

5475 The skills dimension of competency is often assessed indirectly, through observation of the process or quality of work
5476 produced. The activation of “know-what” animated by “know-how” fuses the knowledge and skills dimensions. For
5477 that reason, the usefulness of any element of knowledge in a competency specification is only understandable when
5478 applied at a level of skillfulness, e.g., specified or observed as a level of Bloom’s cognitive process. Therefore, an
5479 element of knowledge and the level of skill with which it is applied are necessarily and naturally conjoined as paired
5480 in the specification of a competency. In this way the CC2020 competency model realizes a performance-based
5481 epistemology that animates an element of knowledge in achieving a task.
5482

5483 **Dispositions** frame the “know-why” dimension of competency and prescribe a requisite character or quality in task
5484 performance. Dispositions shape the discernment of skillful engagement of “know-what” and “know-how.” Specific
5485 to the task at hand, dispositions exert a moderating or controlling influence on a practitioner’s choices by proposing
5486 or projecting a desirable quality onto the outcome. How dispositions moderate knowledge and skill could be thought
5487 of as the “extent that it accounts for the relation between the predictor and the criterion” [21] in that dispositions
5488 connect the ‘better’ or ‘correct’ application of knowledge and skill to the context in which they are applied. For
5489 example, dispositions moderate a practitioner’s capabilities to discern a task as “*professionally accomplished*” rather
5490 than simply “*completed*.” In this sense, dispositions are able to reflect the professional values behind a competency.
5491

5492 Dispositions characterize socio-emotional tendencies, predilections and attitudes that characterize the inclination to
5493 carry out tasks and the sensitivity to know when and how to engage in those tasks [Per1]. Hence, dispositions denote
5494 the values and motivation that guide applying knowledge while designating the quality of knowing commensurate
5495 with a standard of professional performance. “Know-why” exhibits as enacted values and because of the difficulty of
5496 assessing values and intention, disposition is typically assessed indirectly, through the observance of patterns of
5497 behavior or reflective practice.
5498

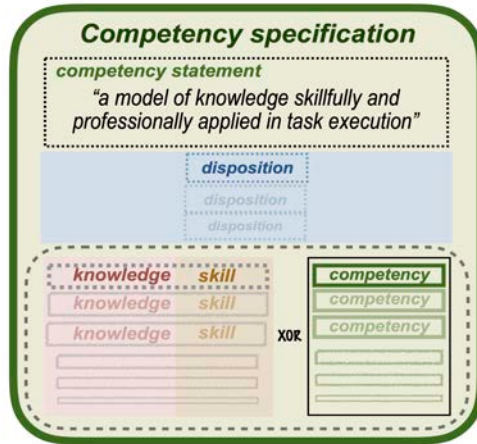
5499 **Task** is the construct that frames the skilled application of knowledge and makes dispositions concrete. Task expressed
5500 as a colloquial prose statement provides the setting to manifest dispositions, where individuals moderate their choices,
5501 actions, and effort necessary to pursue and succeed in an efficient and effective manner. In this sense, task enfolds the
5502 purposeful context of competency, exposing the integral nature of knowledge, skills, and dispositions. To this end,
5503 task definition stipulates pragmatic engagement that reflects professional practice relevant to the particular vision for
5504 the program graduates. In this sense, task descriptions provide an explicit context for the program to develop pedagogy
5505 and graduates to demonstrate competency as a computing professional.
5506

5507
5508

5509 **D.3: The Anatomy of Competency Specification**

5510

5511 An effective specification of competency is a synthesis of 1) a colloquial, prose competency statement that sets out a
5512 task and 2) the component structure of constituent K, S, and D elements necessary to succeed in that task. The graphical
5513 syntax of competency specification is illustrated in Figures D.2, D.3, D.4, and D.5.
5514



5515

Figure D.2. Conceptual Structure of a Competency Specification

CC2020's definition of competency that integrates knowledge, skills, dispositions in task establishes a framework designed to comprehensively describe criteria to support both understanding curricular subject matter in pedagogy and the requisites of task performance in the workplace from which the subject matter derives. A competency statement is a natural language expression of the competency that is more approachable and understandable to the general audience of curricula while the more explicitly expressed component structure facilitates audit and analysis. Figure D.2 illustrates the relationship between a natural language (free-form) competency statement and the component structured representation of knowledge, skills, and dispositions.

In their most simple form, a singular (atomic) competency specification might address the goals for a solitary job function or curricular element constructed from a suitable competency statement and K, S, and D components [1] as per Figure D.3. That atomic competency might then be assimilated as a constituent of a more complex (composite) competency as per Figure D.4. Composite competency specifications unfold as tree structures with branch and leaf nodes. Figure D.5 illustrates this situation where a composite competency specification (C) may combine both atomic (A) and other composite (C) competencies. Competency specifications are normally considered in the aggregate as they are often used to formulate various configurations of educated ability: job descriptions, plans of study, academic degrees, training certifications, professional accreditations and licensure, and standards of performance evaluation. In this sense, more complex competencies are modeled as composites of other constituent, supporting competencies [2].

5516

5517



5518

5519

5520

Figure D.3. Atomic Competency Specification: (A)

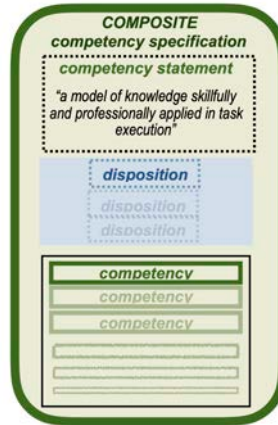


Figure D.4. Composite Competency Node: (C)

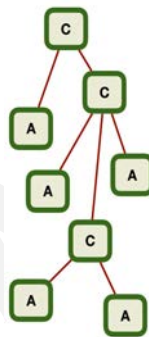


Figure D.5. Competency Tree of Atomic (A) and Composite (C) nodes

The sections that follow we survey the anatomy of a competency specification: the competency statement, the knowledge component (*knowing what*), the skills component (*knowing how*), and the dispositions component (*knowing why*).

D.3.1: The Competency Statement’s Role in a Competency Specification

As a whole, a competency specification expresses a model of knowledge that is skillfully and professionally applied in some task execution. The competency statement of a competency specification is a colloquial expression that succinctly conveys the pertinent ability goals to be attained through a course of study or the capabilities relevant to successfully performing a task in the workplace. The competency statement’s free-form represents the competency in terms that are familiar and comprehensible to a wide audience, typically using a vocabulary familiar to, and that resonates with, the audience. The competency statement is then structurally augmented and amplified in the enumeration of knowledge, skills, and dispositions that complete the competency specification.

While the natural language of the competency statement favors a public audience, the competency component structure is more formal as it enumerates the components, e.g., knowledge elements demonstrated at a skill level and moderating dispositions determined necessary to demonstrate the competency in task. This structural enumeration of components is essential for automating comparative analyses and visualization of curricula. Having both the free-form of the competency statement alongside the more formal component-specific enumeration corroborates that the two perspectives align. Any divergence perceived in these perspectives would suggest the need for a closer reflection upon the usefulness of one or both representations.

D.3.2: Knowledge, “Knowing What,” as a Component of Competency

5553 A single competency expresses knowledge skillfully and professionally applied in some task execution; its vocabulary
5554 will often draw from other implied or stated competencies, at some skill level, reflected through some task that
5555 contextualizes what is intended. The key aspect is expression: how to express the components in meaningful ways,
5556 what knowledge is applied, the expectation of how it is skillfully applied, and what dispositions should be
5557 demonstrated along with a successful task.

5558
5559 The richest and most expressive aspect is the knowledge that can and should be skillfully applied. We present three
5560 perspectives on computing knowledge here. They each play a role in the expectations of computing graduates and
5561 practicing professionals. Computing graduates are normatively expected to skillfully apply computing disciplinary
5562 knowledge (relevant to their academic program), foundational knowledge consistent with baccalaureate education,
5563 and lastly professional knowledge relevant to how graduates operate as professionals. The identification of some
5564 knowledge areas as ‘disciplinary’, some as ‘foundational’ and others as ‘professional’ may be arbitrary but, in the end,
5565 what is needed is an understandable vocabulary useful to the audience for clear and consistent competency statements.
5566

5567 **D.3.2.1: Computing Disciplinary Knowledge**

5568

5569 The encyclopedia of computing knowledge that has accumulated with the efforts of Ka-Ku-Lo over the last half
5570 century is a rich foundation upon which to develop computing competency catalogs for the various subdisciplines of
5571 computing education. These de facto concepts of disciplinary competency are in common use but require formulation
5572 to facilitate interoperability and reusability among the stakeholders both academic and industrial. Employers
5573 frequently identify specific technologies or general knowledge areas (e.g. networking, cloud computing, systems
5574 analysis, and database). As a foundation across the Computing Curricula series, disciplinary knowledge is sometimes
5575 differently labeled or described among computing sub-disciplines (e.g. computer science, information systems).
5576 CC2020’s efforts to promote competency as an overarching framework to describe computing’s role in the classroom
5577 and the workplace presents an opportunity to normalize the vocabulary for describing computing competency. A
5578 normalized vocabulary based upon existing Ka-Ku-Lo-based curriculum specifications can clarify the terms by which
5579 educators identify disciplinary knowledge and its skillful application.

5580
5581 Table D.1 presents a representative summary of computing knowledge areas, extracted from the computing
5582 disciplinary documents published since CC2005. While the table is incomplete, what it provides is a sample high-
5583 level vocabulary for computing knowledge rooted in the collective wisdom of the different computing communities.

5584

5585 Table D.1 Representative Summary of Computing Knowledge Areas

Categorization	Computing Knowledge Area
1. Humans and Organizations	K(C-1.1) Social Issues and Professional Practice
	K(C-1.2) User Experience Design
	K(C-1.3) Security Policy and Management
	K(C-1.4) IS Management and Leadership
	K(C-1.5) Enterprise Architecture
	K(C-1.6) Project Management
2. Systems Modeling	K(C-2.1) Security Issues and Principles
	K(C-2.2) Systems Analysis & Design
	K(C-2.3) Requirements Analysis and Specification
	K(C-2.4) Data and Information Management
3. Software Systems Architecture	K(C-3.1) Virtual Systems and Services
	K(C-3.2) Embedded Systems
	K(C-3.3) Integrated Systems Technology
	K(C-3.4) Intelligent Systems (AI)
	K(C-3.5) Internet of Things
	K(C-3.6) Computer Networks
	K(C-3.7) Platform Technologies
	K(C-3.8) Parallel and Distributed Computing
	K(C-3.9) Security Technology and Implementation
4. Software Development	K(C-4.1) Software Quality
	K(C-4.2) Software Verification and Validation
	K(C-4.3) Software Process
	K(C-4.4) Software Design
	K(C-4.5) Software Modeling and Analysis
	K(C-4.6) Platform-Based Development
	K(C-4.7) Software Development and Management
5. Software Fundamentals	K(C-5.1) Graphics and Visualization
	K(C-5.2) Operating Systems
	K(C-5.3) Algorithms and Complexity
	K(C-5.4) Programming Languages
	K(C-5.5) Software Development Fundamentals
	K(C-5.6) Systems Fundamentals
6. Hardware	K(C-6.1) Architecture and Organization
	K(C-6.2) Digital Design
	K(C-6.3) Circuits and Electronics
	K(C-6.4) Signal Processing

5586
5587
5588
5589
5590
5591
5592
5593
5594
5595
5596

This summary of computing knowledge areas represents a well understood and consistent vocabulary with which we will present computing competency statements and their composition at a very high level of abstraction as illustrations of plausible competency specifications. For reasons that will become clear later in the illustration of competency visualization we order the computing knowledge areas following the semiotic framework developed by Stamper et al. that explicates the expression and transmission of ideas, knowledge, and meaning through human communications [26,27,28]. This ordering offers an ordered arrangement of elements for locating in a cartesian space.

Table D.2 Semiotic Ladder

Semiotic Ladder	Semiotic Layer Description
Social World	Beliefs, expectations, functions, commitments, contracts, law, culture
Pragmatics	Intensions, communications, conversations, negotiations
Semantics	Meanings, propositions, validity, truth, signification, denotations
Syntactics	Formal structure, language, logic, data, records, deduction, software, files
Empirics	Pattern, variety, noise, entropy, channel capacity, redundancy, efficiency, codes
Physical	Signals, traces, physical distinctions, hardware, component density, speed, economics

5597
5598

5599 Although Table D.1 summarizes areas of computing knowledge gleaned and synthesized from the six established
5600 computing curricula (e.g., [5], [14], [15]), this vocabulary does not address many knowledge areas integral to
5601 computing practice. It lacks vocabulary for describing knowledge of a foundational and/or professional nature
5602 common to many (if not all) computing disciplines. Neither does Table D.1 address functional areas of business,
5603 science, or government where an understanding of the application context is crucial for effective computing. The
5604 following sections briefly illustrate these areas that should be given competency attention in baccalaureate education.
5605

5606 **D.3.2.2: Professional and Foundational Knowledge**

5607

5608 Computing disciplinary knowledge alone does not suffice to prepare graduates for successful occupations. While
5609 disciplinary knowledge distinguishes computing professionals among professionals, there are many knowledge areas
5610 other than computing that are foundational, that is they are normative in society and the workplace. Foundational
5611 knowledge deserves careful delineation in computing programs as it is integral to comprehending and succeeding in
5612 the full scope of challenges endemic to the computing practitioner.
5613

5614 There are abilities foundational to workplace conduct that are centered in the individual (e.g., basic academic literacy
5615 in: mathematics, physical sciences, language, social sciences, etc. Other typical foundational knowledge includes
5616 effective communication in written, spoken, and presentational mediums; self-management of time, decorum,
5617 protocols; and many others. Although in-depth study in any of these areas may be appropriate in particular programs,
5618 expectations for the application of foundational knowledge requires some stipulation in baccalaureate computing
5619 curricula.
5620

5621 Employers are seeking individuals who can apply their knowledge of computing technology in specific, commercial
5622 tasks and with a level of prudence evidencing a professional insight. It is well-reported that there is a burgeoning
5623 demand for technology-savvy job applicants as computing's role in commerce, government, and society continues to
5624 expand. Computing job advertisements are replete with openings for applicants possessing a variety of computing
5625 and/or foundational job know-how. However, it is the applicants' facility to effectively apply their computing
5626 knowledge to employers' needs that often predominates in assessment. Beyond specific mentions of applied
5627 'professional' knowledge, this is also evidenced by the common requirements for years-of-experience as a proxy term
5628 for practical, demonstrated workplace acumen.
5629

5630 Industry managers often agree that professional, not just computing or foundational acumen is a primary criterion for
5631 hiring a computing graduate. For example, computing specialists teaming with other professionals from varied
5632 backgrounds lays at the heart of effective technical projects. The idea of teamwork is the "cooperative or coordinated
5633 effort on the part of a group of persons acting together as a team or in the interests of a common cause" [Dic2]. To be
5634 professional a practitioner must demonstrate an effective exchange of ideas through coherent and intelligible
5635 communication. Working in teams is often a normative part of a computing curriculum. Ideally, effective teamwork
5636 should encompass interdisciplinary opportunities where teams include computing expertise as well as proficiency
5637 gained from other areas of study. While not unique to computing professionals, development and mastery among
5638 certain of these abilities is essential to helping a beginner transition from beginner to professional.

5639
5640

Table D.3. Sample Professional and Foundational Knowledge Areas

K(P-1)	Oral Communication & Presentation
K(P-2)	Written Communication
K(P-3)	Problem Solving and Trouble-Shooting
K(P-4)	Project and Task Organization and Planning
K(P-5)	Collaboration and Teamwork
K(P-6)	Research and Self-Starter/Learner
K(P-7)	Multi-Task Prioritization and Management
K(P-8)	Relationship Management
K(P-9)	Analytical and Critical Thinking
K(P-10)	Time Management
K(P-11)	Quality Assurance / Control
K(P-12)	Mathematics and Statistics
K(P-13)	Ethical and Intercultural Perspectives

5641
5642
5643

Where foundational and professional knowledge areas are relevant in competency description, a consistent vocabulary for foundational and professional areas will be essential. Table D.3 presents a sample vocabulary for foundational and professional knowledge. These are representative terms, not an exhaustive list. Similar to Table D.1, this vocabulary is drawn from IT2017 [16] and MSIS2016 [17], both internationally approved computing curricula. The table posits likely candidate domains of workplace acumen proven to be value-added to a computing graduate’s portfolio.

5649
5650 **D.3.2.3: Application Domain Knowledge**

5651

Professional practice in computing is manifested in an organizational or commercial context. Every computing artefact resides within some social context; that is, serving the intention of an individual, a community, or both. Knowledge of that social context informs the choices the computing practitioner is faced with to be interpreted as appropriate, beneficial, or not. To make appropriate choices a professional must possess both foundational, professional, and application context knowledge that complement or integrate with competency that is otherwise specifically computing. To benefit prospective students, employers, legislators, and the citizen electorate, computing curriculum guidelines should be as explicit as possible about the foundational, professional, and application domain experience promulgated programmatically.

5660

Although computing programs variously focused exclusively on technology for software development (i.e. coding bootcamps and academies) have proliferated over the last decade [23], it should be normative for baccalaureate programs in computing to include requirements for application domain education and experience that informs the professional’s potential in a field of practice. Cultural or societal contexts may also require prescribed education and experience: governmental, not-for-profit, non-profit, domestic, or international.

5666

Application domains common to computing include business [4], medicine, engineering, transportation, entertainment, etc. There are many subdisciplines; some are Computing + x and others are x + Computing where “x’s” position indicates whether “x” is the primary disciplinary focus or it is computing’s application domain. For example, the computing subdiscipline of information systems itself has numerous derivatives, x-IS programs, (e.g. accounting information systems, marketing-IS, finance-IS, medical-IS, ...). Each of these x-IS programs is a discipline in its own right augmented with computing. Any delineated domain of application entails particulars of knowledge, skills, and perhaps, distinctive dispositions instrumental to making informed, astute choices that skillfully apply knowledge in artefact design and engagement.

5675

Each of these example areas of computing knowledge deserve the careful categorization and formulation of practice and learning goals that can be served by authoring relevant competency specifications. In some cases, knowledge

5676
5677

5678 among these areas will be integrated in computing competency specifications specific to goals and objectives of a
 5679 program or sub discipline. Others may be distributed among sibling disciplines in an academic setting or among
 5680 particular professional development activities in the workplace. There is mutual benefit to both academia and industry
 5681 when the articulation of the need and value of these knowledge areas associated with computing competency promote
 5682 discussion and the possibility of inter- and intra- disciplinary normalization.
 5683

5684 The synthesis of knowledge areas provided in Tables D.1 and D.3 is at a very high level of abstraction. These are
 5685 provided here for illustration; in general, practical competency specifications need knowledge stipulated with greater
 5686 detail.
 5687

5688 **D.3.3: Skills, “Knowing How,” as Components of Competency**

5689 Competency descriptions focus on applying knowledge skillfully, *observable knowledge in action*. Writing and
 5690 structuring competency statements is significantly simplified by recognizing computing skills as normatively
 5691 cognitive in nature, rather than psychomotor. This simplification correlates with Bloom’s theory of the Cognitive
 5692 Domain in his taxonomy and permits the adoption of a commonly agreed upon vocabulary in the 2001 revisions to
 5693 Bloom’s taxonomy of educational objectives [13]. This reasoning results in *knowing how* expressed as a knowledge
 5694 component paired with a skill level observable in practice. Table D.4 lists verbs that signify skill level.
 5695
 5696
 5697
 5698

Table D.4. Revised Bloom’s Cognitive Skill list [13]

	B-I Remembering	B-II Understanding	B-III Applying	B-IV. Analyzing	B-V Evaluating	B-VI. Creating
Definitions	Exhibit memory of previously learned materials by recalling facts, terms, basic concepts, and answers.	Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions,	Solve problems to new situations by applying acquired knowledge, facts, techniques and rules in a different way.	Examine and break information into parts by identifying motives or causes. Make inferences and find evidence to support	Present and defend opinions by making judgments about information, validity of ideas, or quality of	Compile information together in a different way by combining elements in a new pattern or proposing alternative
Verbs	Choose, Define, Find, How, Label, List, Match, Name, Omit, Recall, Relate, Select, Show, Spell, Tell, What, When, Where, Which, Who, Why	Classify, Compare, Contrast, Demonstrate, Explain, Extend, Illustrate, Infer, Interpret, Outline, Relate, Rephrase, Show, Summarize, Translate	Apply, Build, Choose, Construct, Develop, Experiment, with, Identify, Interview, Make, use, of, Model, Organize, Plan, Select, Solve, Utilize	Analyze, Assume, Categorize, Classify, Compare, Conclusion, Contrast, Discover, Dissect, Distinguish, Divide, Examine, Function, Inference, Inspect, List, Motive, Relationships, Simplify, Survey, Take part in, Test for, Theme	Agree, Appraise, Assess, Award, Choose, Compare, Conclude, Criteria, Criticize, Decide, Deduct, Defend, Determine, Disprove, Estimate, Evaluate, Explain, Importance, Influence, Interpret, Judge, Justify, Mark, Measure, Opinion, Perceive, Prioritize, Prove, Rate, Recommend, Rule on, Select, Support, Value	Adapt, Build, Change, Choose, Combine, Compile, Compose, Construct, Create, Delete, Design, Develop, Discuss, Elaborate, Estimate, Formulate, Happen, Imagine, Improve, Invent, Make up, Maximize, Minimize, Modify, Original, Originate, Plan, Predict, Propose, Solution, Solve, Suppose, Test, Theory

5699 This understanding of skills expressed as *observable knowledge in action* necessitates an expression of a context
 5700 for observing the knowledge skillfully applied, portraying a purpose to be fulfilled. In operational terms K+S
 5701 is normally observed in accomplishing a task, where task conveys a purpose that motivates applying the knowledge
 5702 with a particular level of skill. The task serves as both the circumstance within which to observe K+S as
 5703 operationalized by the actor, but also to animate the dispositions, D, that moderate the actor’s performance to
 5704 complete a task and to what measure of success.
 5705
 5706
 5707
 5708

5709 **D.3.4: Dispositions, “Knowing Why,” as a Component of Competency**

5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764

The meta-language of competency, “knowing what,” “knowing how,” and “knowing why,” crisscrosses domains of scientific fact, practiced behavior, and cultural norms. Scientific (technically-rational) fact and practiced behavior lend themselves to a categorical assessment: true or false, present or absent, consistent or inconsistent, it works or it doesn’t. Dispositions enfold intellectual, social, and moral predilections or tendencies that influence behaviors that do not lend themselves as easily to a categorical assessment. These predilections reflect value judgements that are not amenable to scientific proof. Values may differ or be held differently among individuals or cultures. And, value judgements are also often mutable over time - affected by the experience of practice!

In the broader cultural domains, dispositions may assert positions regarding virtually any desirable quality that motivates human behavior (e.g. ethics, integrity, empathy, accountability, honesty, respectfulness). But in the end, the import of disposition is ultimately realized through individual persons applying their knowledge and skills, through their behavior – individuals leveraging their intellect through responsible decisions and actions [20]. In this applied context, dispositions incline enacted virtues that reflect the values expressed by the actor through their choices, decisions, and actions [24].

An important consideration in the specification of the disposition is the separation of the skilled application of professional or foundational knowledge (such as communication clarity, leadership, creative thinking, and time management, which include significant components from the "know-how" category) from dispositions ("know why"). For example, the development of the disposition categories provided in Table D.5, this was accomplished by analyzing research on job descriptions [18], [19] and other related sources [20] and then removing those statements which were identifiable as a K-S pair, or appear as a competency combining K-S, D, and other components. Hence something as complex as leadership is best modelled as a competency because it has implied K-S pairs and one or more dispositions. Other items may well be a collection of K-S pairs which then would be constituent parts for a competency.

Table D.5 offers a short list of prospective dispositions derived from the literature to round out the knowledge, skills, dispositions components of competency. Disposition as an intrinsic component of competency represents the opportunity to clearly express institutional and programmatic values expected in a graduate’s work. Dispositional expectations enrich the description/assessment of competency and/or the related pedagogy. Ascribing a disposition to a competency indicates a clear commitment to self-reflection and a sober examination of mission, goals, and objectives to reach the clarity that enables its effective integration in curriculum design, the agency of pedagogy, and the character of professionalism.

Disposition is an area that distinctly distinguishes a competency from a learning outcome and is an essential characteristic of a well-structured competency. As such it represents a significant extension in the expressiveness of learning goals and adds language common to professional expectations. However, when used in free form, such terms may easily become vague or difficult to interpret. This is where the specification of a competency, that is the combination of the free-form text with its constituent K+S+D in T framing becomes more valuable. The competency statement is prose that succinctly conveys the essential intention of curricular details, while the structured enumeration of the K-S pairs and D elements conveys intention in action.

5765

Table D.5. Prospective Dispositions [18]-[20]

Disposition	Elaboration
D-1 Proactive	With Initiative / Self-Starter Shows independence. Ability to assess and start activities independently without needing to be told what to do. Willing to take the lead, not waiting for others to start activities or wait for instructions.
D-2 Self-Directed	Self-motivated / Self-Directed Demonstrates determination to sustain efforts to continue tasks. Direction from others is not required to continue a task toward its desired ends.
D-3 Passionate	With Passion / Conviction Strongly committed to and enthusiastic about the realization of the task or goal. Makes the compelling case for the success and benefits of task, project, team or means of achieving goals.
D-4 Purpose-Driven	Purposefully engaged / Purposefulness Goal-directed, intentionally acting and committed to achieve organizational and project goals. Reflects an attitude towards the organizational goals served by decisions, work or work products. e.g., Business acumen.
D-5 Professional	With Professionalism / Work ethic. Reflecting qualities connected with trained and skilled people: Acting honestly, with integrity, commitment, determination and dedication to what is required to achieve a task.
D-6 Responsible	With Judgement / Discretion / Responsible / Rectitude Reflect on conditions and concerns, then acting according to what is appropriate to the situation. Making responsible assessments and taking actions using professional knowledge, experience, understanding and common sense. E.g., Responsibility, Professional astuteness.
D-7 Adaptable	Adaptable / Flexible / Agile Ability or willingness to adjust approach in response to changing conditions or needs.
D-8 Collaborative	Collaborative / Team Player / Influencing Willingness to work with others; engaging appropriate involvement of other persons and organizations helpful to the task. Striving to be respectful and productive in achieving a common goal.
D-9 Responsive	Responsive / Respectful Reacting quickly and positively. Respecting the timing needs for communication and actions needed to achieve the goals of the work.
D-10 Meticulous	Attentive to Detail Achieves thoroughness and accuracy when accomplishing a task through concern for relevant details.
D-11 Inventive	Exploratory / Inventive Looking beyond simple solutions; Examining alternative ideas and solutions; seeks, produces and integrates appropriate alternative

5766
5767

5768

5769

5770

5771

5772

5773

5774

5775

5776

5777

5778

5779

5780

5781

5782

5783

5784

5785

5786

5787

5788

5789

5790

5791

5792

5793

5794

5795

5796

What follows is an illustration of constructing well-structured competency statements and their specification. The purpose is not so much as to define a particular required or desired competency for all computing graduates, but rather to provide a point of discussion about the difficulties and value of such statements and the details contained in their modeled components. Relying on the vocabulary provided in Table D.1 (albeit at a high level of abstraction), and Tables D.3, D.4, and D.5, these examples provide an plausible illustration of how well-structured competency statements can be specified for a program, curriculum, or job description.

D.4: Structuring Competency Statements for Competency Specification

Competency statements have not been the most common means to express learning goals or outcomes. Properly formulated, competency statements should express clear, relevant, and actionable specifications. As such, they differ from learning outcomes in that they imply or imply all four, (K, S, and D) in T, components. In practice, useful competency statements may only imply some of these components, as their primary purpose is communication, not completeness. In the competency explorations carried out by the Task Force (included in the Appendix), the free-form competency statements collected rarely included all four components. Indeed, many expressions of computing competencies were incomplete, and were only explicit about some but not all components. The downside to incomplete competency statements is that they are less useful for assessment, comparison, or other forms of analysis. Hence, the pairing of the free-form statement with its elaborated specification of K, S, and D serves both purposes and in practice acts as means for assessing consistency. Well-structured statements should imply the structured components and in particular communicate a task context where the competency should be observable.

D.4.1: Developing Competency Statements and Specifications

In formulating a good competency statement, the author/designer of a useful statement is best counseled by contemplating the results of a task execution that describe desired actor behavior in clear, relevant, and actionable ways. The K, S, and D vocabularies in Tables D.1-D.5 provide a sample structure for developing and/or parsing

5797 competency statements. A particular competency statement can have a number of K and S pair components. Similarly,
5798 K-S pairs can be moderated by one or more D labels. This concept follows competency theory that provides for a
5799 hierarchical structure for modelling competencies [2]. In this formulation, a competency can be modeled individually
5800 as a K, S, and D in T but may also serve as a constituent to other competencies.

5801
5802 In practice, competency statements for curricular use should not be limited to structured language such as that provided
5803 in Tables D.1-D.5. Such a restriction limits the expression of the competencies in local contexts. Enumerating the
5804 meaning of a free-form statement with structured language by leveraging the K, S, and D in T structure has significant
5805 value. The additional work of making implied components more explicit makes the competency statement richer both
5806 in expression and meaning: more clear, relevant and actionable.

5807
5808 Modeling a competency statement using structured language improves communication and assessment. For
5809 communication, the process of structuring a competency statement into structured language documents the explicit
5810 meanings and helps to uncover the implicit meanings intended in the statement. For assessment, the different K+S and
5811 D components identified are often assessed in different ways. Identifying and classifying goal components promotes
5812 clarity in assessing individual competency components.

5813
5814 As Chapter 5 elaborates, analyzing a competency statement for its various K, S, and D in T modeled competency
5815 enables the comparison of competency statements [3]. Typically, unstructured competency statements taken from
5816 different computing curricula can be difficult to compare. However, if the constituent parts of the statements can
5817 employ a common structured vocabulary, competencies can be compared and modeled through visualization using
5818 automation.

5819
5820 In curricula the concept of disposition observable in a task presents the opportunity to enhance the comprehension of
5821 knowledge and skills as they related to a computing discipline or academic program. Competency statements offer an
5822 opportunity for students to realize more synthesis in their computing education. Applying relevant dispositions
5823 informs the students' educational experience by providing an approach explicit in purpose to the content they learn.
5824 Consequently, these stakeholders directly benefit from these qualities instilled in computing graduates.

5825

5826

5827 **D.4.2: Elaborating Competency Statements**

5828

5829 To illustrate the process of elaborating the competency statement, the work is to enumerate the knowledge, skills,
5830 dispositions and task elements of the statement. Here we present first an example statement drawn from the
5831 System/Software Engineering domain, followed by another example from the Information Systems domain. These
5832 both are presented as 'atomic' examples as per Figure D.3. These two examples (Figures D.6-D.9) are then leveraged
5833 in an example of a compound competency presented in Figures D.10-D.12.

5834

5835 The goal of these examples is to illustrate how to unpack, in a structured form, phrase decompositions that represent
5836 the explicit and implicit K-S-D-T components of the three different free-form competency statements. These are
5837 mapped onto structure vocabulary and analyzed for completeness. This detailed mapping of a competency statement
5838 serves multiple purposes. To begin with, it very much helps one to understand the completeness of the statement, as
5839 well as the K-S pairs expressed or implied. The completeness of the statement suggests the nature of a contextually
5840 situated example that would have the opportunity of generating multiple and distinctive assessment opportunities. It
5841 also provides a connection to what is expected to be assessed, *e.g.*, not just what the students did, but how they did it;
5842 the quality of both their work and the quality of how that work was accomplished.

5843

5844 The most important aspect of this exercise is the support for the actualization of this competency within this program.
5845 It provides a structured way of expressing what needs to be taught, a framework for determining how best to manage
5846 the learning activities, and clear discussion points for how best to assess different aspects of this competency within
5847 the program. For example, learning modules intended to support developing this competency could be inside of a
5848 single course, or across multiple courses. It could describe a key task within a requirements course, or a project-based
5849 course, or even in a learning exercise at an internship or other setting.

5850

5851

Identify and document system requirements by applying a known requirements elicitation technique in work sessions with stakeholders, using facilitative skills, as a contributing member of a requirements

Figure D.6. Sample Free-form Competency Statement for Systems Requirements

5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886
5887

The natural-language text of Figure D.6 can be parsed into three constituent competency phrases for analysis. The list that follows suggests one examination of the explicit and implicit K-S pairs, as well as the implied context of the statement as a whole.

Leveraging the abstract vocabulary of Tables D.1, D.3 and D.4, this results in the following four sets of mappings:

- (i) “Identify and document system requirements” (somewhat) explicitly expects students to be applying [S(B-III)] *Requirements Analysis and Specification* [K(C-2.3)] knowledge and understanding. It also implies students to demonstrate applying [S(B-III)] appropriate *Written Communication* [K(P-2)] knowledge and skills.
- (ii) “applying a known requirements elicitation technique in work sessions with stakeholders” explicitly expects students to be applying [S(B-III)] *Requirements Analysis and Specification* [K(C-2.3)] knowledge and understanding and implies students to be applying [S(B-III)] *Systems Analysis and Design* [K(C-2.2)] knowledge and understanding.
- (iii) “using facilitative skills, as a contributing member of a requirements team” explicitly expects students to be applying [S(B-III)] *Requirements Analysis and Specification* [K(C-2.3)] knowledge and understanding and to be applying [S(B-III)] *Collaboration and Teamwork* [K(P-5)] knowledge and skills.

Extending this to include the dispositional elements (e.g., Table D.5) implicated adds an additional mapping:

- (iv) In context, this whole statement implies students to demonstrate capability of evaluating [S(B-V)] *Requirements Analysis and Specification* [K(C-2.3)] and Analyzing [S(B-IV)] *Collaboration and Teamwork* (P-5). These behaviors are expected to be moderated by students demonstrating that they are *Purposefully engaged* (D-4), with *Judgement* (D-6) and demonstrating that they are *Collaborative* (D-8).

Lastly, completeness warrants including the task specification that is stated or implied:

- (v) The statement is explicit about having a particular (though unspecified) task (T) in which this work which has.

This example statement in Figure D.6 provides an example of a competency-based approach to describing a possible program or course-level goal or outcome. The statement appears complete, in that it reasonably captures all four K-S-D-T elements of a useful competency statement at a level of abstraction consistent with the vocabulary of interest. Note that with a more detailed vocabulary (not presented), each of the K elements could be expanded into other constituent competencies. Based on this level of analysis, the statement expands into an atomic competency as shown in Figure D.7.



Figure D.7. Example Systems Requirements Competency Specification

5888
5889
5890
5891
5892
5893
5894

This competency statement focuses on a central aspect of systems analysis. If employed in a course or program, it sets up the opportunity for (and challenges) the educator teaching a systems and/or software requirements unit (or course)

5895 to set up a learning situation whereby not only will the students be challenged to engage in the context, but also that
5896 the instructor can observe student behavior for assessing to what extent the students demonstrate the K-S-D
5897 components. The relationships to program definition and assessment are explored in more detail in Chapter 5.

5898
5899 To illustrate this statement/analysis process for a different domain in computing, Figure D.8 presents a second
5900 example from the Information Systems (IS) domain related to the area of Enterprise Architecture.
5901

Analyze an enterprise architecture against an organizational business model. Consider several appropriate cloud service approaches. Substantiate the recommendation with cost-benefit details to present to management decision-makers.

Figure D.8. Cloud Services in Enterprise Architecture

5902
5903
5904 Leveraging the abstract vocabulary of Tables D.1, D.3 and D.4, this results in the following four sets of mappings:

- 5905 (i) “Analyze an enterprise architecture against an organizational business model.” It explicitly expects students
5906 to be analyzing [S(B-IV)] *Enterprise Architecture* [K(C-1.5)] leveraging that knowledge and understanding.
5907 This expectation also leverages understanding [S(B-II)] of *IS Management and Leadership* [K(C-1.4)].
5908
5909 (ii) “Consider several appropriate cloud service approaches” explicitly expects students to analyze [S(B-IV)]
5910 *Virtual Systems and Services* [K(C-3.1)] knowledge and understanding.
5911
5912 (iii) “Substantiate the recommendation with cost-benefit details to present to management decision-makers.”
5913 explicitly expects students to be evaluating [S(B-V)] leveraging *IS Management and Leadership* [K(C-1.4)]
5914 knowledge and understanding, This work includes examining and breaking down the details, *i.e.* analyzing
5915 [S(B-IV)] *Research and self-starter/learner* [K(P-6)]. This information is then communicated by applying
5916 [S(B-III)] *Oral communication and presentation* [K(P-1)] knowledge and skills and applying [S(B-III)]
5917 *Written communication* [K(P-2)] knowledge and skills.

5918
5919 Extending this to include the dispositional elements (e.g., Table D.5) implicated adds an additional mapping:
5920

- 5921 (iv) As per items (i) and (iii), this statement implies students to demonstrate the capability of analyzing [S(B-IV)]
5922 *Enterprise Architecture* [K(C-1.5)] and evaluating [S(B-V)] leveraging *IS Management and Leadership* [K(C-
5923 1.4)] knowledge and understanding. In the learning context, these behaviors are expected to be moderated by
5924 students demonstrating that they are *Proactive* (D-1) in seeking out the information that is needed in a *Self-
5925 Directed* (D-2) manner. The purpose of the presentation suitable for management is that they demonstrate
5926 *Professional* (D-5) attitudes and behavior.
5927
5928 (v) Lastly, completeness warrants including the task specification (T) which was stated, but also left open to
5929 different settings in the application of enterprise architecture.

5930
5931 Similar to the previous example, this presents a reasonably complete statement, given the abstract vocabulary
5932 employed. Figure D.9 illustrates this statement, coupled with its mapping.
5933

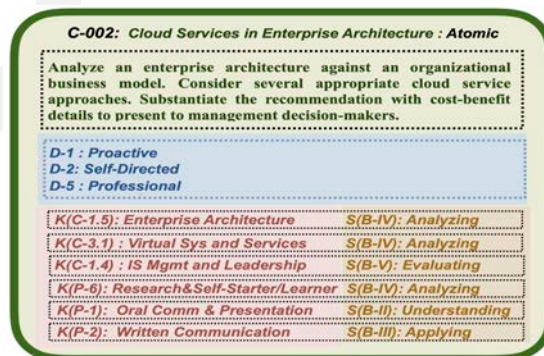


Figure D.9. Cloud Services in Enterprise Architecture

5934
5935

5936
5937
5938
5939
5940
5941

Figures D.6-D.9 present atomic competencies (e.g., those not dependent upon previously-stated/modeled competencies). Figure D.10 presents a statement that can be modeled as being dependent on these statements. This example relates to a competency related to the topic of cloud services and reflects the Information Systems domain.

Propose an enterprise architecture based on the organizational business model and consistent with the mission and objectives of the organization. The architecture should propose appropriate leading edge technologies consistent with the organizational requirements.

5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964

Figure D.10. Cloud Services in Enterprise Architecture

Leveraging the abstract vocabulary of Tables D.1, D.3 and D.4, and the competencies of Figure D.10 results in the following four sets of mappings:

- (i) “Propose an enterprise architecture based on the organizational business model” embraces the enterprise architecture competency displayed in Figure D.8 and modeled as C-002 in Figure D.9. The phrases “consistent with the mission and objectives of the organization” are normatively a part of effective business modeling, so are considered within this competency.
- (ii) “Propose appropriate... technologies consistent with the organizational requirements” explicitly leverages the systems requirements competency displayed in Figure D.6 and modeled as C-001 in Figure D.7.

Extending this to include the dispositional elements (e.g., Table D.5) implicated adds an additional mapping:

- (iv) In the learning context, these behaviors are expected to be moderated by students demonstrating that they are *Meticulous* (D-10) in seeking out the information that is needed in an *Inventive* (D-10) manner.
- (v) Lastly, completeness warrants including the task specification (T) which was stated, but also left open to different settings in the application of enterprise architecture.

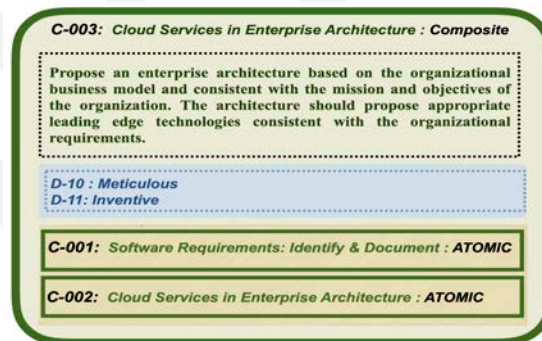


Figure D.11. Cloud Services in Enterprise Architecture

5965
5966
5967
5968

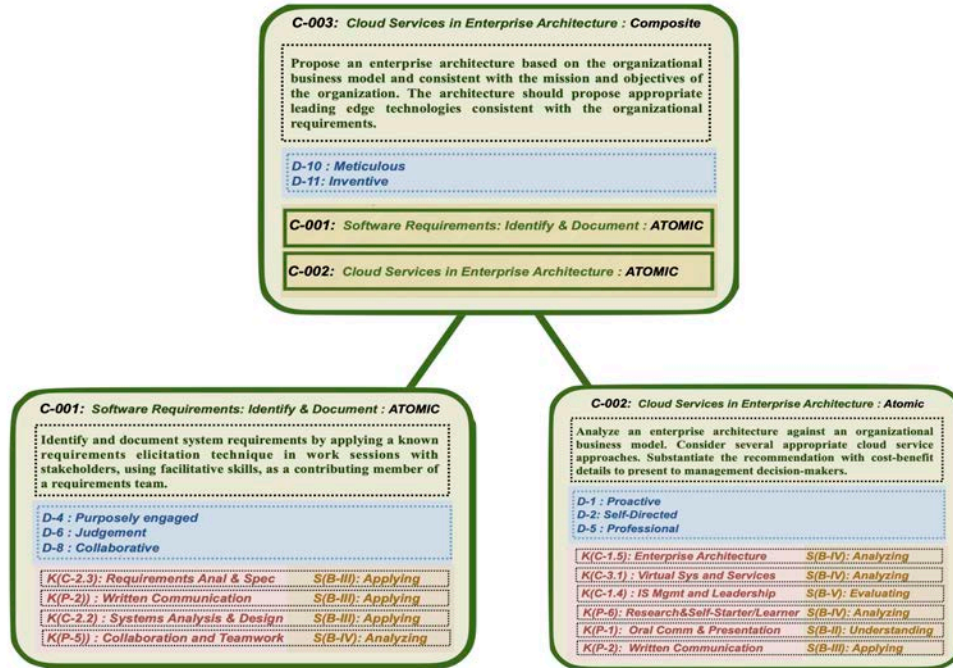


Figure D.12. Composite Competency Specification

5969
5970
5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003

D.5: Competency in Computing Education

Competencies as described here, are broad statements that better capture either aspirational or demonstrable goals of a degree, course, learning module, etc. [2] Competencies serve as outcome expectations at the level of a degree program (or other aggregate structure toward which students are working; for the sake of simplicity, we use the term program). Learning experiences are courses, modules, or other similar sets of learning activities that collectively constitute a program. Each learning experience leads to a set of learning outcomes, which collectively enable students to attain the required competencies. A curriculum specifies, at a minimum, the topics, pedagogical approaches, and learning outcomes for each of the learning experiences.

When described in other contexts, competency statements are often binary; either one has that competency or one does not. However, this is not the nature of competencies as described here or as are useful for computing education. Instead, like other types of learning outcomes, they are demonstrated at a particular skill level(s). The purpose of the K, S, and D in T formulation is to provide demonstrable goals that can be observed and in an educational context assessed for how well the student demonstrates the achievement of those goals. Beyond traditional learning outcomes, competency statements encompass the dispositional, ‘enacted value’ aspects of learning. Consequently, competency specifications at any level of education both inform pedagogy and situates assessment.

One of the most critical aspects of competency development is the use of consistent vocabulary. The K, S, and D in T framework is useful only to the extent that the terminology used for any of the components has a consistent meaning for its constituents. Historically, much of the terminology used for describing competencies is highly context-dependent or ambiguous [2,11]. Consequently the need for both the authors and readers to have useful, juried vocabulary particularly for the terminology used to describe Knowledge, Skills, Dispositions is essential both to communicate and to comprehend the meaning(s) implied. For example, the vocabulary of Tables D.1 and D.3 are very abstract, whereas many of the Ka-Ku knowledge hierarchies developed since CC2005 are relatively detailed. With more detailed vocabulary, more detailed competencies can be described.

This competency-based approach is providing a new mechanism for working with/describing curricula reflecting what graduates can do, vs. just what they know. One reason to make a transition from the Ka-Ku-Lo model to competency-based learning is the skills gap that exists between the needs of industry and the capability of graduates from computing

6004 programs. In particular, the competency vocabulary leveraged in Tables D.3 and D.4 are all drawn from vocabulary
6005 used in computing job descriptions. This connection to the workplace, facilitated by a competency-based approach is
6006 important. For a typical university, an overwhelming majority of computing graduates enter the workplace directly.
6007 While universities are not training grounds for industry, often there is reported a disconnect between the products
6008 produced (computing graduates) by universities and the needs of industry [8,9]. Specifying program expectations as
6009 competencies will be more easily understandable by the employer partners of computing programs as well as graduates
6010 and other constituents. These themes (and others) are developed in more detail in Chapter 5.

6011
6012 The CC2020 project has embraced competency as an underlying theme of its activities and as a principal component
6013 of this report. The task force believes that every career path in computing, whether industrial, or academic, or
6014 government, or any other career demands an intentional level of performance in applicable competencies. It observes
6015 that knowledge is only one component of competency. Adopting competency as the foundational model on which to
6016 base academic program designs is a more effective bridge between the deliverables achievable by academia and their
6017 consumption by the society at large. Thus, it is logical that this report should foster competency-based learning instead
6018 of knowledge-based learning. When used intentionally, this approach ensures that graduates of computing programs
6019 have a better preparation to be effective in their career paths.

6020
6021
6022

6023 **D.6: Competency in Future Curricular Guidelines**

6024

6025 The CC2020 task force is committed to the use of competency in current and future computing curricular reports and
6026 recommends continued development of competency statements. The competency-based approach makes it possible
6027 to compare computing disciplines and facilitate detailed comparisons. Competency implies attaining a level of
6028 professional excellence and performance that goes beyond having only knowledge in a field. These extensions include
6029 technical and professional skills to function in the workplace at an acceptable level of performance. It is important to
6030 employ the competency-based approach in the development of future curricular guidelines within a common frame of
6031 reference. This is a major theme of Chapter 5, with the development of a means of collecting and comparing both the
6032 results of international curriculum guidelines, but also models that come not from just model curricula, but actual
6033 curricula around the world.

6034

6035 This report also assumes that specifying the knowledge, skills and dispositions desirable for computing graduates is
6036 beyond its scope. Specifying the detailed competencies is the responsibility of discipline-specific curricular reports
6037 and, more particularly, individual computing programs themselves.

6038

6039 Given that most graduates of computing programs enter the workplace, it is very important that all computing
6040 programs prepare their graduates properly so they can perform as professionals and engage in productive careers.
6041 While the CC2020 project can only suggest its beliefs, its task force is confident that computing organizations and
6042 programs worldwide will heed the suggestions made in this report and transform their activities where competency
6043 becomes central to their future undertakings. In today's world, graduates must be able to perform in the workplace
6044 with the technical and professional knowledge, skills and better opportunities to develop and understand the
6045 dispositions that help make their knowledge and skills effective in the workplace.

6046

6047 As the work on CC2020 has progressed, so, also, have other international curriculum development efforts, including
6048 IS2020. A cyber security project also constituted an ongoing curricular effort in parallel with the CC2020 project.
6049 ACM published this report, called CSEC2017 [Acm08], in December of 2017. Some projects approaching initiation
6050 include information systems, with a report pending in 2021. Naturally, other curricular updates include software
6051 engineering, computer science, and computer engineering. The task force is hopeful that all future curricular endeavors
6052 adopt the competency-based approach.

6053

6054 Competency's task component suggests opportunities for workplace-bound learning experiences that engage authentic
6055 problems with industrial tools; that encourage employers' active involvement supporting professional development
6056 through internships, co-op programs, and expert mentorship. Dispositions materialized through task encourage
6057 promoting an appreciation for diverse teams, for collaborative norms in project-based activities, and a deliberate and
6058 critical reflective practice fostering effective decision-making and continuous learning.

6059

6060 The competency model for computing education presented herein frames the pattern and philosophy for future
6061 curriculum guidelines. While at the same time a careful consideration has been applied to the interoperability of
6062 competency-based curricular descriptions. The model facilitates the analysis of curricular specifications through
6063 comparison to identify the overlap or omissions that may exist between curricula. The potential to represent curricula,
6064 curriculum fragments, and job descriptions in competency form facilitates a wide variety of study and analysis. The
6065 mission of CC2020 addressed by the development of the competency model for computing education does not enfold
6066 the authoring of computing curricula that is a task that must be undertaken by curriculum guideline endeavors followed
6067 and integrated with the engagement of educational institutions in the shaping and evolution of their computing
6068 programs. And perhaps the greatest incentive supporting the adoption of competency-based curricular specification is
6069 the opportunity for a more efficient and effective partnership between academia and industry in addressing the shared
6070 goals of advancing the benefits of computing to society.
6071
6072
6073

6074 **D.7: Summary**

6075

6076 This appendix deals with the nature of competency – one of the salient features of the CC2020 project. It presented
6077 several competency statements to exemplify the application of the theory. Competency-based curricula are more
6078 expressive in their learning goals, and more easily translated to the language of graduate job descriptions and industry
6079 needs. Recognizing the knowledge-based approaches taken in many computing curricula to date, recent developments
6080 in computing curricula imply that the components of computing curricula should include not just knowledge and skills
6081 but also dispositions, skill levels and typical (maybe “practical”) tasks expected of graduates. We recommend that
6082 future curricular reports adopt this competency approach to describing computing curricula and expand the theoretical
6083 foundation upon which curricula are designed.
6084
6085
6086

6087 **D.8: References for Appendix D**

6088
6089

- 6090 [1] L. Waguespack et al., “Adopting Competency Mindful of Professionalism in Baccalaureate Computing Curricula,” in
6091 EDSIGCON 2019, 2019.
- 6092 [2] S. Frezza et al., “Modelling competencies for computing education beyond 2020: a research based approach to defining
6093 competencies in the computing disciplines,” pp. 148–174, 2018.
- 6094 [3] S. Takada et al., “Toward Understanding Computing Curricula: Visualization Using Competency within the CC2020
6095 Project,” *Educ. Inf. Technol.*, 2020.
- 6096 [4] H. Topi, “Information Systems in CC2020: Comparing Key Structural Elements of Curriculum Recommendations in
6097 Computing,” in 2017 Proceedings of SIGED: IAIME Conference, 2017.
- 6098 [5] Task Group on Information Technology Curricula, *Information Technology Curricula 2017: Curriculum Guidelines for
6099 Baccalaureate Degree Programs in Information Technology*. Association for Computing Machinery and IEEE Computer
6100 Society, 2017.
- 6101 [6] D. Bairaktarova and A. Woodcock, “Engineering Student’s Ethical Awareness and Behavior: A New Motivational Model,”
6102 *Sci. Eng. Ethics*, vol. 23, no. 4, pp. 1129–1157, 2017.
- 6103 [7] H. Han, “Virtue Ethics, Positive Psychology, and a New Model of Science and Engineering Ethics Education,” *Sci. Eng.
6104 Ethics*, vol. 21, no. 2, pp. 441–460, 2014.
- 6105 [8] A. Radermacher and G. Walia, “Gaps Between Industry Expectations and the Abilities of Graduates,” in *Proceeding of the
6106 44th ACM Technical Symposium on Computer Science Education*, 2013, pp. 525–530.
- 6107 [9] S. Billett, “Realising the educational worth of integrating work experiences in higher education,” *Stud. High. Educ.*, vol. 34,
6108 no. 7, pp. 827–843, Nov. 2009.
- 6109 [10] D. L. Schussler, “Defining Dispositions: Wading Through Murky Waters,” *Teach. Educ.*, vol. 41, no. 4, pp. 251–268, Mar.
6110 2006.
- 6111 [11] D. N. Perkins, E. Jay, and S. Tishman, “Beyond Abilities: A Dispositional Theory of Thinking,” *Merrill. Palmer. Q.*, vol. 39,
6112 no. 1, pp. 1–21, 1993.
- 6113 [12] D. R. Lynch, J. S. Russell, J. C. Evans, and K. G. Sutterer, “Beyond the Cognitive: The Affective Domain, Values, and the
6114 Achievement of the Vision,” *J. Prof. Issues Eng. Educ. Pract.*, vol. 135, no. 1, pp. 47–56, Jan. 2009.
- 6115 [13] L. W. Anderson et al., “A taxonomy for learning, teaching, and assessing: A revision of Bloom’s taxonomy of educational
6116 objectives, abridged edition,” White Plains, NY Longman, 2001.

- 6117 [14] ACM/IEEE Joint Task Force on Computing Curricula, Software Engineering 2014 Curriculum Guidelines for Undergraduate
6118 Degree Programs in Software Engineering. IEEE Computer Society and the Association for Computing Machinery, 2015.
6119 [15] M. Kramer, P. Hubwieser, and T. Brinda, "A Competency Structure Model of Object-Oriented Programming," in 2016
6120 International Conference on Learning and Teaching in Computing and Engineering (LaTICE), 2016, pp. 1–8.
6121 [16] Task Group on Information Technology Curricula, Information Technology Curricula 2017, 2017th-10th- ed. ACM and
6122 IEEE, 2017.
6123 [17] H. Karsten and S. A. Brown, MSIS 2016 Global Competency Model for Graduate Degree Programs in Information Systems.
6124 2016.
6125 [18] J. C. Nwokeji, R. Stachel, and T. Holmes, "Competencies Required for Developing Computer and Information Systems
6126 Curriculum," in Proceedings of the 49th Frontiers in Education Conference (FIE'19), 2019.
6127 [19] T. Clear, "THINKING ISSUES: Meeting employers expectations of devops roles: can dispositions be taught?," Inroads, vol.
6128 8, no. 2, pp. 19–21, 2017.
6129 [20] J. M. Gray, "Virtue Ethics : Examining Influences on the Ethical Commitment of Information System Workers in Trusted
6130 Positions," Ph.D. Thesis, Nova Southeastern University, 2015.
6131 [21] Baron, R. M., & Kenny, D. A. (1986). The moderator-mediator variable distinction in social psychological research:
6132 Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology*, 51, 1173-1182. Retrieved
6133 from <https://psych.wisc.edu/henriques/mediator.html> (current September 12, 2019).
6134 [22] Simon, H. (1996), *The Sciences of the Artificial*, 3ed, M.I.T., Cambridge, MA, p. 119.
6135 [23] Waguespack, L., Babb, J. S., Yates, D. (2018). Triangulating Coding Bootcamps in IS Education: Bootleg Education or
6136 Disruptive Innovation?. *Information Systems Education Journal*, 16(6) pp 48-58. <http://isedj.org/2018-16/> ISSN: 1545-679X.
6137 [24] Annas, Julia, "Intelligent Virtue," Oxford Press, 2011.
6138 [25a] Wiggins, G., McTighe, J., & Ebrary, I. (2005). *Understanding by design* (Expanded second edition). Alexandria, VA:
6139 Association for Supervision and Curriculum Development.
6140 [25b] Wiggins, G., & McTighe, J. (2011). *The Understanding by Design Guide to Creating High-Quality Units*. Alexandria, VA:
6141 Association for Supervision and Curriculum Development.
6142 [26] Liu, K., *Semiotics in Information Systems Engineering*, Cambridge University Press, Cambridge, U.K, 2000.
6143 [27] Stamper, R. K., *Information in Business and Administrative Systems*, John Wiley and Sons, New York, NY, 1973.
6144 [28] Stamper, R. K., Althous, K. and Backhouse, J., "MEASUR, Method for Eliciting, Analyzing, and Specifying User
6145 Requirements," In Olle, T. W., Verrijn-Stuart, A. A. and Bhabuts, L., (eds.), *Computerized Assistance During the
6146 Information Life Cycle*, Elsevier, Amsterdam, The Netherlands, 1988.
6147
6148
6149
6150

6151

6152 **Appendix E: From Competencies to Curricula**

6153

6154 Chapter 4 introduced and defined the competency concept and briefly discussed how competencies are related to
6155 graduate outcome expectations for degree programs. Underlying this model is an assumption that graduate
6156 competency specifications provide a foundation for designing learning experiences that are a foundation to the
6157 graduates' ability to execute relevant tasks as computing professionals. In this appendix, we will discuss various
6158 applications of the competency-based approach. We will outline the stakeholder groups who might benefit from the
6159 use of a competency-based model and a variety of ways in which these stakeholders in the educational ecosystem can
6160 benefit from the competency-based approach and utilize it effectively. In addition, we will describe characteristics of
6161 the processes with which competencies are identified and authored. Furthermore, we will discuss how a curriculum
6162 can be derived from a set of competencies.

6163

6164 The fundamental questions we address in this appendix are: a) What are the most appropriate processes and
6165 information sources for deriving competencies for a specific educational program in a specific context? (Section 5.2)
6166 and b) How can competencies be used to guide curriculum design and revision processes? (Section 5.3.)

6167

6168

6169

6170 **E.1: Competency in Future Curricular Guidelines**

6171

6172 In this section, we will define and discuss several key concepts essential for any practical use of the competency-based
6173 approach: stakeholders, competency targets, and the differences between traditional and a competency-based approach
6174 for specifying computing subdisciplines.

6175

6176

6177 **E.1.1: Stakeholders**

6178

6179 The CC2020 project has identified five groups of stakeholders the members of which may benefit from the
6180 competency-based approach to specifying outcome expectations. These five are as follows:

6181

6182

6183

6184

6185

6186

1. Prospective students and their parents
2. Current students
3. Industry professionals
4. Educators
5. Educational organizations and authorities

6187 **Prospective students**, supported by their **parents** or guardians, are considering studying computing at a university.
6188 They need to understand differences in computing programs when making their choices between universities and their
6189 programs of study. A prospective student and their parents might have a basic understanding that the student is
6190 interested in computing as a field of study, but it is likely that few prospective students comprehend the variety of
6191 computing subdisciplines or the differences between them. Members of this stakeholder group are going to be
6192 interested in comparing the various subdisciplines, as well as in understanding the relationship between the
6193 characteristics of specific programs and curriculum standards for different subdisciplines, as well as the relationship
6194 between the outcomes of a program and the expectations of one or more jobs, or a program or subdiscipline and a
6195 career.

6196

6197 **Current students** are students that are enrolled at an institution of higher education. They might consider a choice of
6198 courses from their own institute or another institute (in some cases another department when they intend to take a
6199 hybrid curriculum of multiple subdisciplines), or in another country. Alternatively, they may be interested in moving
6200 to another educational institution. These students would be particularly interested in comparing programs between
6201 different institutions.

6202

6203 **Industry** refers to organizations that (1) are hiring graduates, (2) are collaborating with universities to choose or
6204 specialize a curriculum or need a tailor-made course, or (3) are collaborating in a curriculum by providing internships.
6205 Most importantly, industry professionals and recruiters need to understand what prospective incoming employees have

6206 learned, i.e., which competencies they have acquired during their studies. Computing professionals need various
 6207 specific skills. For example, employers who are looking for software developers might be looking for individuals with
 6208 strong competencies in software development, and thus they might be interested in software engineering graduates.
 6209 On the other hand, if an employer wants individuals who have competencies in understanding and guiding the impact
 6210 of technology on an organization in addition to a foundation in computing, then they might prefer graduates from an
 6211 information systems curriculum. Thus, understanding how particular types of curricula would fit within their employer
 6212 needs would help target which type of graduates they prefer in terms of curriculum studied.

6213 **Computing educators** are faculty members and staff within a single school or university who are responsible for
 6214 designing and implementing educational experiences, which may include a full curriculum leading to a degree or an
 6215 individual course or module as part of one or more curricula. These people may be individual university faculty
 6216 members or teams that design and teach courses, design educational resources (books, massive open online courses
 6217 (MOOCs), websites, presentation slide decks), manage curricula as taught in their school, or assess student entry or
 6218 exit levels. Computing educators need to understand how their current or prospective curriculum align with standard
 6219 curriculum recommendations, as well as understand how well the competencies of graduates match the needs of the
 6220 industry within their target market.

6221 **Educational authorities** are organizations that have authority over university education such as (national) ministries
 6222 of education that govern and finance universities and national or international (e.g., European) bodies that rate, assess,
 6223 or accredit (university) education, or define qualifications or certificates. Educational authorities need to understand
 6224 how well a specific program matches the curriculum standards for the field that it purports to teach. In many countries
 6225 or broader regions, educational authorities are responsible for developing local curriculum standards for various
 6226 disciplines, so they will have to apply the competency model along the lines of the process described later in this
 6227 appendix.

6230
 6231
 6232 **E.1.2: Competency targets**
 6233

6234 Here, we will first define the idea of a “competency target.” A competency target reflects an entity that would be
 6235 defined by providing a set of competencies. These competency targets could be in several categories—curricula,
 6236 curricular standards, jobs and careers. Effectively, anything that can be specified with a set of competencies is a
 6237 potential target.

6238 Both pre-college and college students are driven to some degree by the choice of an eventual career. Our competency
 6239 model can be applied not only to disciplines and to college programs, but also to careers and jobs. We note that a
 6240 career reflects a broad category of specific jobs, just as a discipline reflects a broad category of specific programs.
 6241 Table E.1 clarifies these concepts.

6242
 6243
 6244 Table E.1.

	Education	Workforce
Singular	(Degree) Program	Job
Aggregate	Subdiscipline	Career

6245 Developing competency-based specifications for these targets can enable comparison between various targets. For
 6246 example, a competency-based specification of a career can then form a baseline that can be used to compare against
 6247 various disciplines. The relationship between a career and various disciplines can drive the choice of the
 6248 discipline that a student should specialize in. One could pragmatically decide on a discipline based on the
 6249 distance between a desired career and a discipline, and a similar distance metric could provide guidance regarding
 6250 the preparation a program provides for either a specific job or a career.

6251
 6252
 6253
 6254 **E.1.3: Outcome Expectations and Learning Specifications**
 6255

6256 In this section, we will briefly review the differences between the competency-based approach and the traditional
 6257 approach to specifying degree programs. In computing, it has been a long-term tradition to articulate guidance and

6258 recommendations for educational programs in the form of curricula that specify the number of classroom hours that
6259 are dedicated to specific knowledge units. Knowledge units are typically further categorized into knowledge areas at
6260 a higher level of abstraction. In addition to the contact hours dedicated to it, a specification for a knowledge unit has
6261 traditionally included a list of topics and a set of learning outcomes. For most computing disciplines, curriculum
6262 recommendations do not define how the topics are structured into courses, although many of them include course
6263 exemplars to help programs determine how to organize the topic coverage into structured learning experiences
6264 (courses). The only exception to this practice in the ACM curricula are the IS curricula up to IS 2010, which have
6265 presented the curriculum recommendations as sets of courses.

6266
6267 In a traditional computing curriculum recommendation, a knowledge unit specification might consist of five to 10
6268 topics and a similar number of learning outcomes. Therefore, the level of abstraction of learning outcomes is clearly
6269 quite low. For example, in CS 2013, the Information Management Concepts knowledge unit includes eight topics and
6270 13 learning outcomes to be covered in a total of three core classroom hours (say, 12-15 total student work hours).
6271 Learning outcomes have to be narrowly specified to be achievable within the time available. Furthermore, in this
6272 approach, each learning outcome is associated with a specific knowledge unit. The structure does not include a way
6273 to specify higher level learning outcomes that would include components from multiple knowledge units (or
6274 knowledge areas).

6275
6276 Competencies as degree outcome specifications are quite different from knowledge unit -level learning outcomes:
6277 they reside at a significantly higher level of abstraction and are specified based on the performance requirements
6278 associated with an organizational task. As discussed in Chapter 4, competencies integrate multiple knowledge, skill,
6279 and disposition dimensions in a specific task context. Competency specifications are an excellent way of articulating
6280 the level at which the graduates of a degree program are expected to be able to perform at the time of graduation. As
6281 such, competencies alone do not, however, define a curriculum. In order to move from competencies (as outcomes)
6282 to a curriculum (a set of learning experiences), one has to determine a) the knowledge, skill, and disposition
6283 components of each competency, b) the learning sequencing requirements of these components, and c) a set of
6284 effective pedagogies that will allow the students to attain the required competencies. Chapter 4 has presented the
6285 conceptual groundwork for this process, and Section 5.3 discusses the process at a more detailed level.

6286
6287 As discussed earlier in Chapter 4, competencies have a hierarchical dependency structure. Competencies developed
6288 within various learning experiences will further be integrated with other competencies into larger-scale competencies
6289 as part of a longer term integrative process.

6290
6291
6292

6293 **E.2: Identifying and Authoring Competencies**

6294

6295 In this section, we discuss the processes with which various stakeholder groups can develop competency statements
6296 for their purposes. These processes vary significantly depending on the stakeholder group, such as the following
6297 examples:

- 6298 ○ A faculty team creating competency requirements as a foundation for a global curriculum recommendation
- 6299 ○ A faculty team specifying competency requirements as a foundation for a university curriculum
- 6300 ○ A government or industry group articulating competency requirements for professionals working within an
6301 industry within a country or a region
- 6302 ○ An employer who is writing competency requirements for a specific job role
- 6303 ○ An employer who is writing competency requirements for general undergraduate hires
- 6304 ○ A prospective employee describing their own competencies for the purpose of presenting them to possible
6305 employers.

6306

6307 The process of formulating competencies had emerged in the last few decades as a method for describing educational
6308 outcomes, supporting dialogue about what is expected of education from a variety of stakeholder perspectives, and
6309 articulating the needs of employers for various job or career profiles. A common use of competencies in higher
6310 education policy debates has been as a means with which to facilitate an understanding of the value of education and
6311 educational experiences. This section discusses developing descriptions of competencies from the perspective of
6312 computing curricula and explores the interplay between the specification of competencies and ways of working with
6313 stakeholders in curriculum design and development.

6314
6315 Competency, as described in the previous chapter, is a means of capturing the desirable attributes of graduate
6316 performance in situations where one's profession and expectations for professional expertise form the context. From
6317 the computing education perspective, these are situations in which our graduates act in interaction with computing
6318 environments, systems, and processes.

6319
6320 Descriptions of competence, albeit sometimes often incomplete, are already in common use in the form of learning
6321 outcomes and graduate outcomes associated with courses and degrees worldwide. Learning outcomes are an excellent
6322 starting point in terms of deriving competencies from the academic stakeholder perspective. One shortcoming,
6323 however, is that such outcomes are often structured as ability to apply knowledge or skill to a problem or situation.
6324 Aspects of prudential judgement, for instance commitment to ethical standards, personal investment in the quality of
6325 the outcomes, and dedication to personal standards of quality, communication and collaborative behavior are often
6326 missing or poorly described.

6327
6328 To address this issue, we highlight several perspectives to the processes of deriving competencies in collaboration
6329 with stakeholder groups. All these approaches adopt a requirements analysis model familiar to computing practitioners
6330 and are guided by the definitions and competency component structures presented in Chapter 4.

6331
6332

6333 **E.2.1: Free-form Narratives vs. Semi-formal Specifications**

6334
6335 Chapter 4 presents a semi-formal component structure for competencies in order to build a strong foundation for
6336 understanding the competency concept and for analyzing and comparing competency-based program specifications in
6337 a structured way. As already clear based on Chapter 4, a component-based structural specification is not, however,
6338 always the best way to specify curricula. For example, the curriculum guidance documents in computing that have
6339 followed the competency-based approach so far (IT 2017, MSIS 2016, and SWECOM) have all presented
6340 competencies without any formal specifications or limitations of the structure or vocabulary of the competency
6341 statements.

6342
6343 In the development of IT 2017 and MSIS 2016 industry and/or government documents (such as SFIA, e-CF 3.0 [15]
6344 and Clinger-Cohen [14]) were consulted as a source of guidance for the form of typical competency statements, in
6345 addition to academic research. Competency statements typically start with a command verb and are written to express
6346 an expectation set for an organizational role in a specific task context. The general expectation of a competency
6347 statement is that it should capture elements of knowledge, skills and dispositions, but when written in a typical free-
6348 form narrative, those three components cannot necessarily be identified directly as separate elements without further
6349 analytical work.

6350
6351 Experiences developing competencies as a path to curricula in other disciplines has concluded that it is important that
6352 the initial process of writing competency statements is not constrained by a fixed component structure or narrowly
6353 defined lists of options [9] [10]. When an employer articulates the competencies that their incoming employees are
6354 expected to have or gain rapidly after the beginning of employment, it is unrealistic to expect that they would be
6355 willing to constrain these statements to a limited vocabulary or a tightly specified grammar. In the same way, if the
6356 purpose of competency statements in a curriculum guidance document (either locally or globally) is to convey them
6357 to prospective students or prospective employers, limiting the vocabulary or forcing a highly constrained structure is
6358 unlikely to improve understanding. The competency statements need to be written in a way that they are
6359 understandable and meaningful for the key stakeholder groups that will be using them.

6360
6361 At the same time, as discussed in Chapter 4, there are good reasons to articulate a formal component structure for
6362 competencies and limit the set of possible elements for each component type: in practice, it is impossible to analyze,
6363 compare and visualize competencies effectively unless the narrative free-form is somehow converted into a semi-
6364 structured format. As we will discuss later, it is also possible that the quality of the free-form narratives can be
6365 improved by using the structural and vocabulary analysis as a way to improve their coverage of the competency
6366 structure established in Chapter 4.

6367
6368

6369 **E.2.2: Eliciting competencies**

6370

6371 In many ways, specifying competency statements for a specific context is a requirements specification tasks. Instead
6372 of articulating requirements for the performance of a software application or a system, competency statements specify
6373 performance requirements for individual professionals in a context. Still, the process of eliciting and specifying
6374 competencies shares many characteristics with a requirements discovery and structuring processes.

6375

6376 First and foremost, specification of competencies is typically a collaborative process among multiple stakeholder
6377 groups. Competencies should be derived from interaction with stakeholder groups in collaboration with curriculum
6378 designers. In the process of writing competency statements, the stakeholder groups include, for example, employers,
6379 students, and regulatory/accreditation bodies, as discussed earlier in this appendix.

6380

6381 To derive expressions of competency a range of strategies can be employed, using approaches that could include all
6382 the typical tools of multi-method knowledge discovery processes, such as interviews, surveys, observation, face-to-
6383 face and online focus groups and other collaborative processes, etc. Regardless of the stakeholder type, the
6384 fundamental question driving the process typically is: what tasks should the graduates/future employees be able to
6385 perform in an authentic context at the time when they complete a particular program experience? The discovery
6386 process should lead to statements of professional expectations.

6387

6388 There is surprisingly little existing literature regarding the process of authoring competency statements. Chambers
6389 and colleagues describe a process for deriving competencies, as well as a constrained language with which to describe
6390 them [9] as does Lenburg [11]. Other good examples of how to work with competencies and curriculum design and
6391 implementation are presented in the work of Squires and Larson in Space Systems Engineering [12].

6392

6393 While it is impossible for us to provide specific guidance for each of the stakeholder groups and types of competencies
6394 that might emerge, it is, however, possible to provide general guidance for writing them. Lenburg [11] offers the
6395 following recommendations for writing competency statements:

6396

1. They should be worded as learner-oriented, essential competencies
2. They should be worded in “clear, specific, unadorned, and concise language,” and they should be measurable.
3. They should be action oriented and begin with “the verb that most precisely describes the actual, preferred outcome behavior to be achieved.”
4. They should be consistent with “standards, practice, and real-world expectations for performance,” thus reflecting what “the practitioner actually needs to be able to do.”
5. They should contribute to the “cluster of abilities needed by the graduate to fulfill the expected overall performance outcomes.”

6400

6401 In a free-form competency statement, the focus is typically on the general outcome of the competency in the context;
6402 in this way of expressing a competency, the knowledge, skill, and disposition components might not be fully exposed;
6403 instead, they need to be inferred from the free-form statement in a way described in section 5.2. Articulating the
6404 context is, however, always crucial, since it provides the motivation for the student, making it meaningful to learn and
6405 perform that competency. It is, therefore, important to develop the competency's core content in tandem with
6406 practicing and developing relevant skills and demonstrating dispositions that positively influence learner agency with
6407 respect to sense of self, and responsibility for interactions with others.

6408

6409

6410 **E.2.3: Hierarchical Structure of Competencies**

6411

6412 In order to define the highest level or most abstract competencies of a program, course, or curricular unit, it is necessary
6413 to articulate the knowledge, skill, and disposition components associated with an authentic context. Competency
6414 learning in curricula contexts can be represented as a progression, and this view also allows curriculum designers the
6415 opportunity to define lower order competencies from which a higher-level competency might be derived. A
6416 competency that does not depend on other preceding competencies provides curriculum designers with fundamental
6417 learning activities that are self-contained, in the sense that they involve fundamental knowledge, and skills associated
6418 with the specific dispositions needed to apply that knowledge and skill meaningfully in appropriate contexts.

6419

6420 In our approach, we assume for competency design purposes that a learner develops competencies in a progression,
6421 leveraging competencies she has already attained in the process of developing new ones. Hence competencies do not,
6422

6423

6426 in general, stand alone but coexist in a dependency framework, and each competency may be associated with a set of
6427 preceding competencies in addition to its knowledge, skill, and disposition learning components (see Figure 5.2). This
6428 leads to a directed acyclic graph of competencies, where each competency has a unique set of associated learning
6429 components. Note that the precedence graph is not necessarily a tree, and in most cases will not be, for two reasons:
6430 there may be no single culminating competency in the progression, and a single competency may be a component of
6431 multiple competencies later in the progression.

6432
6433

E.2.4: Deriving Semi-formal Specifications from Free-form Narratives

6434

6435 Section 5.2.1 discussed the differences between semi-formal competency specifications and free-form competency
6436 narratives. In this section, we describe the process of deriving the semi-formal specifications based on the free-form
6437 narratives developed, for example, in discussions with relevant external stakeholders. The purpose of this activity is
6438 to discover the underlying component structure of the free-form competency statements, the development of which
6439 we discussed above in 5.2.2. Through this process we not only gain a form of the competencies that can be used for
6440 analytics or visualization, but we will also gain a significantly more sophisticated understanding of the nuances of the
6441 competencies.

6442

6443 An example of such a process can be derived from the work of Squires and Larson [12]. They draw on earlier work
6444 in the Space Systems Engineering community to define a series of competencies in relation to practice in the
6445 profession. These competencies, as quoted in their paper, are free form and rather abstract in nature.

6446

6447 E.g. "Manage systems engineering", which implies a combination of the knowledge base for maintaining complex
6448 space engineering system solutions, together with skill in space engineering principles and procedures, performed in
6449 space engineering contexts where commitment to the quality and failsafe nature of the outcomes of the process were
6450 demonstrably of high value and always palpable in the context of the decision making of the individual in systems
6451 management and development.

6452

6453 These statements are then analyzed, decomposed and ultimately regrouped in the form we describe in Chapter 4,
6454 having used the original data as a means to derive competency statements that include elements of knowledge, skill
6455 combined with dispositions related to professionalism in an application context.

6456

6457 In terms of the "manage systems engineering" competency example given above, our recommended process would
6458 involve a subsequent decomposition and expansion step as the high-level competency was evolved into knowledge,
6459 skill and contextual dispositional components as is shown in the subsequent explanatory text in the same example.
6460 Once these high-level free form statements are transformed into the structure developed in Chapter 4, curriculum
6461 designers and instructional designers can transform them into statements describing learning activities and experiences
6462 where the relevant context can be created in order for the competency to be observed.

6463

6464

E.2.5: Authoring Free-form Narratives from Competency Components

6465

6466 It is also possible to move in the other direction and author free-form narratives by first focusing on the components.
6467 Identifying the knowledge, skills, and disposition components of a competency first may be a good starting point in
6468 cases when the full identity of the target competency is not clear and needs to be calibrated at the level of the
6469 components first. The danger of this approach is that the author may ignore the fact that the whole is typically much
6470 more than a pure aggregation of the components. Still, if the authors are able to specify the component structure, it
6471 certainly provides valuable guidance for formulating the competency narrative.

6472

6473

6474

6475

6476

E.3: Using Competency Specifications as a Foundation for Curriculum Specifications

6477

6478 One of the essential questions for any academic administrator or faculty member who is developing a degree program
6479 or other collection of learning experiences based on a competency-based approach is as follows: How do we determine
6480

6481 a set of educational experiences that, if not guarantee, at least significantly increase the probability of the proposed
 6482 program’s graduates being able to achieve the competency expectations that have been set for them. It is useless to
 6483 specify competencies as program outcomes unless there is a meaningful mechanism for identifying and structuring a
 6484 set of learning experiences that enable the students to achieve the specified competencies (Chyung et al [13], MSIS
 6485 2016). In other words, a set of competency statements have to be transformed into a curriculum form consisting of
 6486 educational activities that help to scaffold the student’s progression in various types of outcome areas.
 6487

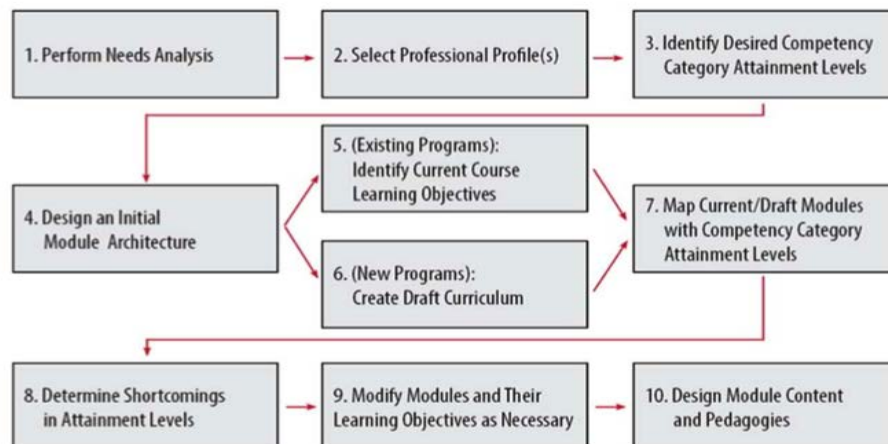
6488 In this section, we will discuss several examples of existing models that have been proposed for the purpose of
 6489 converting a set of competency expectations into a curriculum. As you will see, they appear to share some basic
 6490 characteristics.
 6491

6492 Just as a reminder, we use the following terminology: *Competencies* serve as outcome expectations at the level of a
 6493 degree program (or other aggregate structure toward which students are working; for the sake of simplicity, we use
 6494 the term program). *Learning experiences* are courses, modules, or other similar *sets* of *learning activities* that
 6495 collectively constitute a program. Each learning experience leads to a set of learning outcomes, which collectively
 6496 need to enable the students to attain the required competencies. A *curriculum* specifies, at a minimum, the topics,
 6497 pedagogical approaches, and learning outcomes for each of the learning experiences.
 6498

6499 **E.3.1: Existing Models**

6500 MSIS 2016 [citation] presents a process for deriving a set of learning experiences (referenced as modules) based on a
 6501 set of competency specifications (see Figure 5.1). This process assumes an underlying competency model similar to
 6502 that of MSIS 2016, which includes 10 competency areas, 88 competency categories (each associated with one of the
 6503 competency areas), and several detailed competencies within each category. Furthermore, it recognizes five attainment
 6504 levels for each of the competency categories (awareness, novice, supporting, independent, and expert), specifying that
 6505 educational programs are seldom sufficient to help anybody achieve the expert level.
 6506

6507 The process of developing the learning experiences in the MSIS 2016 model starts with a program needs analysis
 6508 (Step 1) combined with the determination of the job roles that the program plans to focus on (Step 2). Based on the
 6509 outcomes of the first two steps, the program determines competency [category] attainment levels that it assumes its
 6510 graduates to achieve (Step 3), including inclusion of potential brand-new competency statements. Next, the model
 6511 suggests that the program should develop or confirm an initial architecture for the learning experiences (Step 4),
 6512 followed by either verification learning experiences at the learning objective level (Step 5; program based on existing
 6513 courses) or drafting a set of new learning experiences with learning objectives (Step 6; new programs). Following this,
 6514 the model suggests that in Step 7, the results of Step 5/Step 6 are mapped with the competency attainment levels
 6515 specified in Step 3. In Step 8, the differences between Step 3 and Step 7 are identified. Step 9 is for determining
 6516 required modifications to the learning experiences and/or their learning outcomes to address the differences identified
 6517 in Step 8. Finally, in Step 10, detailed learning experiences are designed, including the topics and pedagogies.
 6518
 6519
 6520



6521

Figure E.1 Process for deriving learning experiences from competency specifications

Squires (2009) summarizes another approach for developing a curriculum based on a set of competencies using the International Academy of Astronautics (IAA) Space Industry Systems Engineering competency model. This model itself consists of 10 competency areas and 37 capabilities each associated with one of the competency areas. Furthermore, the model specifies four proficiency levels: Participate (Know), Apply (Perform), Manage (Lead), and Guide (Strategize). Squires’s process includes the following items:

1. Select the competency model to use
2. Validate the most important (‘critical’) competencies
3. Determine the current curriculum’s ability to enable the students to attain the required proficiency levels
4. Determine the proficiency levels that the future curriculum needs to attain
5. Identify the gap areas between the current and the future curriculum
6. Assess and fix the gaps.

In this model, the sixth state is the part of the process that addresses the curriculum and specifies how the curriculum (both topics and pedagogy) have to be modified (or created in the case of a new program) so that the curriculum will enable students to attain the required proficiency levels.

Finally, Chyung et al [13] propose another six step process, which covers both authoring of the competencies and curriculum design based on them. In this process, the first three steps include— borrowing the authors’ terminology— the use of three sources of data for determining the competencies (alumni & industry analyses, professional standards & curriculum benchmarking, and departmental goals and curriculum review). Thus, the fourth step, actual authoring of the competencies, will be based on two types of external resources (employer needs and national and global competency/curriculum guidance) and internal goals and review processes. Chyung et al [13] include the development of the learning experiences as the fifth step, during which “the key to this ongoing process is to carefully align the stated course objectives, the competencies that apply to that course, and the graded course assignments” (p. 311). In this process, it is important to ensure that course goals are aligned with “applicable competencies” and that the course includes learning processes that “both help students acquire those competencies and assess the extent to which they have been successful.”

Summarizing key findings from these three models suggests the following:

- In all of the models, the characteristics of the learning experiences that constitute the curriculum are determined based on the outcome expectations specified with competencies.
- All of the models assume that the program competencies are identified (at least partially) based on existing competency models (developed by industry/government groups or by professional societies).
- Two of the three models recognize that identifying the outcome expectations as a set of competencies is not sufficient; in addition, in these models an expected attainment level needs to be specified for each competency.
- None of the models provide specific guidance for the process of deriving learning experiences from competencies. It appears that the models imply a need to first derive a set of learning outcomes associated with each of the competencies. Then, the learning outcomes need to be organized into learning experiences. The sets of learning outcomes within each learning experience will determine the topics that the students need to be engaged with and the forms that this engagement takes (pedagogy).
- Each of the approaches offers at least hints regarding the need to continuously assess the extent to which the implemented learning experiences enable the students to attain the expected competencies at the expected level.

E.3.2: Building Curricular Guidelines by Based on Competency Specifications

In this section, we will discuss the special characteristics of the processes that entities providing educational recommendations to a large number of programs need to follow when developing those recommendations. As discussed earlier in this report, there are two existing recommendations (IT 2017 and MSIS 2016) that are based on a competency-based approach. These reports were developed separately, and they are not fully mutually consistent. The

6577 other curriculum recommendations (CE 2016, CS 2013, CSEC 2017, and SE 2014) are with one exception (IS 2010)
6578 based on the Ka-Ku-Lo (knowledge area -- knowledge unit -- learning objective) model.
6579

6580 The most important decisions that future curriculum recommendation authors need to make are as follows: a) will
6581 they follow some competency-based approach, focusing first on the outcome expectations specified with competencies
6582 and b) if the answer to the first question is affirmative, will they also include traditional Ka-Ku-Lo material and/or
6583 course exemplars in addition to the competencies? Neither question is trivial, and particularly in the context of the
6584 second one, there are arguments supporting both approaches. On one hand, just focusing on competencies emphasizes
6585 the fact that providing globally applicable guidance regarding curriculum elements (learning experiences, Ka-Ku-Lo
6586 elements) is very challenging because of the differences in local educational architectures and basic requirements. It
6587 is more effective just to specify the outcome expectations with competencies and provide strong guidance for
6588 transforming competencies into curriculum components. On the other hand, many schools and departments using
6589 professional society curriculum recommendations do not have the required resources to go through the time-
6590 consuming and resource intensive process of deriving learning experience specifications (courses) from the
6591 competencies. They expect learning experience (course) specifications. In addition, providing program-level
6592 recommendations in the form of outcome expectations will not remove the usefulness of the frequently maintained
6593 Ka-Ku-Lo structures.
6594

6595 What are the resources that professional society recommendations should use to determine the recommended
6596 competencies? Obviously academic expertise will continue to be important, as has been the case throughout the history
6597 of curriculum recommendations. At the same time, engaging in a dialogue with relevant industry partners is even more
6598 important in a competency-based approach than it used to be. One of its main benefits is, after all, that it allows
6599 programs to align well with the employer expectations and make the alignment clearly visible through a formal
6600 mapping process.
6601

6602 Not coincidentally, there are several computing/IT competency models that have been developed in extensive, well-
6603 funded government/private sector processes, such as e-CF 3.0 [15] and SFIA (see Appendix B). Using them as a
6604 source of inspiration and improved understanding of industry requirements makes sense. Depending on the
6605 subdiscipline, there might be specialized industry or government efforts that provide additional focused guidance
6606 (such as, for example, the use of the Clinger-Cohen report in the context of MSIS 2016; CIO Council [14].
6607
6608

6609 **E.3.3: Building University-level Curricula Based on Competency Specifications**

6610
6611 The examples specified in Section 5.3 have a relatively good direct fit at the university level. The key element of
6612 university level competency-based outcome expectation identification processes is that the process to identify the
6613 competencies includes input from key employer and/or alumni partners, professional/academic society guidance, and
6614 program's own identity and resource availability. Once the competencies have been identified, the authoring of the
6615 learning experiences requires the identification of learning experience outcomes based on the competencies,
6616 configuring the learning outcomes into groups that represent learning experiences, and then designing learning
6617 activities within each learning experience based on the learning outcomes. Obviously, this is seldom a process that
6618 starts from a clean slate—existing learning experiences form the foundation for the work.
6619
6620

6621 **E.3.4: Specifying Program Outcomes as Competencies from Pedagogical Requirements**

6622
6623 One of the positive but potentially resource-intensive impacts of specifying program outcomes with competencies is
6624 that enabling students to develop skills and dispositions in many cases requires a different set of pedagogical
6625 assumptions and approaches compared to a mostly knowledge-based specification of outcomes (and their assessment).
6626 In practice, competency-based outcome specification will lead to a broader set of types of learning experiences, often
6627 including a much stronger focus on various forms of experiential learning from interactive simulations to intensive
6628 projects to field experiences to internships and co-ops. Particularly domain-specific skills and dispositions require a
6629 learning environment that is different from a traditional classroom environment.
6630
6631
6632

6633 **E.4: Competencies and Stakeholder Value**

6634

6635 Competencies, through their task context, are closer to the language with which employers describe their needs than
6636 the traditional Ka-Ku-Lo model can achieve. Consequently, competency specifications communicate value for the
6637 (prospective) employer organization more directly and transparently than knowledge-based specifications do.
6638 Therefore, competencies help other stakeholders such as students, parents and the public sector understand what future
6639 careers degree programs are aligned with.

6640

6641 Competencies as a conceptual framework for valuing the outcomes of higher education can be traced back to the
6642 1970's and legal, nursing and teacher vocational training programs in the U.S. These programs emphasized the
6643 acquisition of the behavior exhibited by outstanding professionals as a way to identify and develop desired skill sets
6644 through education and training [8]. The resulting approach focused on training by mimicking desirable behavior, and
6645 ultimately did not produce the intended competencies. Consequently, these experiments did not attract much of a
6646 following. Renewed interest from labor organizations and vocational education in the concept emerged throughout
6647 the late 1980's, but, it was only at the end of the 1990s that higher education began to renew its participation in the
6648 conversation. Klink, Boon, and Schlusmans [5, p. 2] highlight the following contributing factors: 1) a shift in the job
6649 market towards increased career and professional mobility; 2) the emergence of the "knowledge worker" and
6650 "knowledge economy" in which application of knowledge and skills and "the motivation to keep learning" are
6651 considered essential to personal and professional growth; 3) new trends in higher education in response to an
6652 increasingly dynamic and complex world that makes acquisition of technical knowledge insufficient; and 4)
6653 innovations in learning sciences and education, such as participatory learning, deep learning, and contextualization.
6654 As these trends became integrated into the mainstream of higher education, they also evoked a switch in the value
6655 proposition of education "from knowing to learning", and ultimately to ability to perform in a relevant and high value
6656 manner in professional contexts.

6657

6658 The link between competency and professionalism and high levels of professional performance in relevant domain
6659 areas has long been a part of promotion and salary processes in the public and private sector. Competencies emerge
6660 immediately in the discourse of job advertisements and employer vernacular. Van der Klink and Boon[4] make the
6661 case for the popularity of the concept due, ironically, to the lack of clarity over the term competency and maintain that
6662 the number of definitions "is probably incalculable" [5]. A literature study by Stoof, Martens, and Van Merriënboer
6663 [6] places the word in the "wicked words" category, meaning that its limits are hard to determine, which makes
6664 complete agreement on its meaning elusive. Despite its continuing fuzziness, the term promises to be useful in
6665 bridging the gap between educational outcomes and job requirements [7].

6666

6667 CC2020's definition of competency provides the opportunity for mutually consistent specifications of practitioner
6668 competency: relating attributes possessed by an applicant to those required by an employer. Of significant mutual
6669 benefit both to computing employers and to the academy would be the development of specification standardization
6670 between curricular competency and employer job descriptions. Modeled competencies offer the opportunity for
6671 academic computing to clearly describe their graduates' capabilities while at the same time help employers to more
6672 clearly communicate their functional job requirements. In such a circumstance the computing educators would have
6673 the opportunity to weigh their educational goal descriptions against industry needs. As a result, human resource
6674 activities in industry would find it easier to identify likely institutional sources of graduates with relevant competency
6675 profiles as prospective future employees.

6676

6677 Competency offers a contextualized model through which communication of practitioner capabilities of graduates can
6678 be realized. This in turn better serves the coordination and collaboration among institutions of computing education
6679 along with the human resource activities of industry. Furthermore, this model may better facilitate advising prospective
6680 students who wish to align their studies with clearly described employment opportunities. All the while such a
6681 collaboration can influence curricula in educational programs by providing a better understanding of job markets they
6682 may wish to serve. In any case, specific competency descriptors offer a facilitating bridge in the dialog between
6683 academia and industry locally, nationally, and internationally.

6684

6685 The explicit fusion of knowledge and skills adopted in CC2020 emphasizes the role of practice in the process of
6686 demonstrating "knowing" [25]. Enhancing the existing learning outcomes approach, which has been a prominent
6687 feature of curricular description, competency's fusion of knowledge and skills advocates for an explicit goal of
6688 crystalizing the dimensions of practical professional capability in curriculum description. The intrinsic role of task in

6689 both pedagogy and assessment provides a natural opportunity for an explicit articulation of the interdependence of
6690 curriculum and employability.

6691
6692
6693

6694 **E.5: Assessing Competencies**

6695
6696
6697
6698
6699
6700
6701
6702
6703
6704
6705

For competency modelling to be useful for computing education, it is also essential that there be an effective coupling to assessment and curricula expectations. When examining issues related to competency modelling, this is, in fact, one of the issues that has been prevalent in the literature [1]. Outcomes assessment, both at the degree-program and course level are essential to the quality management of effective educational processes. In current practice, the target of most outcomes assessment in computing is predominantly declarative knowledge and skills related to computing (e.g. examine the definition of learning outcomes enshrined in the Bologna Process [1]). This is corroborated by Fuller, who observes that "Professional skills and attitudes form an increasingly large part of the requirements of computer science graduates. Students are assessed on their knowledge and cognitive skills but not on the attitudes that will lead them to practice in the workplace what they have been taught in the classroom"[3].

6706 Hence the assessment of dispositions, and particularly student performance of the integration of disposition, skill and
6707 knowledge provides significantly more power to curricula designers working to improve the definition of outcomes
6708 at different levels of abstraction.

6709
6710
6711

6712 **E.6: Summary**

6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723

The main focus of this appendix has been on two key questions related to the use of competencies for specification of degree programs in computing: first, what processes should be used to produce competency statements (and how do we determine whether or not the produced statements are an appropriate reflection of the outcome expectations) and second, how do we create curricula (specifications of learning experiences) based on the competency specifications. Furthermore, the chapter also discussed the value of the competency-based approach to various stakeholder groups and pointed out the essential role competency specifications can serve in assessment. Overall, the goal of this appendix has been to provide applied guidance for the use of competencies in program and curriculum guidance development.

6724 **E.7: References for Appendix E**

6725
6726
6727
6728
6729
6730
6731
6732
6733
6734
6735
6736
6737
6738
6739
6740
6741
6742
6743
6744
6745
6746
6747
6748

- [1] I. Baumgartner and V. Shankararaman, 'Actively linking learning outcomes and competencies to course design and delivery: experiences from an undergraduate Information Systems program in Singapore', in *Proceedings of the 2013 IEEE Global Engineering Education Conference*, Berlin, Germany, 2013, pp. 238–246.
- [2] E. Pikkarainen, 'Competence as a Key Concept of Educational Theory: A Semiotic Point of View', *J. Philos. Educ.*, vol. 48, no. 4, pp. 621–636, Nov. 2014, doi: 10.1111/1467-9752.12080.
- [3] U. Fuller, A. Pears, J. Amillo, C. Avram, and L. Manilla, 'A Computing Academic Perspective on the Bologna Process', *ACM SIGCSE Bull.*, vol. 38, no. 4, pp. 115–131, Dec. 2006.
- [4] M. Van der Klink and J. Boon, 'The investigation of competencies within professional domains', *Hum. Resour. Dev. Int.*, vol. 5, no. 4, pp. 411–424, 2002.
- [5] M. Van der Klink, J. Boon, and K. Schlusmans, 'Competences and Vocational Higher Education: Now and in Future.', *Eur. J. Vocat. Train.*, vol. 40, no. 1, pp. 67–82, 2007.
- [6] A. Stoof, R. L. Martens, and J. J. Van Merriënboer, 'What is competence? A constructivist approach as a way out of confusion', *Onderwijs Res. Dagen ORD Leiden Neth.*, 2000.
- [7] D. Kennedy, A. Hyland, and N. Ryan, 'Learning outcomes and competences', *Introd. Bologna Object. Tools*, pp. 2–3, 2009.
- [8] G. Grant, 'On competence: a critical analysis of competence-based reforms in higher education.', 1979.
- [9] D. W. Chambers and J. D. Gerrow, 'Manual for Developing and Formatting Competency Statements', *Journal of Dental Education*, vol. 58, no. 5, pp. 361–66, 1994.
- [10] F. E. Weinert, 'Definition and selection of competencies: Concepts of Competence', *Organization for Economic Co-operation and Development*, 1999.
- [11] Lenburg, C. B. (1999). The framework, concepts and methods of the competency outcomes and performance assessment (COPA) model. *Online Journal of Issues in Nursing*, 4(2), 1-12.
- [12] A. Squires and W. Larson, 'Improving systems engineering curriculum using a competency-based assessment approach', *International*

- 6749 *Journal of Intelligent Defence Support Systems*, vol. 2, no. 3, p. 184, 2009.
- 6750 [13] Chyung, S. Y., Stepich, D., & Cox, D. (2006). Building a competency-based curriculum architecture to educate 21st-century business practitioners. *Journal of Education for Business*, 81(6), 307-314.
- 6751 [14] CIO Council. (2012). Clinger-Cohen core competencies & learning objectives. *Washington, DC: CIO Council*.
- 6752 [15] European Qualifications Framework (EQF). European Commission. Learning Opportunities and Qualifications. <https://ec.europa.eu/plotus/en/content/descriptors-page/>. Accessed 2019Aug17.
- 6753 [16] Voorhees, R. A., & Bedard-Voorhees, A. (2016). Principles for Competency-based education. In *Instructional-Design Theories and Models, Volume IV* (pp. 49-80). Routledge.
- 6754 [25] Wiggins, G., & McTighe, J. (2011). *The Understanding by Design Guide to Creating High-Quality Units*. Alexandria, VA: Association for Supervision and Curriculum Development.
- 6755
- 6756
- 6757
- 6758
- 6759
- 6760
- 6761

DRAFT

6762

6763 **Appendix F: Repository Development**

6764

6765 An approach that can be taken is to create an experimental repository for the eventual structure of competency
6766 encoding. This section describes the development of a digital repository using data drawn from published curriculum
6767 guidelines using screen scraping and vocabulary machine learning tools. The goal is to design a framework that can
6768 accommodate a three-dimensional concept of competency (knowledge, skill, and disposition) regardless of how one
6769 defines those three dimensions.

6770

6771 **F.1: Repository Development**

6772

6773 In the experimental repository, a select team used Eduglopedia [Edu1] as the source for knowledge area elements.
6774 This open and free global encyclopedia for higher education contains more than three thousand course descriptions
6775 and more than nine hundred program descriptions from approximately five hundred institutions. Furthermore, it uses
6776 Bloom’s Cognitive Process Dimension [And1] as a placeholder for skill.
6777

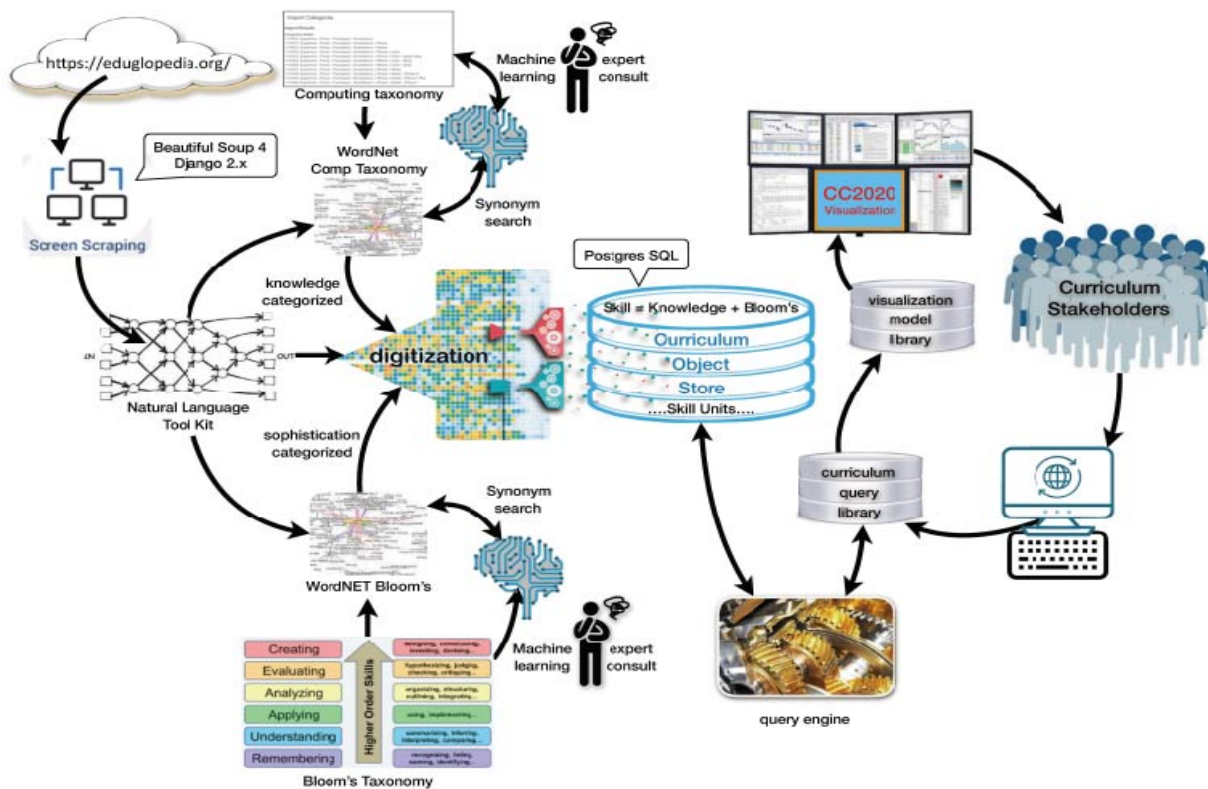


Figure F.1. Repository development process

6778

6779

6780

6781 The repository development uses Beautiful Soup (a Python package) to screen scrape the Eduglopedia that generates
6782 XML descriptions of various knowledge areas and relationships within curricula. It also uses tools such as synonym
6783 search and machine learning to generate a computing taxonomy and to identify specific verbs that applies to the
6784 different levels of Bloom’s Cognitive Process Dimension. It also generates a database of skills and knowledge as a

6785 source for queries that in the end allows for visualization of a curriculum. Figure F.1 shows the result of the repository
6786 development process. The repository data collection and visualization support are done as follows:

- 6787 1. Use Beautiful Soup to screen scrape Eduglopedia and obtain XML descriptions of various knowledge areas
6788 and relationships within curriculum.
6789 2. Use tools such as synonym search and machine learning (as well as review by human experts) to:
6790 (a) generate a computing taxonomy, and
6791 (b) identify verb sets that apply to the specific cognitive process levels of Bloom's.
6792 3. Generate/digitize a database of knowledge elements in semiotic order, and skills as applied knowledge in
6793 Bloom's cognitive process ordering.

6794 The resulting repository, the curriculum object store, serves as the source for queries that interfaces with a library
6795 of representational models allowing users to select, visualize, and/or compare curricular specifications: curriculum
6796 guidelines, program catalogs, course descriptions, accreditation standards, and job advertisements.
6797

6798 **F.2: References for Appendix F**

- 6799 [And1] Anderson (2001). Bloom's Cognitive Process Dimension.
6800 [Edu1] Eduglopedia. <https://eduglopedia.org>
6801
6802
6803

6804

6805 **Appendix G: Additional Visualizations and Analyses**

6806

6807 This appendix shows visualizations that were considered during the CC2020 project. Note that some of the
6808 terminology that appears in this appendix does not necessarily comply with the terminology that was given in Chapters
6809 4 to 6.

6810

6811

6812

6813 **G.1: Use Case-based Analysis**

6814

6815 This section gives four example use cases. Note that the two use cases in G.1.1 and G.1.2 are taken from
6816 [Tak5].

6817

6818

6819 **G.1.1: Case 1: Question from Prospective Student**

6820

6821 A student is interested in entering undergraduate education in computing and wants to know what type of
6822 curriculum would best fit her interests. She might have some ideas about dispositions that are relevant in
6823 her future curriculum, and/or have a preliminary view on domains that would provide her with future job
6824 opportunities. She might start by checking promising dispositions (or, alternatively, she could start by
6825 choosing the knowledge categories and areas – we show only the first scenario, but the alternative would
6826 lead to the same results). She would see a list of dispositions (Figure G.1(a)), from which she would choose,
6827 resulting in the interface showing the chosen dispositions as shown in Figure G.1(b). Note that the
6828 dispositions are indicated by color, as there is no order dimension.

6829

	Dispositions
	Proactive
	Self-directed
	Passionate
	Purpose-driven
	Professional
	Responsible
	Adaptable
	Collaborative
	Responsive
	Meticulous

	Dispositions
✓	Proactive
	Self-directed
✓	Passionate
	Purpose-driven
	Professional
✓	Responsible
	Adaptable
	Collaborative
✓	Responsive
	Meticulous

(a) Before choosing

(b) After choosing

Figure G.1. Choosing dispositions by a prospective student

6830

6831

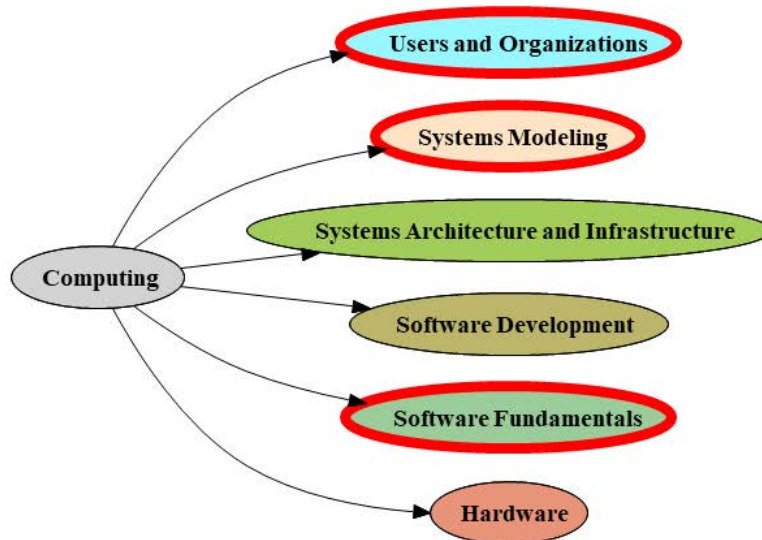
6832

6833

6834

6835 The student may also indicate which knowledge categories and knowledge areas seem interesting for her.
6836 Figures G.2 and G.3 show a possible process. She first chose three categories: Users and Organizations,
6837 Systems Modeling, and Software Fundamentals. In Figure G.2, the ellipse of these three categories are
6838 highlighted with red borders. If needed, the student could indicate which individual knowledge areas are

6839 most relevant. Figure G.3(a) shows the knowledge areas for each of the chosen three categories. The student
 6840 chose the knowledge area User Experience Design for Users and Organizations category, and Systems
 6841 Analysis and Design and Requirements Analysis and Specification for Systems Modeling category; again
 6842 the ellipse of the chosen knowledge areas are highlighted with red borders. The student did not want to
 6843 make a detailed choice in the category of Software Fundamentals. The resulting choices are shown in Figure
 6844 G.3(b).
 6845



6846
6847
6848

Figure G.2. The student's choice of computing categories



6849
6850

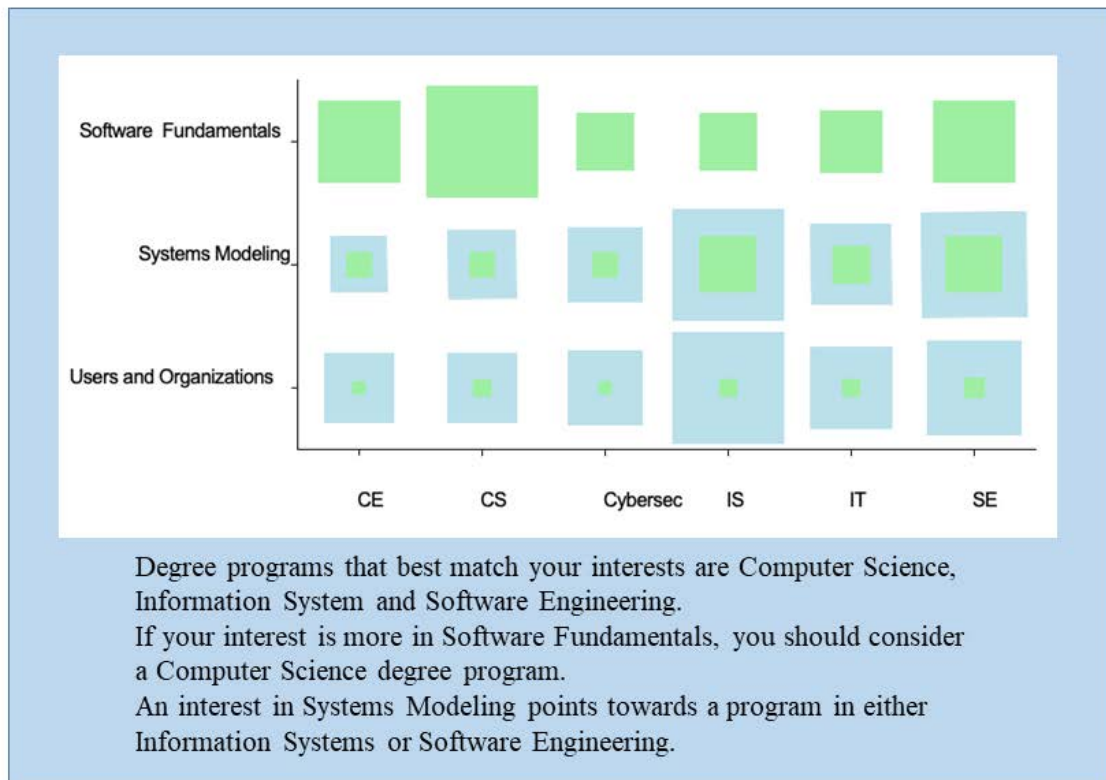
(a) Choosing knowledge areas

(b) Final result

6851
6852
6853
6854
6855
6856
6857
6858
6859
6860
6861
6862
6863
6864
6865

Figure G.3. Detailed choice of knowledge areas

If the student is satisfied with this set of knowledge areas, she may confirm and ask for a global view of how the various curricula match her interests. Based on the student’s choices, the system searches for curricula that fit this intended content. In Figure G.5, the intended knowledge categories (which have been partly specified into knowledge areas) are mapped for each of the curricular guidelines. The blue squares indicate the extent to which the knowledge area/category is relevant in the corresponding curriculum. The green square is the relative match of the student choices to that of the curriculum. The calculation of the size of the blue and green squares is not fixed yet, but for example, the green square could be based on the weights that were given in the Knowledge table shown in Appendix D. Since the student is more interested in software modeling, based on the message given in Figure G.4, the student decides to explore details regarding SE and her favored knowledge categories. By hovering over a square (Figure G.5), the corresponding competencies are listed. Also displayed are the dispositions linked to the competencies along with the relative level computed from the student choices.



6866
6867
6868

Figure G.4. Mapping of chosen knowledge categories to the curricular guidelines

6869

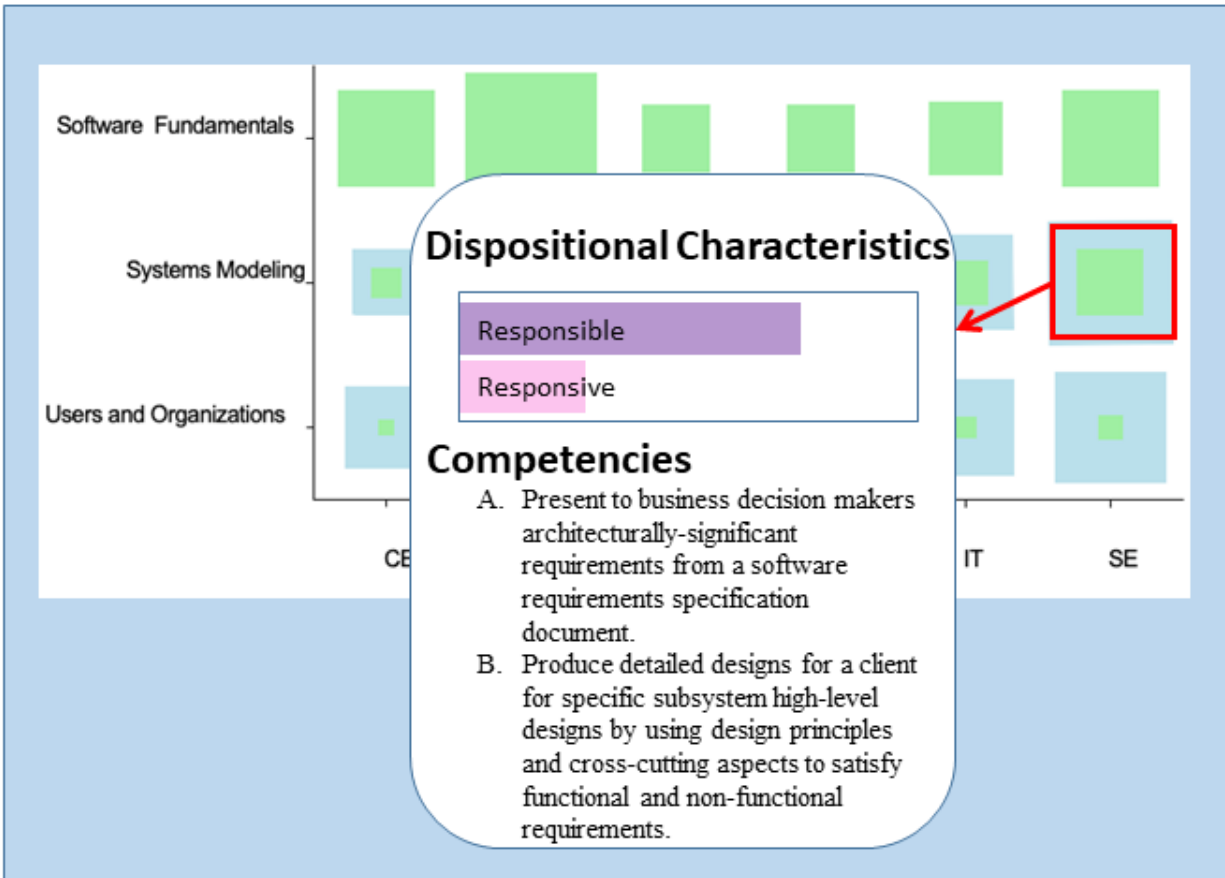


Figure G.5. Disposition and competency details

6870
6871
6872
6873
6874
6875
6876
6877
6878
6879
6880
6881
6882
6883
6884
6885
6886

G.1.2: Case 2: Question from Industry

A user from industry has developed a list of relevant knowledge areas for which relevant skills, knowledge levels, and/or dispositions are required for the company’s computing employees. She wants to find out which curriculum might potentially provide professional education for the company’s employees, in their context. Initially, CS and IT seem to be available and promising.

Similar to the process that the student took in Figures G.1 and G.2 in Case 1, she decides to choose Hardware, Software Fundamentals, and Software Development as categories that seem relevant, and removes the other three categories. She then checks the knowledge areas for each of the chosen categories, and chooses the areas that she believes to be relevant for her, resulting in Figure G.6.

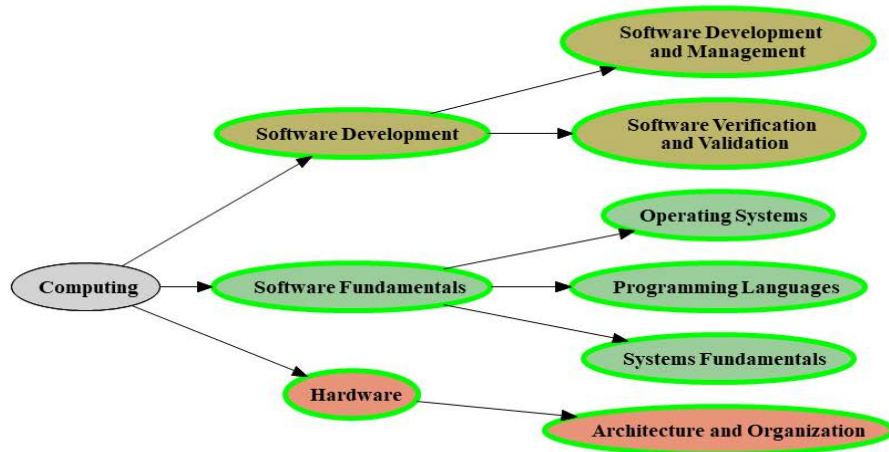


Fig. G.6. Result of knowledge areas selection

6887
6888
6889
6890
6891
6892
6893
6894
6895
6896

The user is now able to indicate for each of the selected knowledge areas to, either or both, indicate what skill level would be required, and what dispositions are relevant. Suppose that the user indicates that she is willing to provide specifications for the knowledge area System Fundamentals. In Figure G.7, the skill level is specified by using a slider, and the disposition is specified by choosing from a menu.

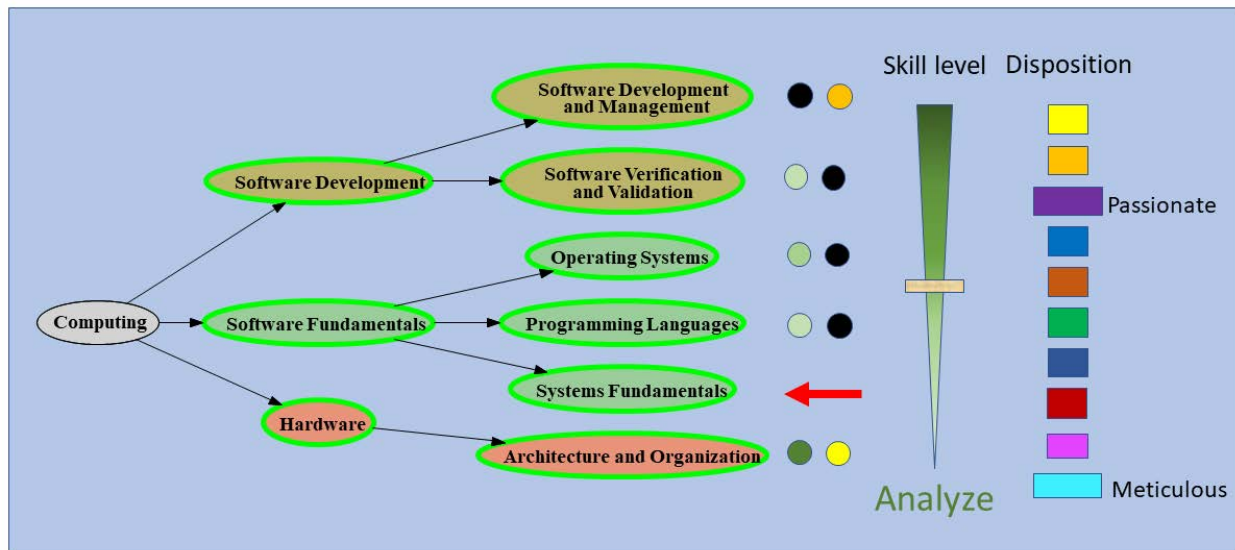


Figure G.7. Detailing skill and disposition

6897
6898
6899
6900
6901
6902
6903
6904
6905
6906

When all relevant specifications for the selected knowledge areas have been provided, the system generates a radar chart comparing the knowledge level for selected curricula. The distance from the center indicates the skill level related to each knowledge category. Figure G.8 compares CS and IT. The radar chart has been augmented with the specification from the user. In the example, it seems IT is the best match for the user's required knowledge levels. This is because there is a complete coverage of the user's specifications and the curriculum content; that is, the blue CS surface completely overlaps the user's green specification surface.

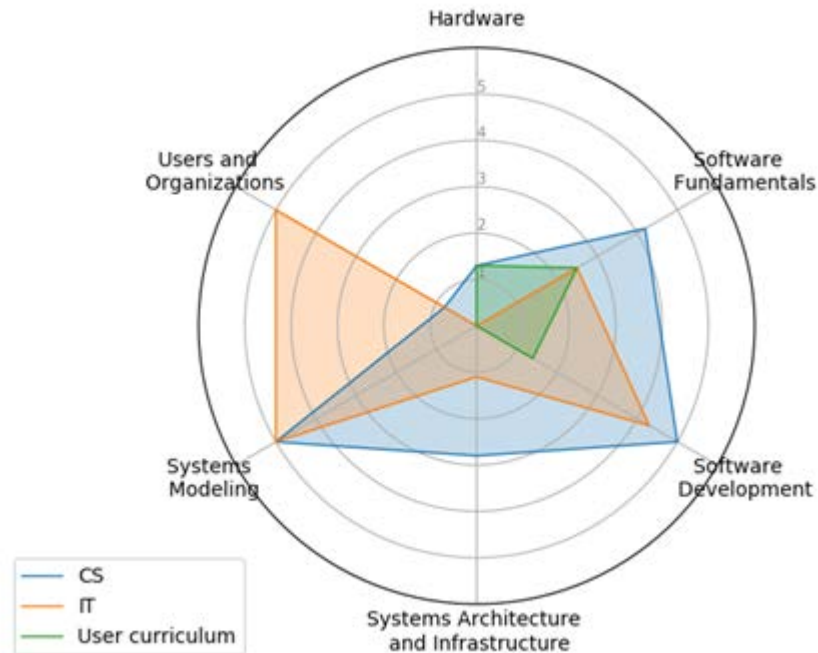


Fig. G.8. Comparison of CS and IT based on knowledge level

6907
6908
6909
6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924

G.1.3: Case 3: Question from Teacher

A teacher in a university faculty of Computing aims at developing a course for her domain “Human Factors in Computing”. Instead of a course, it could also be a textbook, an interactive electronic learning environment, or a mixture of these. The content of this course will be considered an essential part in the Bachelor curriculum for the departments IT, SE, and CS. She decides to find out what would be relevant for each of these curricular guidelines, in order to compose a course that will be sufficient for all departments.

Similar to the process that the student took in Figures G.1 and G.2 in Case 1, she decides to choose Software Fundamentals, Software Development, and Users and Organization as categories that seem relevant, and removes the other three categories. She then checks the knowledge areas for each of the chosen categories and chooses the areas that she believes to be relevant for her, resulting in Figure G.9.

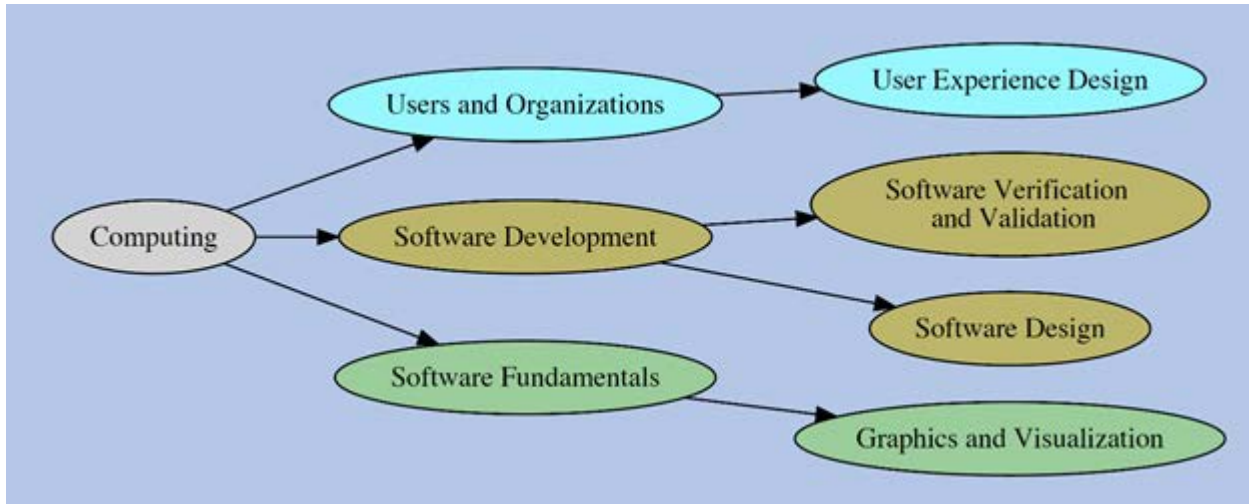


Figure G.9. Chosen knowledge areas for the new course design

6925
6926
6927
6928
6929
6930
6931
6932
6933
6934

For each of the chosen knowledge areas (User Experience Design; Software Verification and Validation; Software Design; and Graphics and Visualization), she will be able to find the relevant competency statements and dispositions from the chosen curriculum guidelines (IT, SE, and CS). Figure G.10 shows what she will get for the area User Experience Design after she chose the dispositions and competencies to keep for her course design. The process is the same for the other chosen knowledge areas.

Information Technology	Software Engineering	Computer Science
<p>Dispositions</p> <ul style="list-style-type: none"> Passionate <input checked="" type="checkbox"/> Professional <input type="checkbox"/> <p>Competencies</p> <p>A. Design an interactive application, applying a user-centered design cycle and related tools and techniques (e.g., prototyping), aiming at usability and relevant user experience within a corporate environment. Skill level: <input type="checkbox"/> Apply</p> <p>B. For evaluation of user-centered design, articulate evaluation criteria and compliance to relevant standards. Skill level: <input checked="" type="checkbox"/> Analyze</p>	<p>Dispositions</p> <ul style="list-style-type: none"> Passionate <input type="checkbox"/> Collaborative <input checked="" type="checkbox"/> <p>Competencies</p> <p>A. Design an interactive application, applying a user-centered design cycle with related tools and techniques (modes, navigation, visual design), to optimize usability and user experience based on an analysis of user needs within an environment. Skill level: <input type="checkbox"/> Apply</p>	<p>Dispositions</p> <ul style="list-style-type: none"> Passionate <input type="checkbox"/> Purpose-driven <input checked="" type="checkbox"/> <p>Competencies</p> <p>A. Create a simple application, together with help and documentation, that supports a graphical user interface for an enterprise and conduct a quantitative evaluation and report the results. Skill level: <input type="checkbox"/> Analyze</p> <p>B. Analyze and evaluate a user interface that considers context of use, stakeholder needs, state-of-the-art response interaction times, design modalities taking into consideration universal ... Skill level: <input checked="" type="checkbox"/> Evaluate</p>

Figure G.10. Potentially relevant competencies with their skill level, and dispositions for the User Experience Design area.

6935
6936
6937
6938
6939

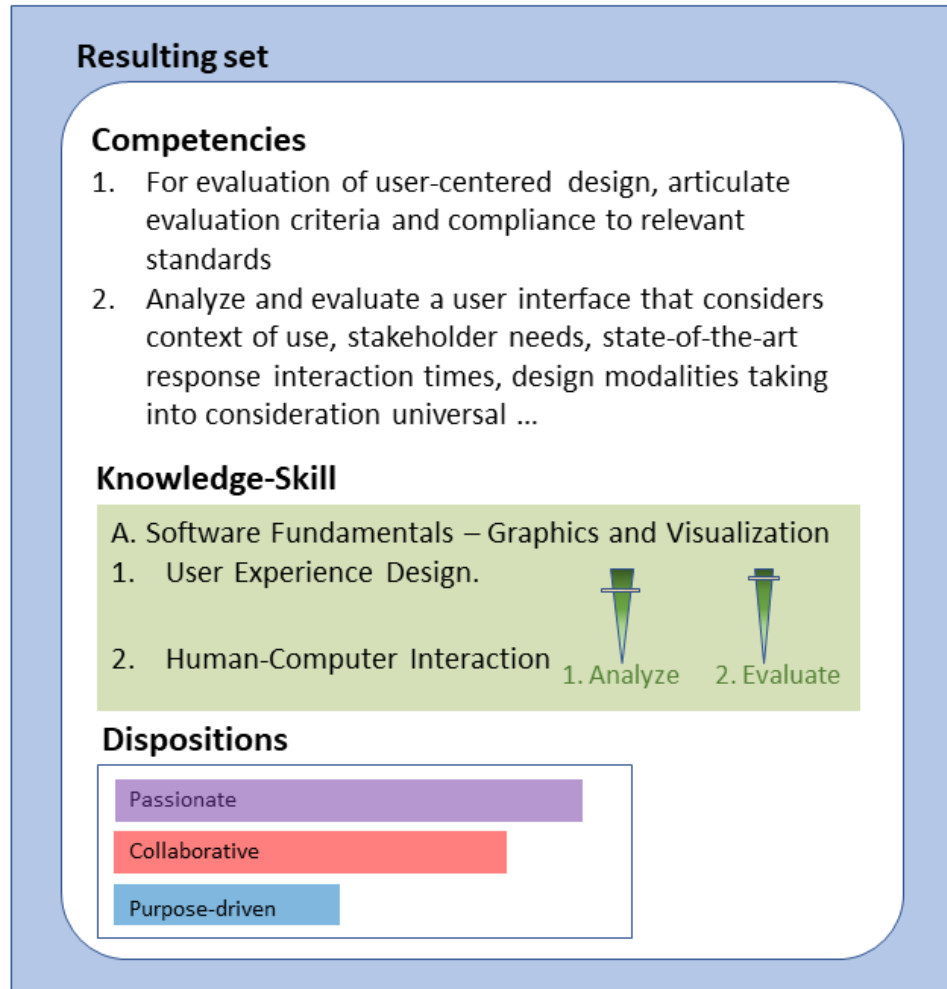


Figure G.11. Resulting chosen set of competencies and dispositions

6940
6941
6942
6943
6944
6945
6946
6947

Then the user may ask for an overview of the total set of chosen dispositions and competences with their skill levels (Figure G.11), and, if satisfied, for a PDF, to consider this as the User Experience part for her course to serve students from the different departments.

6948
6949
6950
6951
6952
6953
6954

The same procedure can help her to specify other parts of her course, in this example on Software Verification and validation, on Software Design, and on Graphics and Visualization. The user in this example might well consider finding adequate competences like Interactive Application Design at the Software Design knowledge area, and the disposition of Collaborative Attitude in the Software Verification and Validation knowledge area.

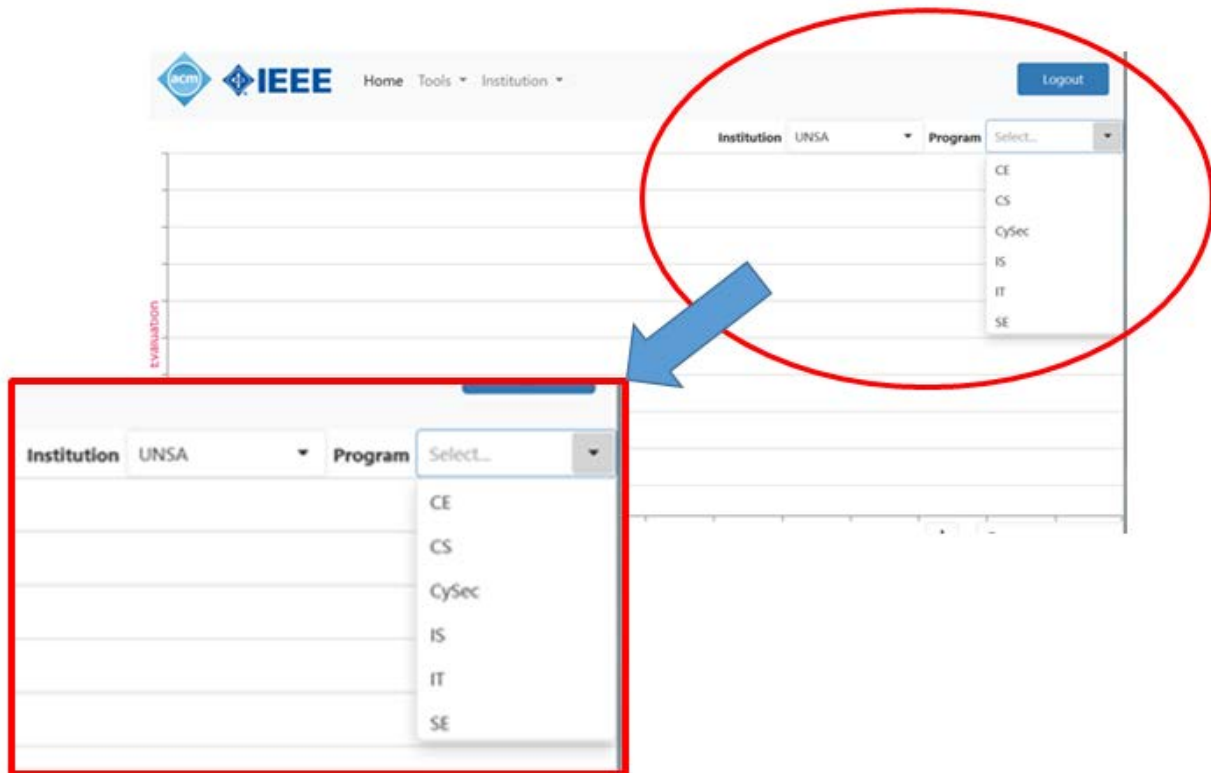
6955
6956

G.1.4: Case 4: Question from Educational Authority

6957
6958
6959
6960
6961

An official examiner on behalf of the Government’s Ministry of Education needs to assess or accredit a bachelor computing curriculum of one of the country’s public universities. We presume she has listed the names of all universities in her country or region. She might start by selecting the name of the university to be assessed, and choosing the program of one of the departments, e.g., Software Engineering (SE) (Figure G.12).

6962



6963

6964

Figure G.12. Indicating the Institution to be assessed and choosing the curriculum

6965

6966

6967

Figure G.13 shows the result: a graph that indicates the minimum and maximum weights of each knowledge element (given on the X axis) in the curriculum guidelines for SE. Below the graph, the six knowledge areas are displayed.

6968

6969

6970

6971

The user may expand each of these knowledge areas to access the individual knowledge elements. In Figure G.13, the user has expanded the Users and Organizations knowledge area resulting in the knowledge elements, e.g., Social Issues and Professional Practice, Security Policy and Management, etc. On the right side, the user has started inserting the actual weight for each element as found in the BA curriculum description of the department assessed. For each weight inserted, the interactive visualization will update the graph to show how the faculty scores compared to the guidelines. So, for example, in Figure G.13, as the institution values have not yet been input yet, all of the knowledge elements have the Evaluation value 0 (in the y-axis). In Figure G.14, the graph has been updated to reflect the input value, e.g., the value for Social Issues and the Professional Practice is 6.

6972

6973

6974

6975

6976

6977

6978

6979

6980

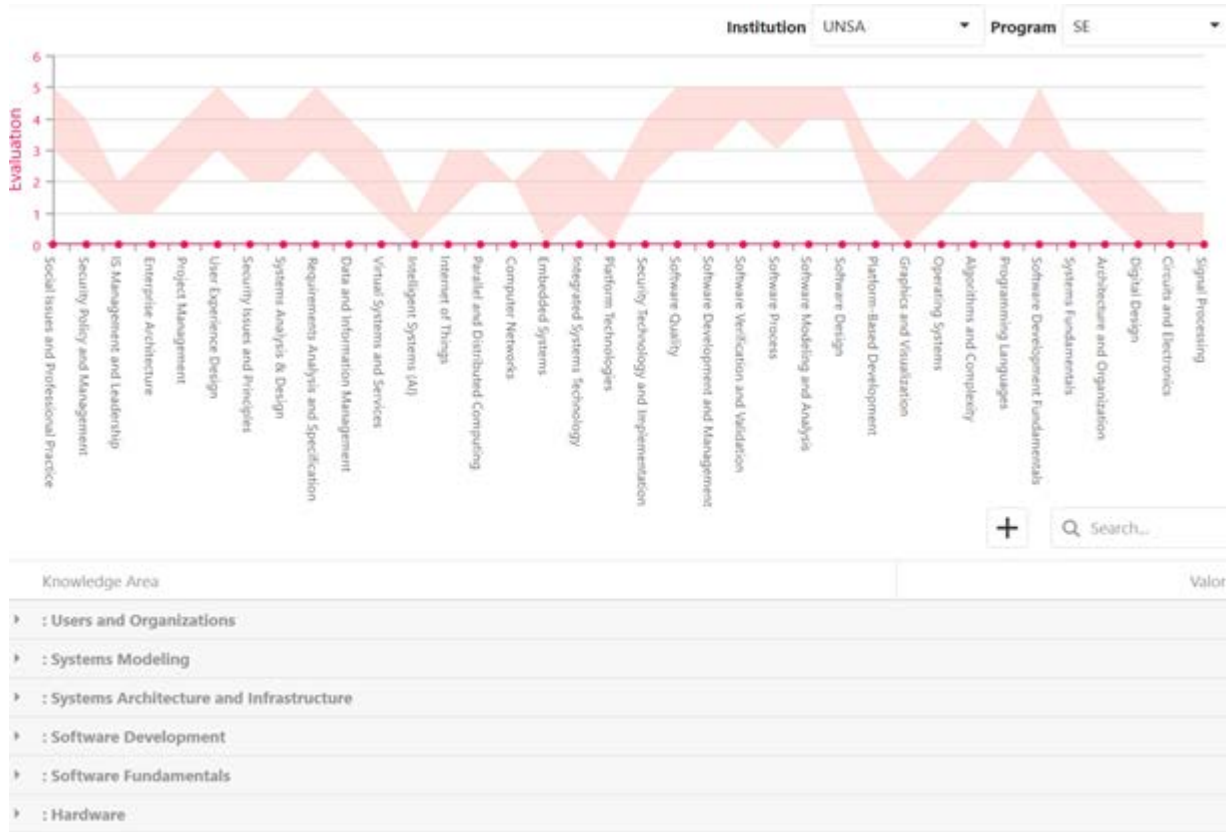


Figure G.13. Weight-range of knowledge areas in the curriculum guidelines for SE

6981
6982
6983
6984

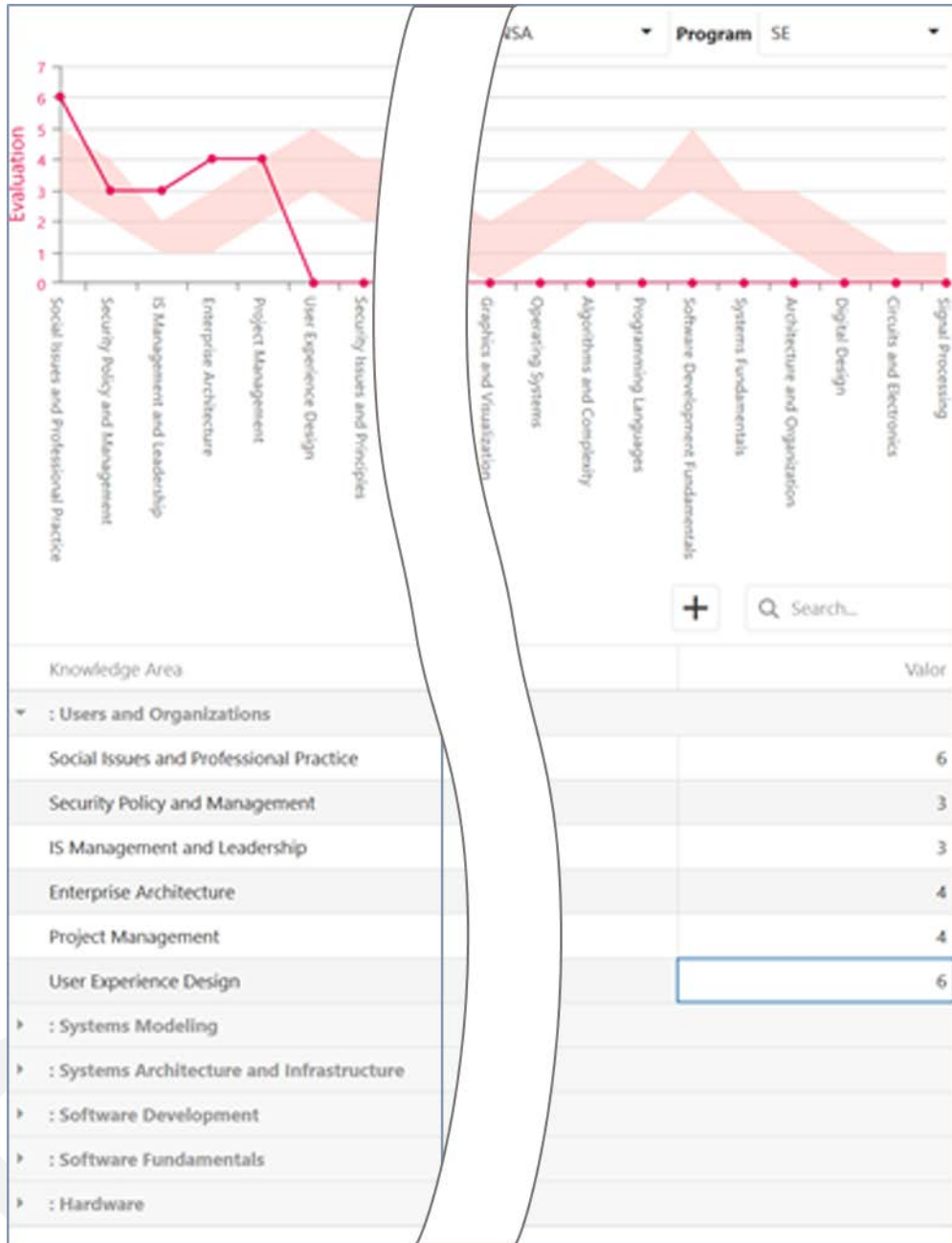


Figure G.14. Inserting weights found in the BA curriculum description of the department (note: middle part has been elided)

6985
6986
6987
6988
6989
6990
6991
6992
6993

When the user has finished this input process for all knowledge domains, the resulting comparison looks like Figure G.15, showing that this faculty generally conforms to the guidelines, is relatively strong in the domain of Users and Organization, and relatively weak on Software Development.

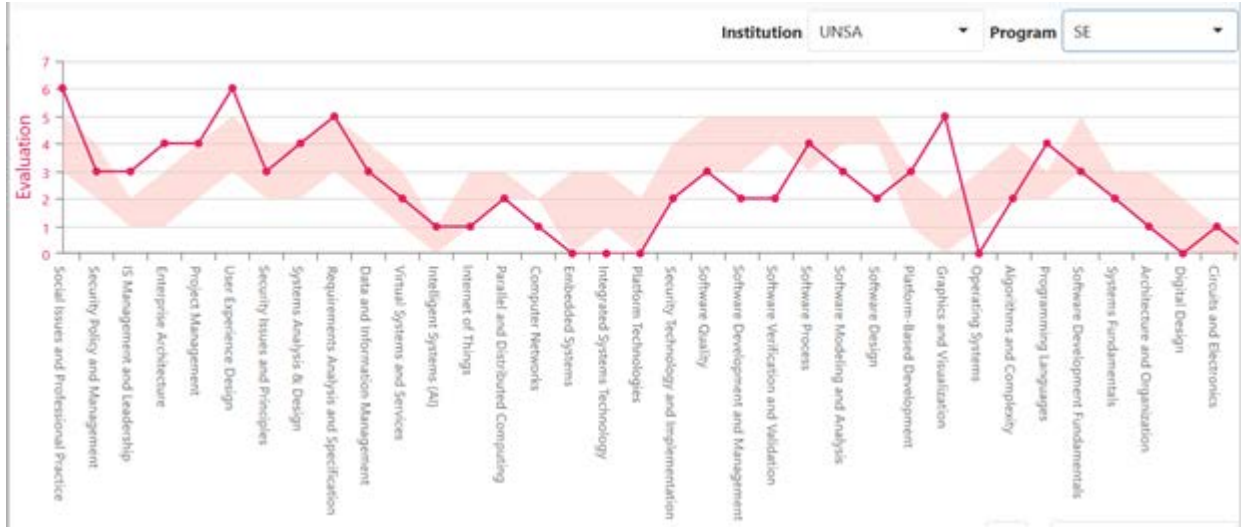


Figure G.15 Resulting state of the department's curriculum compared to the guidelines

6994
6995
6996
6997
6998
6999
7000
7001
7002
7003
7004
7005
7006
7007
7008

G.2: Comparison of Competency Specifications

Figure G.16 shows two competency specifications in table format side by side. Ref# and Title denotes the reference number and title, respectively, of a competency specification. The other three columns show the competency statement, dispositions, and knowledge-skill pairs for the competency specification. The colors show changes and similarities between the two competency specifications. For example, Disposition D-2 and Knowledge K(X-3) and K(X-4) are colored pink as they are the same. However, the corresponding skills for K(X-3) and K(X-4) are different so they are colored orange.

Ref#	Title	Competency Statement	Dispositions	Knowledge-Skill Pairs
CA1	lorem ipsum	something	D-1	K(X-1) B-3
			D-2	K(X-2) B-4
				K(X-3) B-3
				K(X-4) B-2
CA2	lorem ipsum	something	D-3	K(X-3) B-4
			D-2	K(X-4) B-3
				K(X-6) B-2

Figure G.16. Side by side comparison of competency specification
(Note: Values are example and not actual values.)

7009
7010
7011
7012
7013
7014
7015
7016
7017
7018
7019
7020
7021

G.3: Various Visualizations of Knowledge

Figures G.17, G.18, and G.19 are all visualizations of the same data, specifically the Knowledge table in Appendix D.

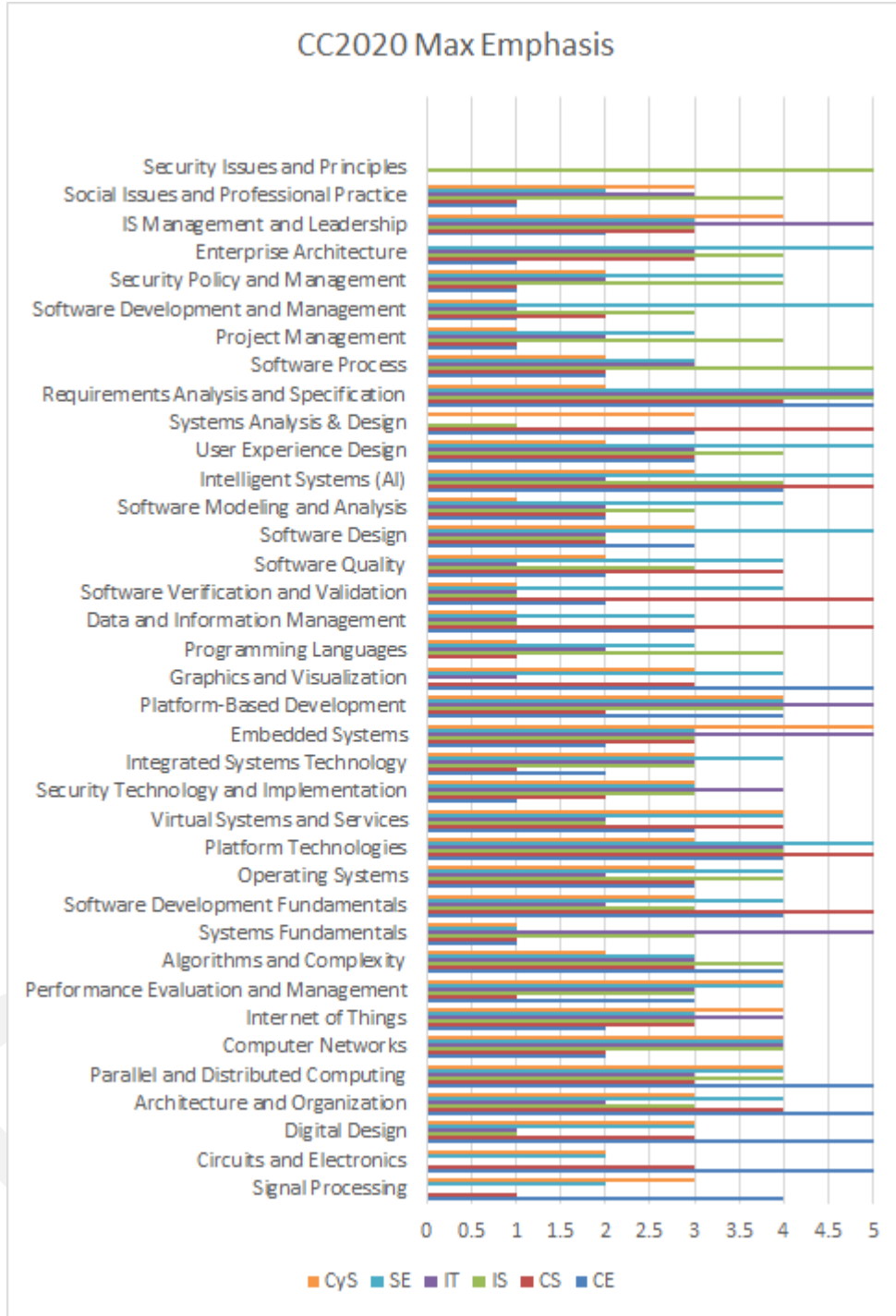
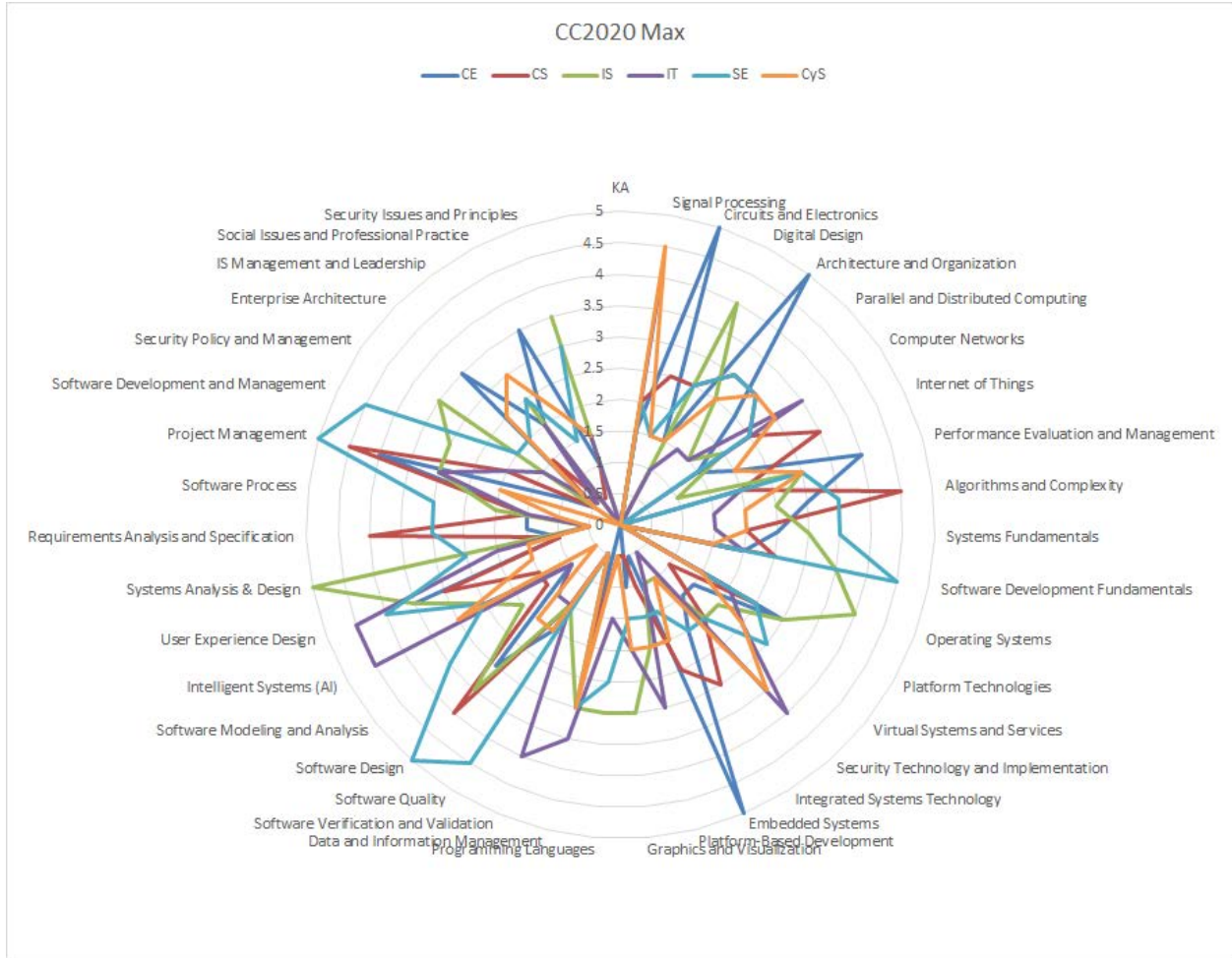


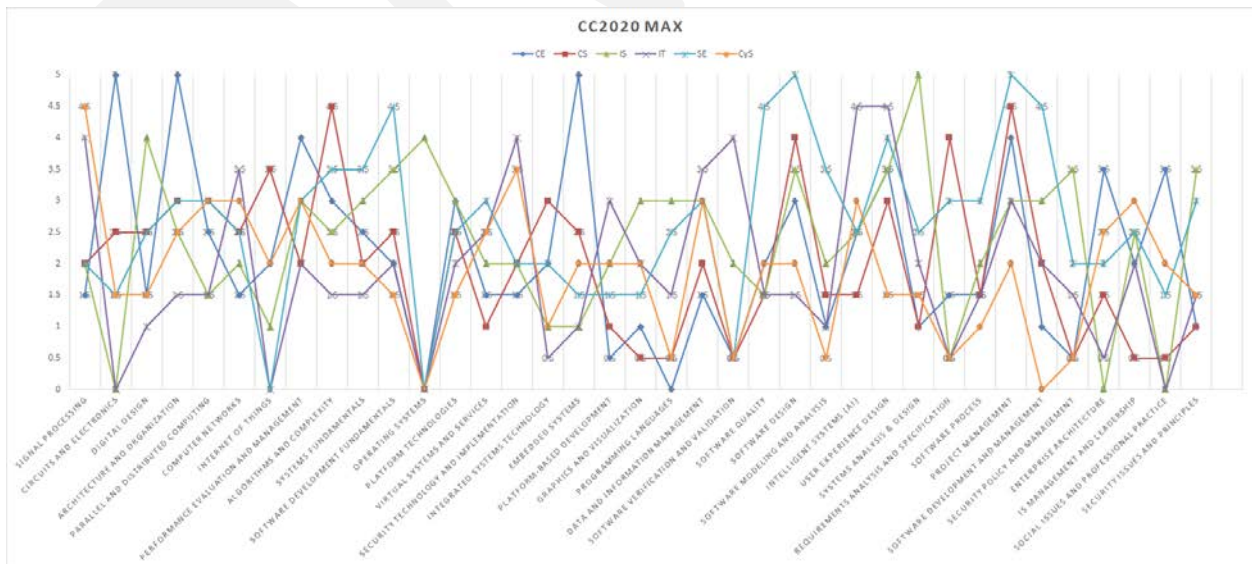
Figure G.17. Bar chart showing the maximum emphasis of knowledge areas

7022
7023
7024
7025
7026



7027
7028
7029
7030
7031

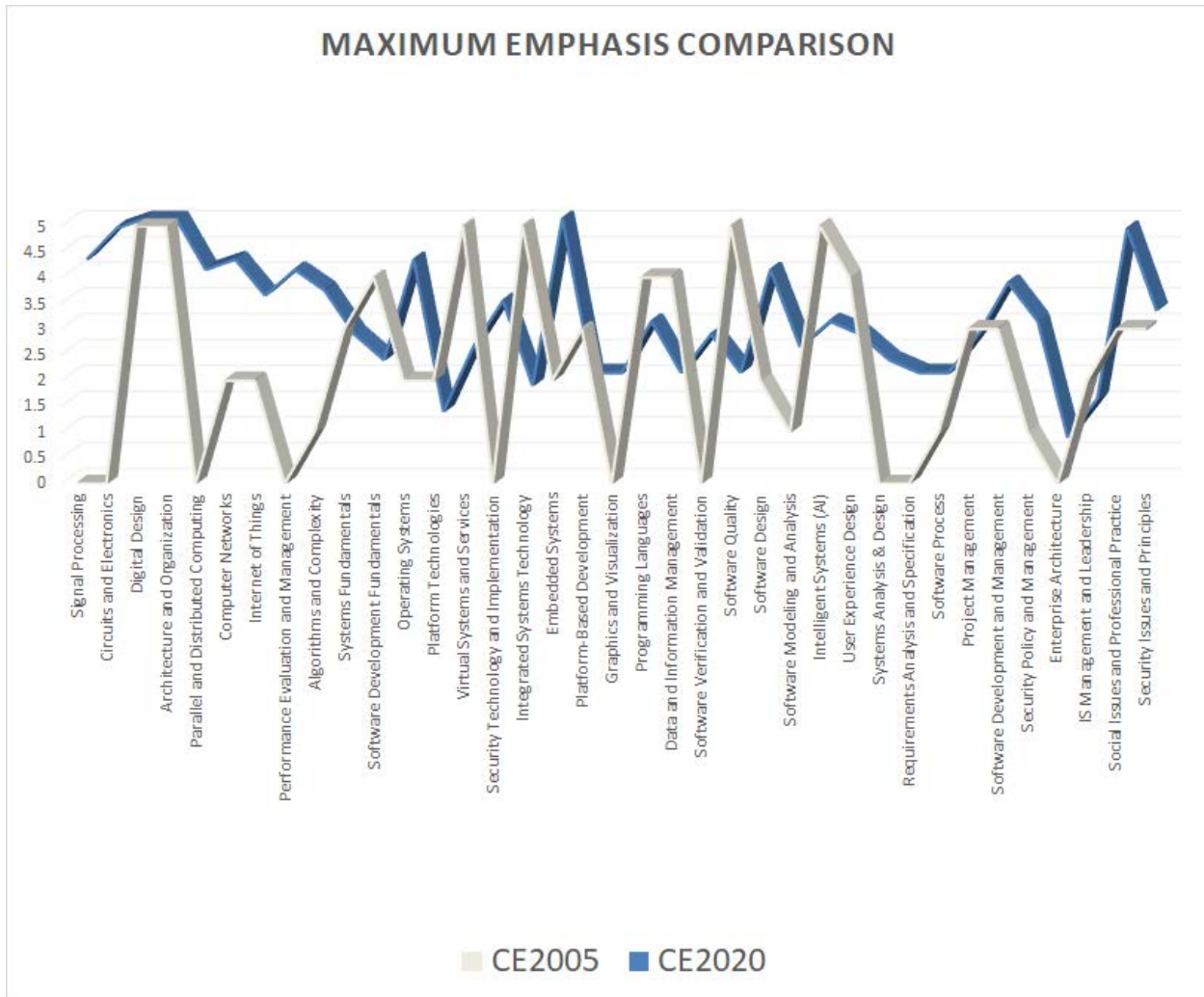
Figure G.18 Radar Chart showing maximum emphasis of knowledge areas



7032
7033

Figure G.19 Line Chart showing maximum emphasis of knowledge areas

7034
7035
7036



7037
7038
7039
7040
7041
7042
7043
7044

Figure G.20. Ribbon Chart comparing the maximum emphasis of knowledge areas between CE 2005 and CE 2020

Figure G.20 compares the CE knowledge area emphasis between the values that were given in CC2005 and CC2020. It shows that some of the knowledge areas, such as Signal Processing and Software Verification and Validation, had 0 emphasis indicating that they did not exist in CE2005.

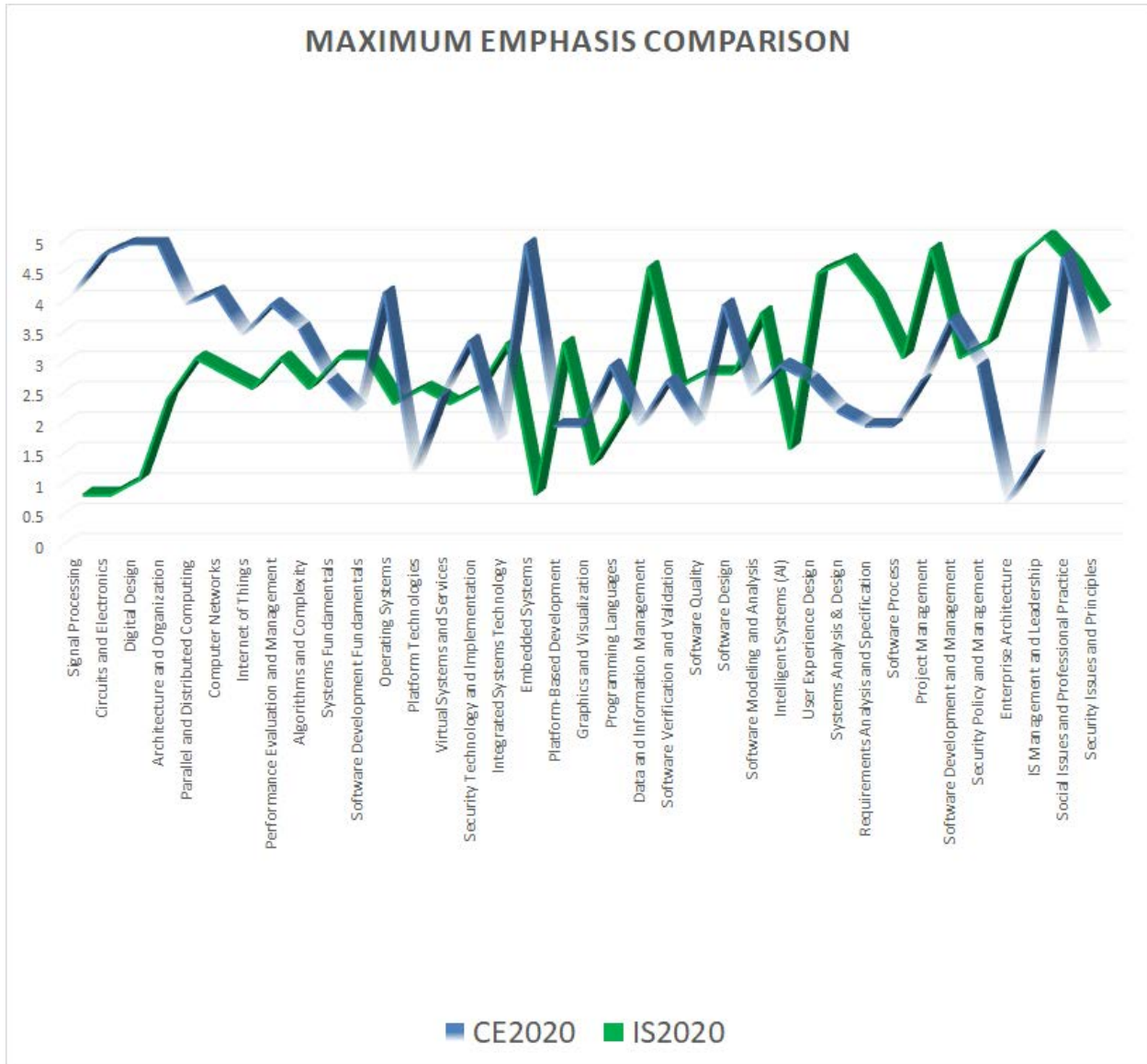


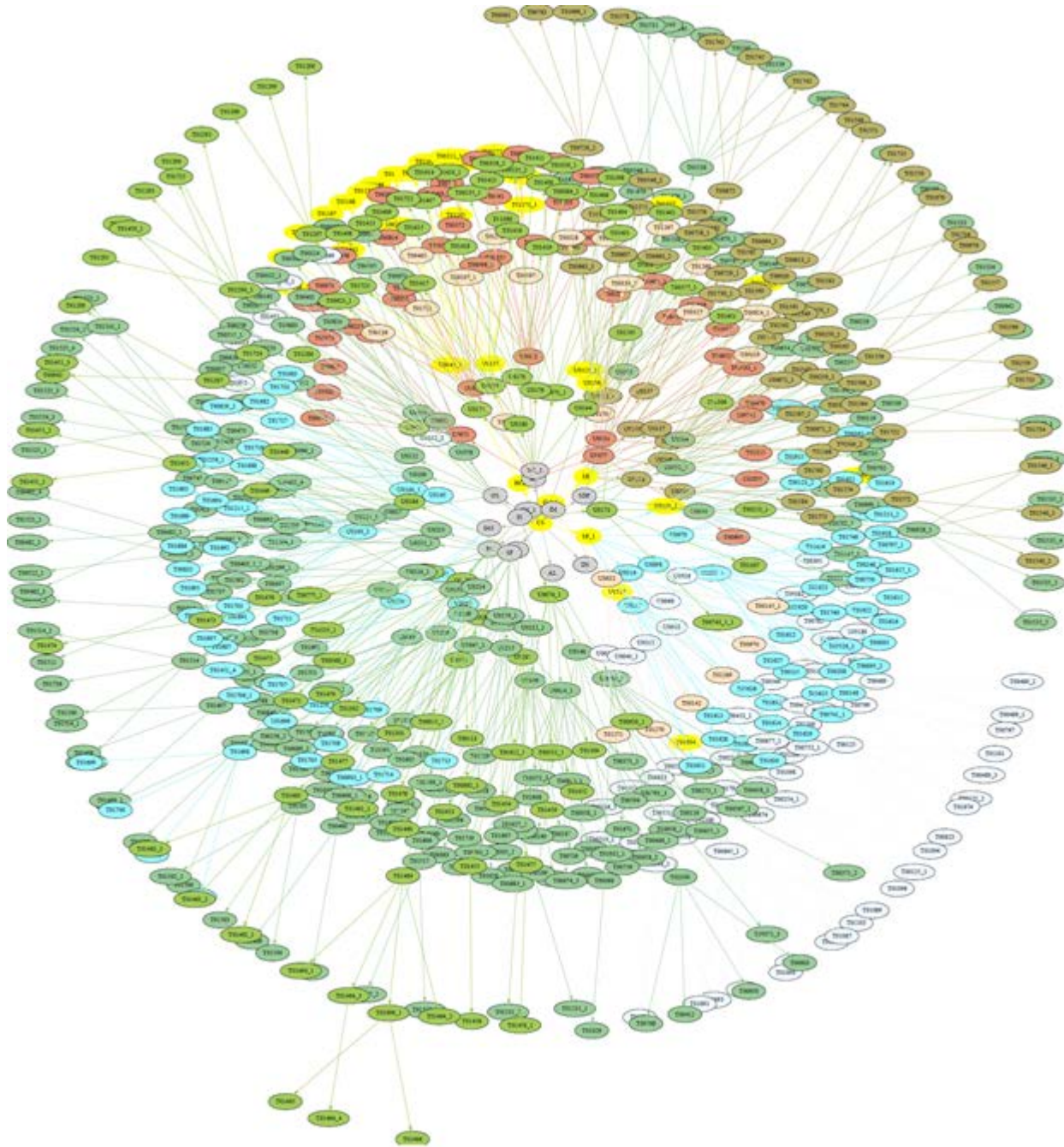
Figure G.21. Ribbon Chart comparing the maximum emphasis of knowledge areas between CE 2020 and IS 2020

Figure G.21 shows a comparison between the emphasis on knowledge areas for the curriculums CE and IS (in 2020). It shows a systematic difference at the left in the region of Hardware and Software Fundamentals where CE has a strong emphasis. At the other end of the graph, IS is emphasizing knowledge on Users and Organization more than CE.

G.4: Visualizing Full Curricula

The visualization in Figure G.22 centers around the CS node that links the knowledge areas (KAs), their core knowledge units (KUs) and their respective topics [Mar2]. KA nodes are near the center and colored gray. KU nodes are placed just outside of the KA nodes having labels starting with U, and topics nodes are

7063 placed in the outer part having labels starting with T.
7064
7065
7066



7067
7068
7069
7070
7071
7072
7073
7074

Figure G.22. Graph-based structure of the core components of CS2013.

To provide a detailed example, consider the highlighted parts in yellow. These yellow nodes represent the core aspects of the CS2013 curriculum specification that relates to User Experience Design (in CS2013 labelled “HCI”).

7075

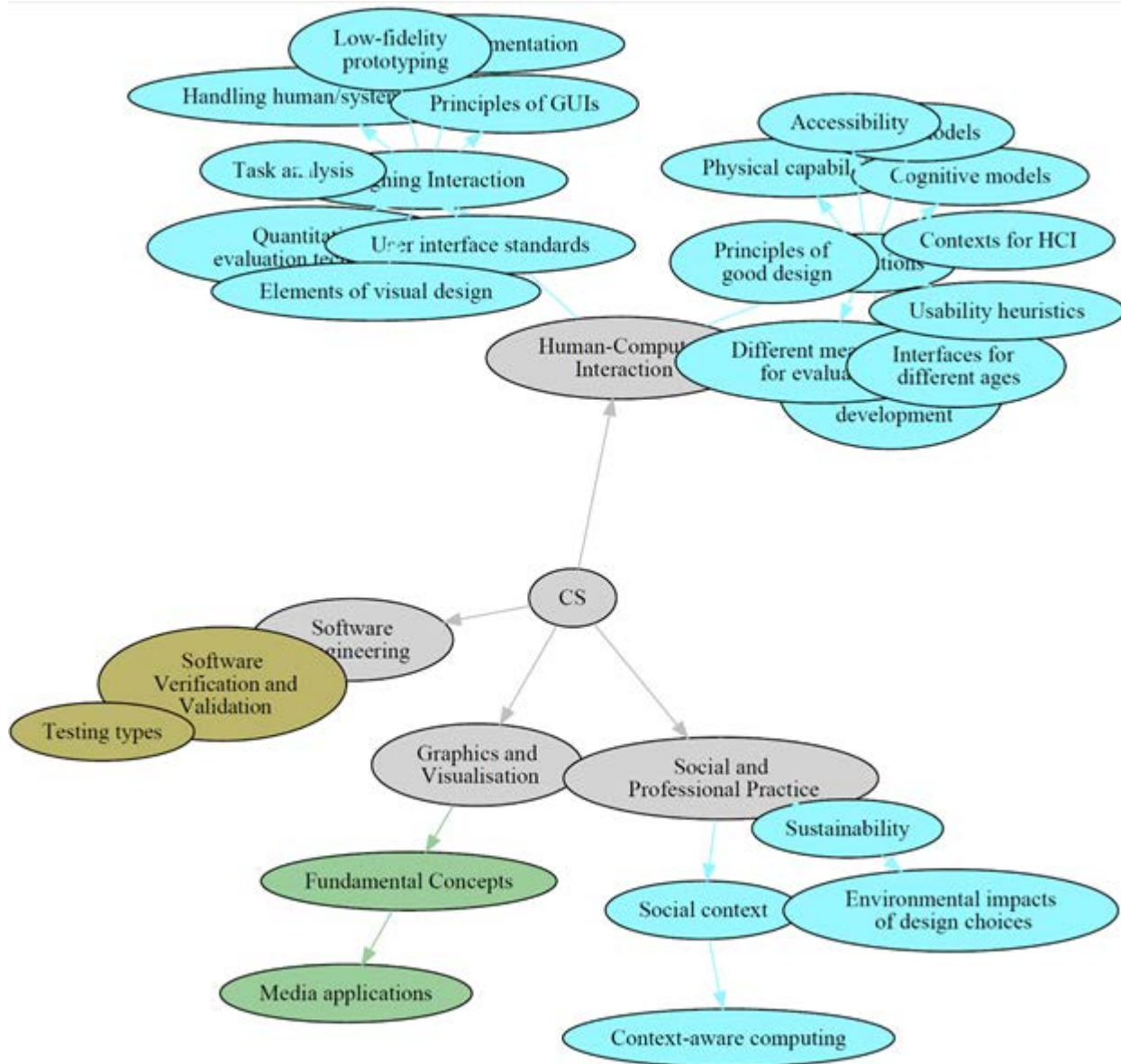


Figure G.23. Close-up of the HCI part of Figure G.22

7076
7077
7078
7079
7080
7081
7082
7083
7084
7085
7086
7087
7088
7089
7090
7091

Figure G.23 shows the actual labels of these nodes. Note that this includes not only HCI (Human-Computer Interaction), but non-HCI areas such as Software Engineering, Graphics and Visualization, and Social and Professional Practice. This is because the CS2013 curriculum mentions a link from these areas to HCI aspects. For example, the part concerned with Testing types which belongs to Software Engineering states “Testing types, including human computer interface, usability, reliability, security, conformance to specification (cross-reference IAS/Secure Software Engineering)” [p.182, CS2013]. As “human computer interface” is concerned with HCI, this part of Software Engineering is included in Figure G.23.

7092
7093
7094
7095
7096
7097
7098
7099
7100
7101
7102
7103

G.5: References for Appendix G

- [Tak5] E. Cuadros-Vargas, J. Impagliazzo, S. Gordon, L. Marshall, H. Topi, G. van der Veer, L. Waguespack, "Toward Understanding Computing Curricula: Visualization Using Competency within the CC2020 Project," *Educ. Inf. Technol.*, 2020.
- [Mar2] Marshall, Linda (2014) A graph-based framework for comparing curricula. Ph.D. thesis, University of Pretoria, South Africa.

DRAFT

7104
7105
7106
7107
7108
7109
7110
7111
7112
7113
7114
7115
7116
7117
7118
7119
7120
7121
7122
7123

Appendix H: Glossary and Nomenclature

This report has been written for the global community of educators, industry, students, and the general public. However, across the world in computing, different terms can be used to mean the same thing, and the same terms can have different meanings. While it would be ideal to have a consistent naming system globally, the task force recognizes that many terms are entrenched in a country’s or region’s culture. In an attempt to ensure transparency and readability, a list has been compiled for terms that could be confusing. A set of CC2020 definitions are summarized in this appendix and have been translated into the most common languages in the world. Hopefully, this list will enable the reader to understand the terminology used in computing in their own language around the world.

H.1: CC2020 Report Definitions

Table H.1 lists the working definitions that appear in this report, as compiled by the CC2020 task force.

Table H.1 Definitions for CC2020

Term	CC2020 Definitions
Accreditation	Official approval given by an organization stating that somebody or something has achieved a required standard
Algorithm	A set of rules to be followed in calculations or other problem-solving operations
AP	Advanced placement not used outside the USA
Baccalaureate	A bachelor’s degree
Chair of Department	Head of Department or Chair of Department
Class	A group of students studying the same course or degree
College	Outside the USA it can be another name for a High School or an organisational unit in a university, in the USA it is a term for a university
Community College	Two-year school post high school primarily used in the USA, very rarely used elsewhere
Competency	Knowledge + Skills + Dispositions in context
Core course/curriculum	Compulsory courses towards a degree
Course	A component of a degree or in some countries a whole degree
Credit hours	The number of hours for each credit towards a degree
Credits	The points a student receives after passing the assessments towards the course or degree
Curriculum	All the different courses of study that are taught in a school or for a particular subject
Engineering	Concerned with the design, building, and use of something. It does not imply a title of engineer
Faculty	Teachers in a university
Freshman	Freshman a term for a first year degree student, generally common

Term	CC2020 Definitions
Graduate, Post-Graduate	Graduate – has completed a Bachelor’s degree - Post Graduate – Masters and Doctoral degrees
Informatics	European term for computing or sometimes information systems or computer science
Information and Communication Technology (ICT)	A fairly common global term for the "computing" technology industry as a whole. Used in some places interchangeably with Information Technology
Information Technology (IT)	A branch of "Computing" with an approved curriculum. A fairly common global term for the "computing" technology industry as a whole. Used in many places interchangeably with Information and Communication Technology
Junior	USA term for a third year student
K-12	Kindergarten to year 12 (not used outside the USA and Canada)
Lecturer	A teacher in a university
Lecturer	Teacher in a university
Middle School	Also known as intermediate school. Different meanings in different countries generally two or three year schools at either 4 to 6, year 7 to 8 or 9, or 11 - 14.
Module	Either a course or a part of a course
Paper	Usually a product a student produces to pass a course or examination
Professor or a visiting professor	A Visiting Professor or in some countries (USA) a part-time Professor
Program(me)	All the courses that make up a degree
Quarter	A quarter of an academic year
Semester	Half an academic year
Senior	USA term for a fourth year student
Sophomore	USA term for a second year student
Subject	Similar to a course but not always used at university level
Technology	The application of scientific knowledge for practical purposes, especially in industry
Trimester	One third of an academic year
Undergraduate	Studying towards a bachelor’s degree

7124
7125

7126
7127
7128
7129
7130
7131
7132
7133
7134

H.2: Definitions/Nomenclature on a Global Scale

Tables H.2 provides translations for the CC2020 working definitions into Arabic, Hindi, Japanese, Chinese and Russian. Similarly, Table H.3 translates the list of working definitions into French, Italian, German, Spanish (Latin American and European), and Portuguese.

Table H.2 Definition Equivalents for Arabic, Hindi, Japanese, Chinese, and Russian

Term	CC2020 Definitions	Arabic	India (Hindi)	Japanese	Chinese	Russian
Accreditation	Official approval given by an organization stating that somebody or something has achieved a required standard	اعتماد أكاديمي	मान्यता	認定	学科评估	Аккредитация
Algorithm	A set of rules to be followed in calculations or other problem-solving operations	خوارزمية	विधि	アルゴリズム	算法	Алгоритм
AP	Advanced placement not used outside the USA	NA		アドバンスド・プレ イスマメント	大学先修課	Not used
Baccalaureate	A bachelor's degree	بكالوريوس	स्नातक	学士	学士	Бакалавриат
Chair of Department	Head of Department or Chair of Department	رئيس القسم	विभागाध्यक्ष	学部長 or 学科長	系主任	Заведующий кафедрой
Class	A group of students studying the same course or degree	صف دراسي	कक्षा	クラス	班级	Группа, поток (several groups at a big lecture)
College	Outside the USA it can be another name for a High School or an organizational unit in a university, in the USA it is a term for a university	كلية	महाविद्यालय	大学	学院	Not used in the sense
Community College	Two-year school post high school primarily used in the USA, very rarely used elsewhere	كلية المجتمع		NA	大专	Not used
Competency	Knowledge + Skills + Dispositions in context	كفاءة	योग्यता	コンピテンシ	胜任力	Компетенции

Term	CC2020 Definitions	Arabic	India (Hindi)	Japanese	Chinese	Russian
Core course/curriculum	Compulsory courses towards a degree	مقررات أساسية / خطة دراسية	पाठ्यक्रम	必修コース/カリキュラム	核心课程/课程体系	Обязательные курсы
Course	A component of a degree or in some countries a whole degree	مقرر دراسي	विषय	コース	课程	Курс
Credit hours	The number of hours for each credit towards a degree	ساعات معتمدة		単位取得時間	学时	Часы
Credits	The points a student receives after passing the assessments towards the course or degree	نقاط مكتسبة		単位	学分	Кредиты
Curriculum	All the different courses of study that are taught in a school or for a particular subject	خطة دراسية	पाठ्यक्रम	カリキュラム	课程体系	Учебный план
Engineering	Concerned with the design, building, and use of something. It does not imply a title of engineer	هندسة	अभियांत्रिकी	エンジニアリング or 工学	工程	Разработка, проектирование
Faculty	Teachers in a university	عضو هيئة تدريس	संकाय	講師 or 教員 (or 学部)	学部	ППС (abbreviation) Профессорско-преподавательский состав
Freshman	Freshman a term for a first year degree student, generally common	طالب السنة الجامعية الأولى		1年生	一年级学生	Первокурсник
Graduate, Post-Graduate	Graduate – has completed a Bachelor's degree - Post Graduate – Masters and Doctoral degrees	خريج، طالب الدراسات العليا	स्नातक, स्नातकोत्तर	学部卒, 大学院卒	毕业、研究生院	Выпускник, Аспирант
Informatics	European term for computing or sometimes information systems or computer science	المعلوماتية	सूचना विज्ञान	情報学	情報学	Информатика
Information and Communication Technology (ICT)	A fairly common global term for the "computing" technology industry as a whole. Used in some places interchangeably with Information Technology	تقنية المعلومات والاتصالات	सूचना और संचार प्रौद्योगिकी	情報通信技術	信息和通信技术	Информационно-коммуникационные технологии ИКТ

Term	CC2020 Definitions	Arabic	India (Hindi)	Japanese	Chinese	Russian
Information Technology (IT)	A branch of "Computing" with an approved curriculum. A fairly common global term for the "computing" technology industry as a whole. Used in many places interchangeably with Information and Communication Technology	تقنية المعلومات	सूचना प्रौद्योगिकी	情報技術	信息技术	Информационные технологии
Junior	USA term for a third year student	طالب السنة الجامعية الثالثة		3年生	三年级学生	No special term for this
K-12	Kindergarten to year 12 (not used outside the USA and Canada)	مراحل التعليم العام			K-12	Детский сад to year 7
Lecturer	A teacher in a university	محاضر	व्याख्याता	講師 or 教員	讲师	Лектор
Middle School	Also known as intermediate school. Different meanings in different countries generally two or three year schools at either 4 to 6, year 7 to 8 or 9, or 11 - 14.	المرحلة المتوسطة	माध्यमिक विद्यालय	中学校	中学	Средняя школа 7-15/16
Module	Either a course or a part of a course	وحدة		NA	模块	Раздел, модуль
Paper	Usually a product a student produces to pass a course or examination	ورقة أو مقالة علمية	परीक्षा	試験答案やレポートなどの成果物	论文	Работа, документ
Professor or a visiting professor	A Visiting Professor or in some countries (USA) a part-time Professor	أستاذ جامعي أو أستاذ زائر	प्राध्यापक	非常勤教授 or 非常勤講師	访问教授	No special term for this
Program(me)	All the courses that make up a degree	برنامج دراسي		プログラム	培养方案	Специальность
Quarter	A quarter of an academic year	NA	तिमाही	学期 (クォーター)	NA	Not used
Semester	Half an academic year	فصل دراسي	छमाही	学期 (セメスター)	学期	Семестр
Senior	USA term for a fourth year student	طالب السنة الجامعية الرابعة		4年生	四年级学生	Старшекурсник
Sophomore	USA term for a second year student	طالب السنة الجامعية الثانية		2年生	二年级学生	No special term for this

Term	CC2020 Definitions	Arabic	India (Hindi)	Japanese	Chinese	Russian
Subject	Similar to a course but not always used at university level	موضوع	विषय	科目	科目	Предмет
Technology	The application of scientific knowledge for practical purposes, especially in industry	التقنية	प्रौद्योगिकी	テクノロジー	技术	Технология
Trimester	One third of an academic year	NA		NA	NA	Not used
Undergraduate	Studying towards a bachelor's degree	طالب المرحلة الجامعية	पूर्वस्नातक	学部生	本科生	Студент

7135
7136
7137

7138
7139
7140

Table H.3 Definition Equivalents for French, Italian, German, Spanish (Latin America), Spanish (Europe), and Portuguese

Term	CC2020 Definitions	French	Italiano	German	Spanish Latin America	Spanish Europe	Português
Accreditation	Official approval given by an organization stating that somebody or something has achieved a required standard	Accréditation	Accreditamento	Akkreditierung	Acreditación	Acreditación	Credenciamento
Algorithm	A set of rules to be followed in calculations or other problem-solving operations	Un algorithme	algoritmo	Algorithmus	Algoritmo	Algoritmo	Algoritmo
AP	Advanced placement not used outside the USA			fortgeschrittene Platzierung	Not used	Not used	Validacao de creditos
Baccalaureate	A bachelor's degree	Baccalauréat (to Enter university)	Laurea	Bachelor, Bachelorabschluss	Also "Estudios de grado"	Also "Estudios de grado"	Bacharelado
Chair of Department	Head of Department or Chair of Department	Doyen Directeur Responsable de département	Direttore del dipartimento	Abteilungsleiter (m)/ Abteilungsleiterin (f)	Jefe de departamento	Director de departamento	Chefe de Departamento
Class	A group of students studying the same course or degree	Une classe	Classe	Klasse	Clase	Clase	Turma
College	Outside the USA it can be another name for a High School or an organizational unit in a university, in the USA it is a term for a university	Université	Collegio	Hochschule (Polytechnic)	Universidad	Universidad	Faculdade ou Instituto de Tecnologia

Term	CC2020 Definitions	French	Italiano	German	Spanish Latin America	Spanish Europe	Português
Community College	Two-year school post high school primarily used in the USA, very rarely used elsewhere	Classe préparatoire (to enter prestigious schools) Brevet de Technicien du Supérieur Institut Universitaire de Technologie	Centro di formazione	NA	Not used	Not used	Curso Técnico Profesionalizante
Competency	Knowledge + Skills + Dispositions in context	Compétences	Competenza	Kompetenz	Competencia	Competencia	Competência
Core course/curriculum	Compulsory courses towards a degree	Cours du tronc commun	Corsi obbligatori	Kernpflichtfach (core course)/ Kerncurriculum (core curriculum)	Asignaturas básicas	Asignaturas obligatorias	Disciplinas obrigatórias
Course	A component of a degree or in some countries a whole degree	Un cours	Insegnamento	Studiengang, Lehrgang	Curso	Curso	Disciplina
Credit hours	The number of hours for each credit towards a degree	Le temps de présentiel	Ore corrispondenti ad un credito formativo	Semesterwochenstunden	Horas por crédito	Horas por crédito	Hora-aula
Credits	the points a student receives after passing the assessments towards the course or degree	Les crédits ou ECTS	Crediti	Leistungspunkte	Créditos	Créditos	Créditos
Curriculum	All the different courses of study that are taught in a school or for a particular subject	Contenu pédagogique Programme pédagogique	Curriculum	Studienplan/ Lehrplan	Plan de estudios	Plan de estudios	Currículo
Engineering	Concerned with the design, building, and use of something. It does not imply a title of engineer	Ingénierie	Ingegneria	Technik	Ingeniería	Ingeniería	Engenharia

Term	CC2020 Definitions	French	Italiano	German	Spanish Latin America	Spanish Europe	Português
Faculty	Teachers in a university	Faculté / Institut /École (institution) Enseignant-Chercheur (people)	Collegio dei professori	Kollegium, Lehrkörper	Profesor	Profesor	Corpo docente
Freshman	Freshman a term for a first year degree student, generally common	Un étudiant de première année	Matricola	Studienanfänger (m)/ Studienanfängerin (f)	Estudiante de primer semestre	Estudiante de primer curso	Calouro
Graduate, Post-Graduate	Graduate – has completed a Bachelor’s degree - Post Graduate – Masters and Doctoral degrees	Licence (bac+3) Master (bac+5) Doctorat (bac+8)	laureato; laureato magistrale;	Studienabsolvent, Postgraduierter (m)/ Postgraduierte (f)	Graduado, Maestro, Doctor	Graduado, Post graduado	Graduado, Pos-graduado (Mestrado e Doutorado)
Informatics	European term for computing or sometimes information systems or computer science	Informatique (CS never Used in french)	Informatica	Informatik	Informática	Informática	Informática
Information and Communication Technology (ICT)	A fairly common global term for the "computing" technology industry as a whole. Used in some places interchangeably with Information Technology	Technologie de l’Information et de la Communication	Tecnologie dell’Informazione e della Comunicazione	Informations- und Kommunikationstechnologie	Tecnologías de la Información y la Comunicación	Tecnologías de la Información y la Comunicación	Tecnologia da Informação e Comunicação

Term	CC2020 Definitions	French	Italiano	German	Spanish Latin America	Spanish Europe	Português
Information Technology (IT)	A branch of "Computing" with an approved curriculum. A fairly common global term for the "computing" technology industry as a whole. Used in many places interchangeably with Information and Communication Technology	Technologie de l'Information	Tecnologie dell'Informazione	Informatik/ Informationstechnologie/ Informationstechnik	Tecnologías de la Información	Tecnologías de la Información	Tecnologia da Informação
Junior	USA term for a third year student	Un étudiant de troisième année		Student/ Studentin im 3. Studienjahr	Estudiante de tercer semestre	Estudiante de tercer curso	Veterano do terceiro
K-12	Kindergarten to year 12 (not used outside the USA and Canada)	Le primaire (3-10yo) Le secondaire au Collège (11-15yo)		vom Kindergarten bis zum Abitur	Educación preuniversitaria	Educación preuniversitaria	Educação Básica
Lecturer	A teacher in a university	Enseignant	Docente	Dozent (m) / Dozentin (f)	Profesor de tiempo completo	Profesor Titular	Professor
Middle School	Also known as intermediate school. Different meanings in different countries generally two or three year schools at either 4 to 6, year 7 to 8 or 9, or 11 - 14.	École maternelle (3-6) École élémentaire (6-10) Collège (11-15) Lycée (16-18)	Scuola media	Hauptschule (Year 5-9), Mittelschule (Year 5-10), Gymnasium (year 5-12)	Educación básica	Educación primaria (4-12)	Ensino Fundamental
Module	Either a course or a part of a course	Un module / une unité d'enseignement	Modulo	Modul	Módulo	Módulo	Módulo

Term	CC2020 Definitions	French	Italiano	German	Spanish Latin America	Spanish Europe	Português
Paper	Usually a product a student produces to pass a course or examination		Scritto: prodotto da uno studente per superare un esame	wissenschaftliche Arbeit	Prueba, examen	Examen, Trabajo, Artículo, Prueba	Artigo, if the last work of the degree is called Trabalho Final de Curso
Professor or a visiting professor	A Visiting Professor or in some countries (USA) a part-time Professor	Professeur invité	Professore	Professor, Gastprofessor (visiting professor)	Profesor visitante	Profesor visitante (visiting professor) Profesor asociado a tiempo parcial (part-time teacher)	Proferssor Titular ou Professor visitante
Program(me)	All the courses that make up a degree	Programme pédagogique	Manifesto degli studi	Studienplan/ Lehrplan (list of courses which make up degree), Studiengang (Study programme)	Plan de estudios, Programa Educativo	Plan de estudios	Currículo
Quarter	A quarter of an academic year	Un demi-semestre	Trimestre	ein Viertel eines akademischen Jahres	Not used	Not used	bimestre
Semester	Half an academic year	Un semestre	Semestre	Semester	Semestre	Semestre (often known as “Cuatrimestre” because classes last for 4 months + one of examinations)	Semestre
Senior	USA term for a fourth year student	Un étudiant de quatrième année		Student/ Studentin im 4. Studi enjahr	Estudiante de cuarto semestre	Estudiante de cuarto curso	Veterano do quarto (if last year Formando)
Sophomore	USA term for a second year student	Un étudiant de deuxième année		Student/ Studentin im 2. Studi enjahr	Estudiante de segundo semestre	Estudiante de segundo curso	Veterano do segundo ano
Subject	Similar to a course but not always used at university level	Un sujet	Materia	Fach	Asignatura	Asignatura	Matéria

Term	CC2020 Definitions	French	Italiano	German	Spanish Latin America	Spanish Europe	Português
Technology	The application of scientific knowledge for practical purposes, especially in industry	Une technologie	Tecnologia	Technologie	Tecnología	Tecnología	Tecnologia
Trimester	One third of an academic year	Un trimestre (3m) Un semestre (5m)	quadrimestre	Trimester	Trimestre	Trimestre	Trimestre
Undergraduate	Studying towards a bachelor's degree	La Licence (L1-L3)	non laureato	grundständiges Studium	Estudios de grado	Estudios de grado	Graduação

7141
7142
7143
7144

DRAFT

7145
7146
7147

7148 **Appendix I: Sustainable Computing and Engineering Competence** 7149 **in China**

7150
7151

7152 China and its education ministry have embraced competency as an important element in the development of computing
7153 and engineering programs. Over the past few years, publications emerged surrounding the importance of competency
7154 in computing and engineering education. The Forum of Chinese Twenty-Experts on Computing Education, in which
7155 more than twenty senior professors on computing have engaged, has recently published its “Blue Book” [Blu1] to
7156 address the need for competency in university environments, particularly as it applies to computing and engineering
7157 education programs. The China Computer Federation also emphasized computing education for competencies in its
7158 *2018 Future Computer Education Summit (FCES 2018)* publication [Imp1]. The first of these publications addresses
7159 the need for program agility in response to a rapidly changing technological world. The remainder of this section
7160 summarizes the “Blue Book” philosophy and ways China expects to adapt to technological change over the next dozen
7161 years. The second of these publications uses a modern approach that competency is a triad of knowledge, skill, and
7162 disposition as explained earlier in this report.

7163
7164

7165 **I.1: Adaptable and Sustainable Competencies**

7166

7167 Over 2017-2019, a working group of the mentioned forum has written a “Blue Book” titled, “Computing Education
7168 and Sustainable Competence” [Blu1]. The emergence of this work in China has opened new ideas in the transformation
7169 of university computing and engineering education in China. The emerging fields of information technology (IT) and
7170 artificial intelligence (AI) have created novel opportunities for industry and academia. The internet has made possible
7171 new modern services and businesses coupled with innovative applications. The emerging AI industry has provided
7172 fertile ground for new industrial sectors such as smart enterprises and public services. The new industrial revolution
7173 (i.e., Industry 4.0) promises advances in networked intelligent manufacturing, service-oriented manufacturing, and
7174 robotics for industry and modern services.

7175

7176 Change on such a global scale brings new challenges for an information society. Societal changes present challenges
7177 for a digitally networked cognitive society, for sustainable development of society and the environment, and for the
7178 transference of information and knowledge. People also change. Younger generations have new attitudes and demands
7179 for professional development that require multi-dimensional approaches to learning with sustainable competencies so
7180 they can adapt to an evolving future. That means education must also change—especially at the university level.

7181

7182 Educational challenges facing students include global competitiveness, massive open online courses (MOOCs) and
7183 the disruption caused by them, changes in university functions, and the educational reform is needed for an information
7184 society. These changes place a strain on learning systems and society. To address these challenges, it is important to
7185 adjust to changes in society by developing sustainable competencies for higher education. *Sustainable competency*
7186 refers to the ability to (a) adapt to change and competitiveness of the future society, (b) be creative based on the
7187 missions and technology, and (c) perform and promote social and technical development [Blu1 p7]. Trans-boundaries
7188 and the rapid changes of new economies require that computing and engineering talents have stronger sustainable
7189 competencies for the future.

7190

7191 In recent years, the Chinese government and universities have explored and practiced new reformations of
7192 contemporary innovations in higher engineering education. The Chinese Ministry of Education (MOE) has been
7193 developing “New Engineering” educational initiatives to cultivate people for the future development of new
7194 technologies and new economies. The MOE is promoting innovative and entrepreneurial education as well as
7195 university-industry collaborative educational projects to nurture students with a more innovative attitudes and a new
7196 consciousness for creativity and practical ability. Many Chinese universities have participated in such engineering
7197 education reform. This practice has led to an ongoing, competency-oriented movement in higher education.

7198

7199 An example of this movement is the “open loop university” at Stanford University. Under the new concept, students
 7200 would enter the “open loop” at multiple times throughout their professional life. Called *Stanford 2025*, this movement
 7201 presents a new concept of learning. Its ethos involves open education, purpose learning, and incremental development.
 7202 Its characteristic of paced education promises to transform four-year-systems into lifetime multi-year systems over
 7203 three phases that include calibrate, elevate, and activate. This axel flipped, self-fulfilling approach becomes a
 7204 continuing spiral of Knowledge \implies Ability, followed by Ability \implies New Knowledge. Such a purposeful way of
 7205 learning instills professional development as a driving force for learning.
 7206
 7207

7208 I.2: Agile Education for Sustainable Competencies

7209
 7210 Faced with multiple objectives and individualized demands for human development, it became important to create a
 7211 sustainable competency-oriented agile form of education. *Agile education* is an approach that combines theory,
 7212 knowledge, ability, and human quality into a comprehensive education system [Blu1 p25]. It is a method to stimulate
 7213 student interest by enhancing their potential creative ability and their ability to adapt to change. Agile education
 7214 realizes multiple iterative rounds of knowledge, learning, and promotional ability. It encourages the rapid learning of
 7215 theory, techniques, practice, and efficient coordinated education by multi-university and multi-domain educational
 7216 resources.
 7217

7218 The concept of agile education derived its inspiration from industrial concepts, agile manufacturing, and agile software
 7219 development from the 1990s. New product development and manufacturing models had to adapt to the rapid change
 7220 of new technologies, new products, and new demands. For example, agile manufacturing became an advanced
 7221 manufacturing technology with models appearing in the late 1990s. This strategy integrates agile virtual enterprise
 7222 alliances, advanced flexible production technologies, and high-quality workers to promote industrial competence. As
 7223 another example, agile software development emerged as a new customer demand-oriented software strategy that
 7224 applies iterative construction models and cyclic progression methods to promote development efficiency.
 7225

7226 Because of past successes, agile education is becoming a massive, customized mode of education that emphasizes the
 7227 cultivation of accurate outcomes and content teaching. Its intent is to decompose dramatically training ability, to
 7228 restructure content concurrently, and to rearrange content iteratively by means of advanced education networks and
 7229 resources from multi-universities and multi-sectors. Agile education seeks to develop accurate coordination
 7230 procedures for teaching and learning and to strengthen students’ ability through exploratory, active and gradual
 7231 learning procedures. Methods used include fostering university-industry cooperation for co-education, promoting
 7232 teamwork and interaction of interdisciplinary teachers and students, and integrating education at the university with
 7233 society.
 7234

7235 Sustainable competency-oriented agile education involves instilling sustainable competencies in students over a four-
 7236 year education experience. The learning perspective includes fundamental courses and general education followed by
 7237 technical and core courses, followed by interdisciplinary elective courses, and culminating with individual
 7238 development. A curriculum often defines this learning perspective. The practice perspective includes industry visits,

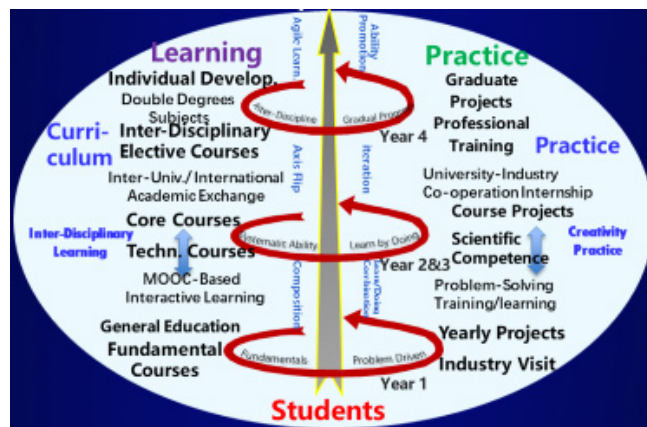


Figure H.1. Agile education and sustainable competencies

7239 yearly projects, course projects and scientific
7240 competence, professional training, industry
7241 internship and graduate (capstone) projects. The combination of learning and practice develops sustainable
7242 competencies in students, which is at the heart of agile education. Figure H.1 illustrates this concept. The practice of
7243 agile education derives from practical experience of the recent reform and innovation actions of world-wide higher
7244 computing and engineering education. Examples of such practices include the open-loop university, interconnection
7245 networked learning, MOOC-based online and offline learning, flipped classroom, concurrent learning and doing,
7246 project-based learning, university and industry cooperation, and innovation and entrepreneurship education.

(Courtesy of Prof. Xu Xiaofei)

I.3: Factors Affecting Agile Computing and Engineering Education

7249 In agile education, there is an emphasis on multiple cultivation objectives based on diversity of students. Its purpose
7250 is to develop “education-on-demand” through a combination of a major-oriented program with individualized
7251 learning. Universities should provide massive customized education systems through multiple cultivation objectives
7252 of its students. The teaching system would consist of a curriculum, teaching processes, teacher-student learning
7253 activities, resources, and quality evaluation methods. It is likely that universities would have to restructure their
7254 traditional education systems to establish multiple cultivation objectives, to construct a flexible composite curriculum,
7255 to develop iterative learning procedures, and to create a collaborative teaching support system. Considerations include
7256 multiple cultivation objectives, course and module classifications, systematic core courses, combining theory with
7257 practice, establishment of agile teaching and learning processes, and generating collaborative educational resources.

7260 In agile education, universities would classify their curricula and courses into multi-clusters of modules according to
7261 the needs of individualized students or groups of specialties and directions to provide an environment of adaptive
7262 knowledge learning and ability training. Course modules include general education clusters and fundamentals clusters,
7263 specialty core course modules, interdisciplinary course modules, elective course modules, and experimental practice
7264 clusters. It might be necessary to make flexible compositions of the course modules, courses, or micro-courses to
7265 adapt the programs to student needs since they have more choices for their development. Because of emerging
7266 technologies and applications, it would be necessary to reconstruct the core courses in developing a systematic and
7267 flexible curriculum. The universities would have to redesign course content and teaching methods because of the
7268 interdisciplinary learning, the iterated cultivation and development of creativity, and the adaptability of the students.

7271 Agile education is a multi-phase process of iterative learning. For computing programs, the *iterative learning* process
7272 begins with students entering a computing course to gain knowledge and to develop an ability for enhancement. The
7273 process includes:

- 7274 ■ To know basic concepts of computers and their systems (year 1 courses and yearly projects)
- 7275 ■ To understand and grasp systems, components, and techniques of computers (year 2 and year 3 courses as
7276 well as project-based learning)
- 7277 ■ To design and develop systems and applications for computers (year 4 courses, projects, and internships)

7278 Ultimately, students will gain further knowledge, technology, and a systematic understanding of computing through
7279 this three-round form of iterative learning and practice.

7281 As already mentioned, agile education is a
7282 combination of theory and practice. The emphasis
7283 is to combine course-based teaching and project-
7284 based learning as well as to enhance the hands-on
7285 ability of students through wider and deeper
7286 practical activities. Figure H.2 captures that
7287 philosophy. In addition to knowledge transfer and
7288 skill training, it is important to create opportunities
7289 and environments for students to develop and
7290 mature. Creating such opportunities for students is
7291 very important. Therefore, the educational
7292 procedure for agile education consists of two significant points. First, the teaching and learning procedure should be
7293 very flexible and agile to allow students to arrange their individual learning plan and course selection based on their
7294 own interests and characteristics. Second, teaching faculties should restructure the teaching and learning procedures



Figure H.2. Agile education: Combining theory with practice
(Courtesy of Prof. Xu Xiaofei)

7295 as well as management rules to accommodate the needs of the interdisciplinary learning, iterated student cultivation,
7296 and the individual development of students. Suggestions for doing this includes a robust credit system, a flexible
7297 teaching plan, interdisciplinary studies, cumulative study and examinations, and credit for creativity practice.
7298

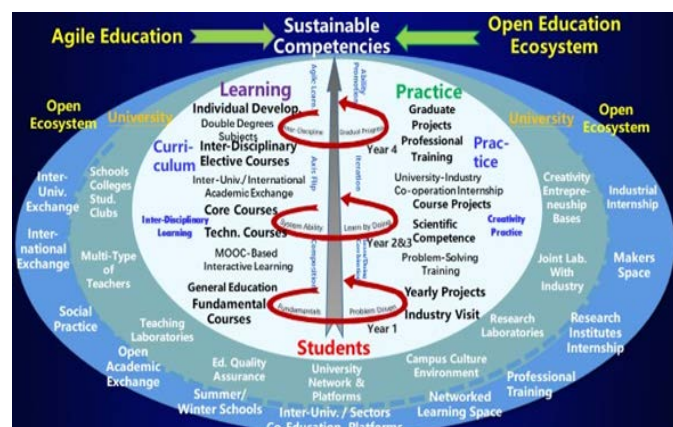
7299 Agile education requires collaborative organization and resources. To implement agile education, it is necessary to
7300 coordinate the teaching and learning organization and resources, including multidisciplinary teams, educational
7301 facilities from multi-schools and multi-universities, as well as training resource from industry and society. Universities
7302 should establish virtual inter-school and inter-university collaborative teaching centers for agile education and purpose
7303 learning by combining teaching resources from multi-institutions. It is also important to consider MOOC resources
7304 through inter-university collaborative teaching and learning approaches.
7305
7306

7307 **I.4: Open Education Ecosystems for Agile Education**

7308 It is important to address the role of university management to implement agile education. Universities need to reform
7309 their management and support systems as well as their ecosystems for agile education. Focusing on individual
7310 students, small groups, and flexible learning are key elements for a successful transition. Universities should build an
7311 advanced agile education system, reform and restructure their management and support systems, and build an open
7312 education ecosystem for sustainable competencies. Suggestions for doing this include (a) setting up a flexible study-
7313 term and a full credit system for iterated learning and individualized cultivation, (b) establishing an undergraduate
7314 supervisor system and small group learning for individualized cultivation and development of students, (c) developing
7315 micro-courses and small course modules for flexible composition of learning contents, (d) developing ability-oriented
7316 courses and learning units as learning models, and (e) establishing a university-industry collaborative education
7317 system that includes internships, creative projects, and entrepreneurship.
7318
7319

7320 It is also important to create support resources for agile education. This transformation to agile education requires
7321 abundant educational resources. These include advanced online and offline course resources, networked IT support
7322 platforms, laboratories for creative projects, entrepreneurship bases, student colleges, big data service platforms, and
7323 advanced infrastructure and facilities. In modern universities, they should build education support systems on IT
7324 enabled network platforms to provide intelligent, coordinative, precise, and efficient services for agile education.
7325

7326 Quality assurance is an important part of agile education. Universities should create an ability-oriented agile education
7327 quality evaluation and assurance system to measure educational processes and results and to set up effective report
7328 mechanisms for improving teaching and learning quality. Analysis of teaching and learning status is useful for
7329 dynamic assessment of process evaluation, phase evaluation, and comprehensive evaluation for iterative learning and
7330 improvement. Quality metrics for agile education include key performance indicators (KPIs) for attainment (assessing
7331 development results of students' ability compared to education proposition and expectation), for process (assessing
7332 the quality of cultivation processes and key points), for cultivation bodies (assessing student quality and teacher
7333 quality, and for resource (assessing the investment for teaching and learning resources).
7334



7335 Agile education is conducive to open ecosystems
7336 for learning. Educational ecosystems encourage
7337 active promotion and constraint roles by
7338 engendering and developing an evolutionary education system. An open education ecosystem is a student-centric
7339 education system and environment that coordinates or integrates educational resources inside and outside a university.
7340 International resources are also possible through multi-channel collaborations for agile education and sustainable
7341 competencies development. A student-centric educational ecosystem can lead to interdisciplinary and comprehensive
7342 education, university-industry co-education, international joint education, creative and entrepreneurship education,
7343 and campus culture-based education. These in turn lead to agile education. Figure H.3 illustrates these findings.
7344
7345

Figure H.3. Open Ecosystems for Agile Education
(Courtesy of Prof. Xu Xiaofei)

7346 I.5: Service-oriented Computing Education

7347
7348 Service-oriented education is a natural outgrowth of agile education. To cultivate, ensure and enhance the sustainable
7349 competencies of students continuously in their entire professional life would become the important missions and
7350 educational service functions of universities in the future. Educational transformational trends suggest that:
7351

7352 Qualified Graduates ==> Student Lifetime Sustainable Competencies

7353
7354 Service-oriented education is a new form of student lifetime sustainable development education. It performs a multi-
7355 phased, interdisciplinary, ongoing, and adaptive education
7356 that provides continuous multiphases of agile education
7357 services for sustainable competencies during students'
7358 entire professional life. Fundamentally, service-oriented
7359 education leads to:

- 7360 ▪ Student lifetime centric sustainable development
- 7361 continuous education
- 7362 ▪ Individualized development purpose cultivation
- 7363 and learning
- 7364 ▪ Open trans-boundary and interdisciplinary co-
- 7365 education services
- 7366 ▪ Iterated multi-phased agile education and learning
- 7367 ▪ Professional online and offline education centers
- 7368 ▪ Smart education service networked platforms



7369 Figure H.4. Promise of service-oriented education
7370 (Courtesy of Prof. Xu Xiaofei)

7371 Ultimately, the process leads to student lifetime sustainable
7372 competencies. Figure H.4 illustrates the promise of service-oriented education.

7373 In conclusion, to cultivate innovative talent with sustainable competencies to adapt the development of future
7374 emerging technologies and economies, it is important to reform and restructure current higher education systems,
7375 models, and ecosystems with new forms of engineering and education for sustainable competencies. As a new and
7376 advanced education form, agile education promises to improve higher computing and engineering education. These
7377 new forms of advanced education models and approaches (e.g., agile education, service-oriented education) will
7378 achieve realization with practice and exploration at universities throughout China and beyond.

7381 I.5: References for Appendix I

7382
7383
7384 [Blu1] “Blue Book” (2019) *The Blue Book Working Group of the Forum of Chinese Twenty-Experts on Computing Education*
7385 *in China* Computer Education for Sustainable Competence, China Higher Education Publishing, January 2019 (in
7386 Chinese).
7387 [Imp1] Impagliazzo, John. (2018) The Role of Competency and the Future of Computer Education. *Communications of the*
7388 *China Computer Federation*; Volume 14, Number 9; September 2018. (in Chinese)
7389

7390
7391
7392
7393
7394
7395
7396
7397
7398
7399
7400

Appendix Z: Contributors and Reviewers

The CC2020 Steering Committee and Task Force would like to acknowledge all contributors and reviewers who have provided valuable input, assistance, and feedback on this project.

DRAFT

7401
7402

7403 **Acknowledgements**

7404
7405

7406 The CC2020 project acknowledges organizations and people who have contributed to this important effort. The project
7407 acknowledges the Association for Computing Machinery (ACM), the IEEE Computer Society (IEEE-CS), the
7408 Association for Information Systems (AIS), Association of Information Technology Professionals (AITP) EDSIG,
7409 and ACM SIGCHI. It also acknowledges project collaborators such as the Information Processing Society of Japan
7410 (IPSJ), the Chinese Computing Federation (CCF), the Latin American Center on Computing [EL Centro
7411 Latinoamericano de Estudios en Informática (CLEI)], ACM India, GRIN (Italian Association of Computer Scientists)
7412 ACM Europe, Informatics Europe, I4All, and the West Texas A&M University.

7413

7414 The CC2020 Project would like to thank the National Strategic Planning and Analysis Research Center (NSPARC)
7415 at Mississippi State University for its support of the Web development for this project, as well as its support of some
7416 of the publication costs associated with this project.

7417

7418 The CC2020 project also acknowledges the many people who participated in this project including the CC2020 task
7419 force and its steering committee. Appendix Z lists other people who have contributed content to the project and those
7420 who have volunteered their time to review the CC2020 report as it progressed.

7421

7422

7423

7424

7425
7426

7427
7428
7429
7430

References

R1: References for Report

7431
7432
7433
7434
7435
7436
7437
7438
7439
7440
7441
7442
7443
7444
7445
7446
7447
7448
7449
7450
7451
7452
7453
7454
7455
7456
7457
7458
7459
7460
7461
7462
7463
7464
7465
7466
7467
7468
7469
7470
7471
7472
7473
7474
7475
7476
7477
7478
7479
7480
7481
7482
7483
7484
7485
7486
7487
7488
7489
7490

- [Aac1] American Association of Community Colleges (AACC). <https://www.aacc.nche.edu/>
- [Acc1] Accenture: Technology Vision 2020: We, the Post-Digital People (<https://www.accenture.com/us-en/insights/technology/technology-trends-2020>)
- [Acm00] ACM Website. <https://www.acm.org/>. Accessed 2019 May 6.
- [Acm01] ACM Curricula Reports Website. <https://www.acm.org/education/curricula-recommendations>. Accessed 2019 March 11.
- [Acm02] ACM (2005). *Computing Curricula 2005 The Overview Report*. ACM and IEEE Computer Society, ISBN: 1-59593-359-X.
- [Acm03] ACM (2010) *IS 2010 Curriculum Guidelines for Undergraduate Degree Programs in Information Systems*, Association for Computing Machinery (ACM) and Association for Information Systems (AIS).
- [Acm04] ACM (2013). *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, Association for Computing Machinery and IEEE Computer Society.
- [Acm05] ACM (2014). *Software Engineering Curricula 2014 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*, Association for Computing Machinery and IEEE Computer Society.
- [Acm06] ACM (2016). *Computer Engineering Curricula 2016 Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering*, Association for Computing Machinery (ACM) and IEEE Computer Society, ISBN: 978-1-4503-4875-1, DOI: 10.1145/3025098, Web link: <https://dx.doi.org/10.1145/3025098>.
- [Acm07] ACM (2017). *Information Technology Curricula 2017 Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology*, Association for Computing Machinery (ACM) IEEE Computer Society (IEEE-CS), ISBN: 978-1-4503-6416-4, DOI: 10.1145/3173161, Web link: <https://dl.acm.org/citation.cfm?id=3173161>.
- [Acm08] ACM (2017). *Cybersecurity Curricula 2017 Curriculum Guidelines for Post-Secondary Degrees in Cybersecurity*. Association for Computing Machinery (ACM), IEEE Computer Society (IEEE-CS), Association for Information Systems Special Interest Group on Information Security and Privacy (AIS SIGSEC), International Federation for Information Processing Technical Committee on Information Security Education (IFIP WG 11.8), ISBN: 978-1-4503-5278-9, DOI: 10.1145/3184594, Web link: <https://dl.acm.org/citation.cfm?id=3184594>. Also at <https://cybered.hosting.acm.org/wp/>
- [Acm09] ACM CCECC (2014). *Information Technology Competency Model of Core Learning Outcomes and Assessment for Associate-Degree Curriculum*. Available at <http://ccecc.acm.org/guidance/information-technology>.
- [Acm10] ACM (2017). *Computer Science Transfer Curriculum 2017 Computer Science Curricular Guidance for Associate-Degree Transfer Programs with Infused Cybersecurity*, The Association for Computing Machinery (ACM) Committee for Computing Education in Community Colleges (CCECC), ISBN: 978-1-4503-5369-4, DOI: 10.1145/3108241, Web link: <http://dx.doi.org/10.1145/3108241>.
- [Acm11] ACM (2017) Topi, H., Karsten, H., Brown, S. A., Carvalho, J. A., Donnellan, B., Shen, J., et al. (2017). MSIS 2016: Global Competency Model for Graduate Degree Programs in Information Systems, 40(1). Retrieved from <http://aisel.aisnet.org/cais/vol40/iss1/186>
- [Acm12] ACM (2001). *Computing Curricula 2001: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, Association for Computing Machinery and IEEE Computer Society. <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2001.pdf>. Accessed 2019 Jun 26.
- [Acm13] ACM. 1968. Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science. *Communications of the ACM (CACM) Volume 11 Issue 3, March 1968*, pp 151-197.
- [Acm14] ACM. 1972. Ashenhurst, Robert L (Ed.). *Curriculum recommendations for graduate professional programs in information systems. Communications of the ACM (CACM), Volume 15 Issue 5, May 1972, Pages 363-398.*
- [Acm15] ACM. 1973. Couger, J. (Ed.). *Curriculum Recommendations for Undergraduate Programs in Information Systems, Communications of the ACM, Volume 16, Number 12, December 1973, pp. 727-749.*
- [Acm16] ACM Education Policy Committee. 2018. *Lighting the Path: From Community College to Computing Careers*. <https://www.acm.org/binaries/content/assets/education/lighting-the-path-from-community-college-to-computing-careers.pdf>. Accessed 2019Aug17.
- [Als1] Alsop, Steve. 2005. *Beyond Cartesian Dualism: Encountering Affect in the Teaching and Learning of Science*. Vol. 29. Springer Science & Business Media, Berlin, Germany.
- [And1] Anderson (2001). Bloom’s Cognitive Process Dimension.
- [And5] L. W. Anderson et al., “A taxonomy for learning, teaching, and assessing: A revision of Bloom’s taxonomy of educational objectives, abridged edition,” White Plains, NY Longman, 2001.
- [Ann1] Annas, Julia, “Intelligent Virtue,” Oxford Press, 2011.
- [Atc1] Atchison, William F. et al., (1968) Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science, *Communications of the ACM: Volume 11 Issue 3, March 1968*.
- [Are1] Are-We-In-Your-State, <https://www.areweinyourstate.org/>
- [Aus1] Austing, Richard H. et al. Curriculum '78: Recommendations for the undergraduate program in computer science— a report of the ACM curriculum committee on computer science, *Communications of the ACM: Volume 22 Issue 3, March 1979*.
- [Bac1] Bachelor degrees in Japan (2017). http://www.niad.ac.jp/media/001/201901/no9_13_2017data_fuki_Bachelor_English.pdf. Accessed 2019 June 27.
- [Bai1] Baiyere, A., Donnellan, B., Venkatesh, V., and Wyatt, J. (forthcoming). “Internet of Things (IoT) – A Research Agenda for Information Systems,” Conditionally accepted for publication in *Communications of the AIS*. (Forthcoming)

- 7491 [Bai2] D. Bairaktarova and A. Woodcock, "Engineering Student's Ethical Awareness and Behavior: A New Motivational Model," *Sci. Eng. Ethics*, vol. 23, no. 4, pp. 1129–1157, 2017.
- 7492 [Ban1] Wilder, Coleen R. and Ozgur, Ceyhun O. (2015) *Business Analytics Curriculum for Undergraduate Majors*, published in *INFORMS Transactions on Education* 15(2):180-187. <https://doi.org/10.1287/ited.2014.0134>, Accessed 2010 Jan06.
- 7493 [Bar1] Baron, R. M., & Kenny, D. A. (1986). The moderator-mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology*, 51, 1173-1182. Retrieved from <https://psych.wisc.edu/henriques/mediator.html> (current September 12, 2019).
- 7494 [Bau1] I. Baumgartner and V. Shankaraman, "Actively linking learning outcomes and competencies to course design and delivery: experiences from an undergraduate Information Systems program in Singapore," in *Proceedings of the 2013 IEEE Global Engineering Education Conference*, Berlin, Germany, 2013, pp. 238–246.
- 7495 [Bill] S. Billett, "Realising the educational worth of integrating work experiences in higher education," *Stud. High. Educ.*, vol. 34, no. 7, pp. 827–843, Nov. 2009.
- 7496 [Biz1] Business Dictionary. <http://www.businessdictionary.com/definition/entrepreneurship.html>
- 7497 [Biz2] Business Dictionary. <http://www.businessdictionary.com/definition/innovation.html>
- 7498 [Bhe1] BHEF (2016) *Data Science and Analytics (DSA) Competency Map, The Business Higher Education Framework (BHEF)*, version 1.0, produced in November 2016. https://s3.goeshow.com/dream/DataSummit/Data%20Summit%202018/BHEF_2016_DSA_competency_map_1.pdf. Accessed 2020Jan06.
- 7499 [Blo1] Bloom, B.S. (Ed.). Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R. (1956). *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. New York: David McKay Co Inc.
- 7500 [Bls1] Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 2016-17 Edition, Computer and Information Technology Occupations; <https://www.bls.gov/ooh/computer-and-information-technology/home.htm>. Accessed 2017 Dec 2.
- 7501 [Bls2] Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 2016-17 Edition, Computer and Information Technology Occupations, on the Internet at <https://www.bls.gov/ooh/computer-and-information-technology/information-security-analysts.htm>. Accessed 2017 Dec 2.
- 7502 [Blu1] "Blue Book" (2019) *The Blue Book Working Group of the Forum of Chinese Twenty-Experts on Computing Education in China* Computer Education for Sustainable Competence, China Higher Education Publishing, January 2019 (in Chinese).
- 7503 [Bol1] Bologna Declaration. <http://www.ehea.info/cid100210/ministerial-conference-bologna-1999.html>
- 7504 [Bol2] Bologna Working Group on Qualifications Frameworks (2005). *A Framework for Qualifications of the European Higher Education Area*. The Danish Ministry of Science, Technology and Innovation.
- 7505 [Bur1] Burning Glass. (2017). *The Digital Edge: Middle-Skill Workers and Careers*. https://www.burning-glass.com/wp-content/uploads/Digital_Edge_report_2017_final.pdf. Accessed 14 May 2019.
- 7506 [Cas1] Cassel, L., & Topi, H. (2016). *Strengthening Data Science Education Through Collaboration*. Retrieved from http://www.computingportal.org/sites/default/files/Data%20Science%20Education%20Workshop%20Report%201.0_0.pdf
- 7507 [Ccw1] CC2020 Project Website. <https://www.cc2020.net/>. Accessed 2019 May 9
- 7508 [Cha1] D. W. Chambers and J. D. Gerrow, "Manual for Developing and Formatting Competency Statements," *Journal of Dental Education*, vol. 58, no. 5, pp. 361–66, 1994.
- 7509 [Chy5] Chyung, S., Stepich, D. & Cox, D. (2006) "Building a Competency-Based Curriculum Architecture to Educate 21st-Century Business Practitioners" in *The Journal of Education for Business* 81(6):307-314.
- 7510 [Cio] CIO Council. (2012). Clinger-Cohen core competencies & learning objectives. Washington, DC: CIO Council.
- 7511 [Cla1] Classroom, The. (2019) Knowledge Based Learning. <https://www.theclassroom.com/knowledge-based-learning-5403738.html>
- 7512 [Cle1] Clear, T. (2017). Thinking Issues: Meeting employers' expectations of devops roles: can dispositions be taught? *Inroads*, 8(2), 19–21. <https://doi.org/10.1145/3078298>
- 7513 [Cle2] Clear, A., Clear, T., Impagliazzo J., Wang, P. (2020). "From Knowledge-based to Competency-based Computing Education: Future Directions, submitted for Review.
- 7514 [Col1] Collins Dictionary; <http://www.collinsdictionary.com/dictionary/english/communication-skills>. Accessed 2017 Dec 2.
- 7515 [Con1] Conceiving-Designing-Implementing-Operating (CDIO). Webpage. <http://www.cdio.org/>. Accessed 9 May 2019.
- 7516 [Cos1] Computer Society of IEEE. <http://www.computer.org/>. Accessed 2019 May 6.
- 7517 [Coul] Couger, J. (Ed.). *Curriculum recommendations for Undergraduate programs in information systems*. *Comm. ACM* 16, 12 (Dec. 1973), 727-749.
- 7518 [Cpt1] CompTIA, *Information Technology (IT) Industry & Association*; <https://www.comptia.org/>. Accessed 2017 Dec 2.
- 7519 [Cpt2] CompTIA, *Building Digital Organizations*, February 2017.
- 7520 [Csp1] CSpashshala. www.cspathshala.org/. Accessed 2019Aug17.
- 7521 [Cua1] Cuadros-Vargas, E. (2018a). Escuela Profesional de Ciencia de la Computacion. <https://education.spc.org.pe/Peru/CS-UTEC/Plan%202018/CS-UTEC-poster.pdf> Accessed 9 May 2019.
- 7522 [Cua2] Cuadros-Vargas, E. (2018b). 3.9 Compatibilidad de la carrera con relación a estandares internacionales. https://education.spc.org.pe/Peru/CS-UTEC/Plan%202018/3_9_Compatibilidad_carrera_.html Accessed 9 May 2019.
- 7523 [Dab1] Dabbagh, N., Benson, A. D., Denham, A., Joseph, R., Al-Freih, M., Zgheib, G., ... & Guo, Z. (2016). Massive open online courses. In *Learning technologies and globalization* (pp. 9-13). Springer, Cham.
- 7524 [Dat1] Data Science Draft Report 1. (2019). <http://www.cs.williams.edu/~andrea/DSReportInitialFull.pdf>. Accessed 2019 May 6.
- 7525 [Dat2] Data Science Draft Report 2. (2019) <http://dstf.acm.org/DSReportInitialFull.pdf>. Accessed 2020 Jan 6.
- 7526 [Dat2] Data Science Task Force; <http://dstf.acm.org/>
- 7527 [Dat3] DataUSA. 2019. <https://datausa.io/profile/cip/computer-and-information-sciences-and-support-services>. Accessed 2019Aug17.
- 7528 [Dav1] Dave, R.H. (1970). *Psychomotor levels in Developing and Writing Behavioral Objectives*, pp.20-21. R.J. Armstrong, ed. Tucson, Arizona: Educational Innovators Press.
- 7529 [Dea1] Deadline.com, <https://deadline.com/2018/07/film-industry-revenue-2017-ibisworld-report-gloomy-box-office-1202425692/>
- 7530 [Del1] Deloitte: Deloitte Insights: Tech Trends 2020 (<https://www2.deloitte.com/us/en/insights/focus/tech-trends.html>)
- 7531 [Den1] Denning, P.J. et al., "Computing as a Discipline," *Communications of the ACM*, vol. 32, no. 1, pp. 9-23, (1989)
- 7532 [Dic1] Dictionary.com; <http://dictionary.reference.com/browse/soft+skills>. Accessed 2017 Dec 2.
- 7533 [Dic2] Dictionary.com; <http://dictionary.reference.com/browse/teamwork>. Accessed 2015 Dec 2.

- 7561 [Dso1] Dsouza, John. <https://www.linkedin.com/pulse/based-learning-11-kbl-knowledge-based-john-dsouza>
- 7562 [Edg1] Education Glossary, <https://www.edglossary.org/content-knowledge/>
- 7563 [Edi1] EDSF (2017). The EDISON Data Science Framework. Available at http://edison-project.eu/sites/edison-project.eu/files/attached_files/node-488/edison-general-introduction-edsf.pdf; accessed February 3, 2019.
- 7564 [Edu1] Eduglopedia. <https://eduglopedia.org>
- 7565 [Ent1] Enterprise Information Technology Body of Knowledge (EITBOK). http://eitbokwiki.org/Main_Page. Accessed 2019Aug17.
- 7566 [Eur1] European e-Competence Framework (e-CF), 3.0. <http://www.ecompetences.eu/>
- 7567 [Eur2] European Higher Education Area (EHEA). <http://www.ehea.info/page-three-cycle-system>. Accessed 2019 Aug 16.
- 7568 [Eur3] European Qualifications Framework (EQF). European Commission. Learning Opportunities and Qualifications. <https://ec.europa.eu/ploteus/en/content/descriptors-page/>. Accessed 2019Aug17.
- 7570 [Exe1] Exeter University. <http://www.exeter.ac.uk/undergraduate/degrees/computerscience/comsci/#Programme-structure>.
- 7571 [Fei1] Fein, L. (1959). The Role of the University in Computers, Data Processing, and Related Fields. *Commun. ACM*, 2(9), 7–14. <http://doi.org/10.1145/368424.368427>
- 7573 [Fre1] Frezza, Stephen, Mats Daniels, Arnold Pears, Åsa Cajander, Viggo Kann, Amanpreet Kapoor, Roger McDermott, Anne-Kathrin Peters, Mihaela Sabin, and Charles Wallace. 2019. “Modelling Competencies for Computing Education beyond 2020: A Research Based Approach to Defining Competencies in the Computing Disciplines.” In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 27. ACM.
- 7574 [Fre2] Frezza, Stephen, Mats Daniels, and Aaron Wilkins, “Assessing Students’ IT Professional Values in a Global Project Setting” In *ACM Transactions on Computing Education, Special Issue on Global Software Engineering Education (19,2) 2019* pp. 9:1-9:34, <http://doi.acm.org/10.1145/3231710>.
- 7575 [Fre5] S. Frezza et al., “Modelling competencies for computing education beyond 2020: a research based approach to defining competencies in the computing disciplines,” pp. 148–174, 2018.
- 7576 [Ful2] U. Fuller, A. Pears, J. Amillo, C. Avram, and L. Manilla, “A Computing Academic Perspective on the Bologna Process,” *ACM SIGCSE Bull.*, vol. 38, no. 4, pp. 115–131, Dec. 2006.
- 7577 [Gar1] Gartner: Top 10 Strategic Technology Trends for 2020 (<https://www.gartner.com/en/publications/top-tech-trends-2020>)
- 7578 [Gra1] Gray, J. M. (2015). *Virtue Ethics: Examining Influences on the Ethical Commitment of Information System Workers in Trusted Positions*. Dissertation: Nova Southeastern University. Nova Southeastern University. Retrieved from http://nsuworks.nova.edu/gscis_etd/364
- 7579 [Gra2] G. Grant, ‘On competence: a critical analysis of competence-based reforms in higher education.’, 1979.
- 7580 [Han1] H. Han, “Virtue Ethics, Positive Psychology, and a New Model of Science and Engineering Ethics Education,” *Sci. Eng. Ethics*, vol. 21, no. 2, pp. 441–460, 2014.
- 7581 [Har1] Harrow, A. (1972). *A Taxonomy of Psychomotor Domain: A Guide for Developing Behavioral Objectives*. David McKay Co., Inc., New York, USA.
- 7582 [Har2] Harvard. https://apps2.campusservices.harvard.edu/cas/empldocs/cs/harvard_competency_dictionary_complete.pdf.
- 7583 [Hel1] Helfert, M. (2008). Business informatics: An engineering perspective on information systems. *Journal of Information Technology Education: Research*, 7, 223–245.
- 7584 [Icd1] <http://icdtranslation.com/skill-based-and-knowledge-based-online-training/>
- 7585 [Iee1] IEEE (2017) Los Alamitos, Calif., Dec. 14, 2017. <https://www.prnewswire.com/news-releases/top-10-technology-trends-for-2018-ieee-computer-society-predicts-the-future-of-tech-300571274.html>
- 7586 [Iee2] IEEE Computer Society (2014) Software Engineering Competency Model: Version 1.0, SWECOM 2014.
- 7587 [Imp1] Impagliazzo, John. (2018) The Role of Competency and the Future of Computer Education. *Communications of the China Computer Federation*; Volume 14, Number 9; September 2018. (in Chinese)
- 7588 [Imp2] Impagliazzo, J., et al. (2018). Developing an Overview of Computing/Engineering Curricula via the CC2020 Project. In *Proc. of the IEEE EduNine Conference*. IEEE Education Society.
- 7589 [Imp3] Impagliazzo J., Parrish, A., & Clear, A. (2018). Innovative Computing Curricula and the CC2020 Project. In *Proc. of the Frontiers in Education (FIE) Conference*.
- 7590 [Ind1] India 1. www.ugc.ac.in
- 7591 [Ind2] India 2. www.naac.gov.in
- 7592 [Ind3] India 3. www.aicte-india.org
- 7593 [Ind4] India 4. www.nbaind.org
- 7594 [Inv1] Investopedia; <http://www.investopedia.com/terms/s/soft-skills.asp>. Accessed 2017 Dec 2.
- 7595 [Ipa1] Information-technology Promotion Agency – Japan (IPA). IT Human Resources Development: i Competency Dictionary. <https://www.ipa.go.jp/english/humandev/icd.html>. Accessed 2019Aug18.
- 7596 [Kak1] Kakeshita, Tetsuro. “National Survey of Japanese Universities on Computing Education: Analysis of Departments Majored in Computing Discipline”, *Olympiads in Informatics*, 2018, Vol. 12, 69–84, https://ioinformatics.org/journal/v12_2018_69_84.pdf (accessed June 29, 2019).
- 7597 [Ken1] D. Kennedy, A. Hyland, and N. Ryan, “Learning outcomes and competences,” *Introd. Bologna Object. Tools*, pp. 2–3, 2009.
- 7598 [Kpm1] KPMG: Technology Trends Index USA (<http://techtrendstrendsindex.kpmg.com/>)
- 7599 [Kra1] Krathwohl, D. R., Bloom, B. S., & Bertram, B. M.: (1973) *Taxonomy of Educational Objectives, the Classification of Educational Goals. Handbook II: Affective Domain*. David McKay Co., Inc., New York, USA.
- 7600 [Kra2] M. Kramer, P. Hubwieser, and T. Brinda, “A Competency Structure Model of Object-Oriented Programming,” in *2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*, 2016, pp. 1–8.
- 7601 [Kuh1] George D. Kuh. (2009). The national survey of student engagement: Conceptual and empirical foundations. *New directions for institutional research* 2009, 141 (2009), 5–20.
- 7602 [Len1] Lenburg, C. B. (1999). The framework, concepts and methods of the competency outcomes and performance assessment (COPA) model. *Online Journal of Issues in Nursing*, vol. 4, no. 2, pp 1-12.
- 7603 [Lin1] Lindenwood University Makerspace Lab, <https://www.lindenwood.edu/about/news/details/converged-media-lab-makerspace-to-open-this-fall/>. Accessed 2019Aug17.
- 7604 [Loc1] Lockoff, J., Wegewijs, B., Durkin, K., Wagenaar, R., Gonzales, J., Isaacs, A. K., Donà dalle Rose, L.F., & Gobbi, M. (Eds.) (2010). *A Tuning guide to formulating degree programme profiles: Including programme competences and programme learning outcomes*.

- 7631 Bilbao, Spain: University of Deusto. Retrieved from <http://core-project.eu/documents/Tuning%20G%20Formulating%20Degree%20PR4.pdf>
- 7632
- 7633 [Lun1] Lunt, B.M., Ekstrom, J.J., Gorka, S., Hislop, G., Kamali, R., Lawson, E., LeBlanc, R., Miller, J. and Reichgelt, H. 2008. Curriculum Guidelines for Undergraduate Degree Programs in Information Technology; <https://dl.acm.org/citation.cfm?id=2593311>. Accessed 2017 Dec 2
- 7634
- 7635
- 7636 [Mar1] Marshall, Linda (2012) A comparison of the core aspects of the ACM/IEEE Computer Science Curriculum 2013 Strawman report with the specified core of CC2001 and CS2008 Review. Computer Science Education Research Conference. pp. 29–34. CSERC 2012, ACM.
- 7637
- 7638
- 7639 [Mar2] Marshall, Linda (2014) A graph-based framework for comparing curricula. Ph.D. thesis, University of Pretoria, South Africa.
- 7640 [Mar3] Marshall, Linda (2017) A Topic-Level Comparison of the ACM/IEEE CS Curriculum Volumes. In: Liebenberg J., Gruner S. (eds) ICT Education. SACLA 2017. Communications in Computer and Information Science, vol 730. pp. 309-324. Springer, Cham.
- 7641 [Mar4] M. Lynne Markus and Heikki Topi. 2015. "Big Data, Big Decisions for Science, Society, and Business: Report on a Research Agenda Setting Workshop." Technical Report. National Science Foundation, USA.
- 7642
- 7643 [Mer1] Merriam-Webster. <https://www.merriam-webster.com/dictionary/competency>. Accessed 2019 June 3.
- 7644 [Mer2] Merriam-Webster. Disposition. <https://www.merriam-webster.com/dictionary/disposition/>. Accessed 2019Aug30.
- 7645 [Mer3] Merriam-Webster. Learning. <https://www.merriam-webster.com/dictionary/learning>
- 7646 [Mer4] Merriam-Webster. Knowledge. <https://www.merriam-webster.com/dictionary/knowledge?src=search-dict-box#synonyms>
- 7647 [Min1] Ministry of Education, New Zealand (2019) What are makerspaces? <http://elearning.tki.org.nz/Teaching/Future-focused-learning/Makerspaces>. Accessed 2019 Jul 3.
- 7648
- 7649
- 7650 [Nas1] National Academies of Science (NAS) (2018). Envisioning the Data Science Discipline: The Undergraduate Perspective: Interim Report, ISBN 978-0-309-46502-1 | DOI 10.17226/24886.
- 7651 [Nas2] National Academies of Sciences, Engineering, and Medicine (2018), National Academies of Sciences, Engineering, and Medicine, "Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments," Committee on the Growth of Computer Science Undergraduate Enrollments, The National Academies of Sciences, Engineering and Medicine, Washington, DC, 2018. pp. 84-87
- 7652
- 7653
- 7654
- 7655
- 7656 [New1] University of Newcastle, Information Technology. <https://www.newcastle.edu.au/degrees/bachelor-of-information-technology>.
- 7657 [Nov1] Novinson, M. 2017. *Top 15 Moneymaking Certifications for 2017*. CRN, The Channel Company. <http://www.crn.com/slideshows/managed-services/300080027/top-15-moneymaking-certifications-for-2016.htm>. Accessed 2017 Dec 2.
- 7658
- 7659 [Nrc1] National Research Council (2012) A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas. The National Academies Press. 500 Fifth Street, NW, Washington, DC 20001. ISBN 13: 978-0-309-21742-2.
- 7660
- 7661 [Nwo1] Nwokeji, J. C., Stachel, R., & Holmes, T. (2019). Competencies Required for Developing Computer and Information Systems Curriculum. In Proceedings of the 49th Frontiers in Education Conference (FIE'19) (pp. 1–9). Cincinnati, OH: IEEE Computer Society.
- 7662
- 7663
- 7664 [Ope1] OpenAir: African Innovative Research, <http://www.openair.org.za/africas-maker-movement-an-overview-of-ongoing-research/>. Accessed 2019Aug17.
- 7665
- 7666 [Ols1] Olson, M. 2016. "A Multilateral Approach to Bridging the Global Skills Gap." *Cornell HR Review*, 5/8/15, p. 1.
- 7667 [Par1] DeVeaux, R. et al. (2017) *Curricular Guidelines for Undergraduate Programs in Data Science*, Annual Review of Statistics and Its Application 4:15–30. <https://www.amstat.org/asa/files/pdfs/EDU-DataScienceGuidelines.pdf>. Accessed 2020Jan06.
- 7668
- 7669 [Per1] Perkins, David N., Eileen Jay, and Shari Tishman. 1993. "Beyond Abilities: A Dispositional Theory of Thinking." *Merrill-Palmer Quarterly* 39 (1): 1–21.
- 7670
- 7671 [Per2] Perry, Mark J. 2018. Table of the Day: Bachelor's degrees for the Class of 2016 by field and gender. Oh, and the overall 25.6% degree gap for men! <http://www.aei.org/publication/table-of-the-day-bachelors-degrees-for-the-class-of-2016-by-field-and-gender-oh-and-the-overall-25-6-college-degree-gap-for-men/>. Accessed 2019Aug17.
- 7672
- 7673 [Per3] Peris-Ortiz, M., Llera, J. J. A., & Rueda-Armengot, C. (2017). Entrepreneurship and Innovation in a Revolutionary Educational Model: École, 42. In *Social Entrepreneurship in Non-Profit and Profit Sectors* (pp. 85-97). Springer, Cham.
- 7674
- 7675 [Per4] Perkins, David N. and Shari Tishman. 2006. Learning that matters: Towards a dispositional perspective on education and its research needs. A report prepared for the Spencer Foundation (2006), 43.
- 7676
- 7677 [Per5] D. N. Perkins, E. Jay, and S. Tishman, "Beyond Abilities: A Dispositional Theory of Thinking," *Merrill. Palmer. Q.*, vol. 39, no. 1, pp. 1–21, 1993.
- 7678
- 7679
- 7680 [Pik1] Pikkarainen, Eetu (2014). "Competence as a Key Concept of Educational Theory: A Semiotic Point of View." *Journal of Philosophy of Education* 48 (4): 621–36.
- 7681
- 7682 [Pro1] Provost, Foster and Tom Fawcett (2013) *Data Science for Business*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
- 7683
- 7684 [Psi1] PSI (2018). 5 Stats About the Skills Gap That Demand Attention. <https://blog.psonline.com/talent/5-stats-about-the-skills-gap-that-demands-attention>. Accessed 27 May 2019.
- 7685
- 7686 [Rad1] A. Radermacher and G. Walia, "Gaps Between Industry Expectations and the Abilities of Graduates," in *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 2013, pp. 525–530.
- 7687
- 7688 [Ram1] Ramos, Teófilo J., Osvaldo M Micheloud, Richard Painter, and Moshe Kam (2013). IEEE Common Nomenclature for Computing Related Programs in Latin America. IEEE.
- 7689
- 7690 [Reg1] Reges, Stuart. 2018. Furor on Claim Women's Choices Create Gender Gap in Comp Sci. *Inside Higher Ed*. <https://www.insidehighered.com/news/2018/06/25/lecturers-explanation-gender-gap-computer-science-it-reflect-womens-choices>. Accessed 2019Aug17.
- 7691
- 7692
- 7693 [Rhu1] Rhumb. Webpage. <https://rhumb.com/>. Accessed 9 May 2019.
- 7694 [Sim1] Simon et al., "What's in a Name? International Interpretations of Computing Education Terminology," in *Proceedings of the 2015 ITiCSE on Working Group Reports*, Vilnius, Lithuania, 2015, pp. 173-186.
- 7695
- 7696 [Sab1] Sabin, M., H. Alrumaih, and J. Impagliazzo. 2018. "A Competency-Based Approach toward Curricular Guidelines for Information Technology Education." In 2018 IEEE Global Engineering Education Conference (EDUCON), 1214–21. Tenerife: IEEE.
- 7697
- 7698 [Sch1] Schussler, Deborah L. 2006. "Defining Dispositions: Wading Through Murky Waters." *The Teacher Educator* 41 (4): 251–68.
- 7699 [Sfi1] Skills Framework for Information Age (SFIA)

- 7700 [Sim1] Simon, Clear, A., Carter, J., Cross, G., Radenski, A., Tudor, L., & Tonisson, E. (2015). What's in a Name? International Interpretations of Computing Education Terminology. *ITICSE2015* (pp. 173-186). Vilnius, Lithuania: ITICSE-WGR'15.
- 7701 [Sim2] Simon, H. (1996). *The Sciences of the Artificial*, 3ed, M.I.T., Cambridge, MA, p. 119.
- 7702 [Sii1] Siirtola, H., Rähkä, K., & Surakka, V. (2013). Interactive Curriculum Visualization. In *2013 17th International Conference on Information Visualisation*, pp. 108-117.
- 7703 [Som1] Sommaruga, L., & Catenazzi, N. (2007). Curriculum visualization in 3D. In *Proc. of the 12th international conference on 3D web technology (Web3D '07)*. ACM. pp.177-180.
- 7704 [Squ1] A. Squires and W. Larson, "Improving systems engineering curriculum using a competency-based assessment approach," *International Journal of Intelligent Defence Support Systems*, vol. 2, no. 3, pp. 184, 2009.
- 7705 [Sta1] Stamper, R. K. (1991) The semiotic framework for information systems research. Information systems research. *Contemporary approaches and emergent traditions*, pp. 515-528.
- 7706 [Sta2] Stamper, R. K., *Information in Business and Administrative Systems*, John Wiley and Sons, New York, NY, 1973.
- 7707 [Sta3] Stamper, R. K., Althous, K. and Backhouse, J., "MEASUR, Method for Eliciting, Analyzing, and Specifying User Requirements," In Olle, T. W., Verrijn-Stuart, A. A. and Bhabuts, L., (eds.), *Computerized Assistance During the Information Life Cycle*, Elsevier, Amsterdam, The Netherlands, 1988.
- 7708 [Ste1] Steinbuch, Karl (1957). *Informatik: Automatische Informationsverarbeitung*. Berlin: SEG-Nachrichten.
- 7709 [Sto1] A. Stoof, R. L. Martens, and J. J. Van Merriënboer, "What is competence? A constructivist approach as a way out of confusion," *Onderwijs Res. Dagen ORD Leiden Neth.*, 2000.
- 7710 [Tak1] Takada, Shingo, et al. (2019) <<Visualization paper>> Springer Nature.
- 7711 [Tak5] S. Takada et al., "Toward Understanding Computing Curricula: Visualization Using Competency within the CC2020 Project," *Educ. Inf. Technol.*, 2020.
- 7712 [Tan1] Tang, Cara, Elizabeth K Hawthorne, Cindy S Tucker, Ernesto Cuadros-Vargas, Diana Cukierman, Simon, Ming Zhang (2016). Global Perspectives on the Role of Two-Year/Technical/Junior Colleges in Computing Education. 21st ACM Conference on Innovation and Technology in Computer Science Education (ITICSE '16), Arequipa, Peru, 204-205.
- 7713 [Tec1] TechTarget. <https://whatis.techtarget.com/definition/concept-map>
- 7714 [Ted1] Tedre, Matti (2015). *The Science of Computing: Shaping a Discipline*. Boca Raton, CRC Press / Taylor & Francis.
- 7715 [Ted2] Tedre, M., & Sutinen, E. (2008). Three traditions of computing: what educators should know. *Computer Science Education*, 18(3), 153-170.
- 7716 [Tes1] <https://www.tes.com/teaching-resource/kbl-knowledge-based-learning-11361440>
- 7717 [The1] <https://www.theesa.com/wp-content/uploads/2019/05/2019-Essential-Facts-About-the-Computer-and-Video-Game-Industry.pdf>
- 7718 [Tre1] Trevelyan, James. 2014. *The Making of an Expert Engineer*. Taylor and Francis Group, Ltd.
- 7719 [Top1] Topi, Heikki. 2017. "Information Systems in CC2020: Comparing Key Structural Elements of Curriculum Recommendations in Computing." In 2017 Proceedings of SIGED: IAIM Conference. Association for Information Systems Electronic Library (AISeL). <http://aisel.aisnet.org/siged2017/9>.
- 7720 [Top5] H. Topi, "Information Systems in CC2020: Comparing Key Structural Elements of Curriculum Recommendations in Computing," in 2017 Proceedings of SIGED: IAIM Conference, 2017.
- 7721 [Tuc1] Tucker, A.B. 1991. Computing curricula 1991. *Communications of the ACM*, 34(6), pp.68-84; <http://dl.acm.org/citation.cfm?doid=103701.103710>.
- 7722 [Van1] M. Van der Klink and J. Boon, "The investigation of competencies within professional domains," *Hum. Resour. Dev. Int.*, vol. 5, no. 4, pp. 411-424, 2002.
- 7723 [Van2] M. Van der Klink, J. Boon, and K. Schlusmans, "Competences and Vocational Higher Education: Now and in Future," *Eur. J. Vocat. Train.*, vol. 40, no. 1, pp. 67-82, 2007.
- 7724 [Vor1] Voorhees, R. A., & Bedard-Voorhees, A. (2016). Principles for Competency-based education. In *Instructional-Design Theories and Models, Volume IV* (pp. 49-80). Routledge.
- 7725 [Wag1] Waguespack, L., Babb, J. S., & Yates, D. (2018). Triangulating Coding Bootcamps in IS Education: Bootleg Education or Disruptive Innovation? *Information Systems Education Journal*, 16(6), 48.
- 7726 [Wag5] L. Waguespack et al., "Adopting Competency Mindful of Professionalism in Baccalaureate Computing Curricula," in *EDSIGCON 2019*, 2019.
- 7727 [Web1] Weber, H. (2017). The New Virtues of Engineering and the Need for Change in the Engineering Curriculum. From https://www.researchgate.net/publication/325924314_The_New_Virtues_of_Engineering_and_the_Need_for_Change_in_the_Engineering_Curriculum
- 7728 [Wef1] World Economic Forum (2020): Jobs of Tomorrow: Mapping Opportunity in the New Economy (<https://www.weforum.org/reports/jobs-of-tomorrow-mapping-opportunity-in-the-new-economy>)
- 7729 [Wei1] F. E. Weinert, "Definition and selection of competencies: Concepts of Competence," *Organization for Economic Co-operation and Development*, 1999.
- 7730 [Wig1] Wiggins, G., McTighe, J., and Ebrary, I. (2005). *Understanding by design (Expanded second edition)*. Alexandria, VA: Association for Supervision and Curriculum Development.
- 7731 [Wig2] Wiggins, G., & McTighe, J. (2011). *The Understanding by Design Guide to Creating High-Quality Units*. Alexandria, VA: Association for Supervision and Curriculum Development.
- 7732 [Will1] Willcox, K., & Huang, L. (2017). Mapping the CDIO Curriculum with Network Models. CDIO, In *13th International CDIO Conference*.
- 7733 [Zuc1] Zucker, R. (2009). ViCurriAS: a curriculum visualization tool for faculty, advisors, and students. In *J. Comput. Sci. Coll.* 25(2);138-145.

7769 R2: Extra References

- 7770
- 7771 [Acc1] Accenture: Technology Vision 2020: We, the Post-Digital People (<https://www.accenture.com/us-en/insights/technology/technology-trends-2020>)
- 7772
- 7773 [Acm20] ACM, & AIS. (2017c). *MSIS 2016: Global Competency Model for Graduate Degree Programs in Information Systems*. <https://www.acm.org/binaries/content/assets/education/msis2016.pdf>. Accessed 14 May 2019.
- 7774
- 7775 [Acm21] ACM (2019). *Computing Competencies for Undergraduate Data Science Curricula* (Initial Draft). <http://www.cs.williams.edu/~andrea/DSReportInitialFull.pdf>. Accessed 9 May 2019.
- 7776
- 7777 [Acm22] ACM Education Policy Committee (2018). *Lighting the Path from Community College to Computing Careers*. Available at <https://www.acm.org/binaries/content/assets/education/lighting-the-path-from-community-college-to-computing-careers.pdf>.
- 7778
- 7779 [And1] Anderson, L. W., & Krathwohl, D.R. (2001). *A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of educational objectives*. New York: Longman.
- 7780
- 7781 [Ash1] Ashenhurst, R. L. (1972). Curriculum Recommendations for Graduate Professional Programs in Information Systems. *Communications of the ACM*, 15(5), 364–398.
- 7782
- 7783 [Atc2] Atchison, W. F. (1981). Computer Education, Past, Present, and Future. *ACM SIGCSE Bulletin*, 13(4), 2–6. <http://doi.org/10.1145/989306.989307>
- 7784
- 7785 [Bar1] Barr, Valerie, and Chris Stephenson. 2011. “Bringing Computational Thinking to K-12: What Is Involved and What Is the Role of the Computer Science Education Community?” *ACM Inroads* 2 (1): 48.
- 7786
- 7787 [Big1] Biggs. Approaches to the enhancement of tertiary teaching. *Higher Education Research and Development* 8, 1 (1989), 7–25.
- 7788
- 7789 [Cas1] Cassel, Lillian N., Robert H. Sloan, Gordon Davies, Heikki Topi, and Andrew McGettrick. 2007. “The Computing Ontology Project: The Computing Education Application.” *SIGCSE Bull.* 39 (1): 519–20.
- 7790
- 7791 [Cou1] Council of Chief State School Officers. 2013. “Knowledge, Skills, and Dispositions: The Innovation Lab Network State Framework for College, Career, and Citizenship Readiness, and Implications for State Policy.” <https://files.eric.ed.gov/fulltext/ED542708.pdf>.
- 7792
- 7793 [Cle1] Alison Clear, Allen Parrish, Gerrit C. van der Veer, Ming Zhang. (2017) *CC2020: A Vision on Computing Curricula*. Panel, Proceedings of the 2017 ACM SIGCSE Technical Symposium, DOI: 10.1145/3017680.3017690
- 7794
- 7795 [Con] Conte, S. D., Hamblen, J. W., Kehl, W. B., Navarro, S. O., Rheinboldt, W. C., Young, D. M., Jr, & Atchinson, W. F. (1965). An Undergraduate Program in Computer Science—Preliminary Recommendations. *Commun. ACM*, 8(9), 543–552. <http://doi.org/10.1145/365559.366069>
- 7796
- 7797 [Cog1] Couger, J. D. (1973). Curriculum recommendations for undergraduate programs in information systems. *Communications of the ACM*, 16(12), 727–749. <http://doi.org/10.1145/362552.362554>
- 7798
- 7799 [Cou2] Couger, J. D., Davis, G. B., Dologite, D. G., Feinstein, D. L., Gorgone, J. T., Jenkins, A. M., et al. (1995). IS'95: Guideline for undergraduate IS curriculum. *Mis Quarterly*, 341–359.
- 7800
- 7801 [Dav1] Davis, G. B., Gorgone, J. T., Couger, J., Feinstein, D. L., & Longenecker, H. E., Jr. (1996). IS'97: model curriculum and guidelines for undergraduate degree programs in information systems. *ACM SIGMIS Database*, 28(1), 101–194.
- 7802
- 7803 [Dav2] Davis, Martin (2011) *The Universal Computer: The Road from Leibniz to Turing*. New York: A K Peters/CRC Press, 2011, ISBN 9781466505209.
- 7804
- 7805 [Eds1] Edström, Kristina. 2017. “Exploring the Dual Nature of Engineering Education: Opportunities and Challenges in Integrating the Academic and Professional Aspects in the Curriculum.” Edited by Anette Kolmos. PhD in Technology and Learning, Stockholm, Sweden: KTH Royal Institute of Technology. <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-217315>.
- 7806
- 7807
- 7808 [Fre3] Frezza, Stephen T., Alison Clear, and Abhijat M. Vichare. 2018. “Voices on the Core of Computing.” In 48th Frontiers in Education Conference (FIE'18), 9. IEEE-CS.
- 7809
- 7810 [Full] Fuller, Ursula, and Bob Keim. 2007. “Should We Assess Our Students’ Attitudes?” In Proceedings of the Seventh Baltic Sea Conference on Computing Education Research, 187–90. Koli National Park, Finland: ACM.
- 7811
- 7812 [Fin1] Fincher, S., & Petre, M. (2014). *Computer Science Education Research*. CRC Press.
- 7813
- 7814 [For1] Forsythe, G. E. (1967). A University's Educational Program in Computer Science. *Commun. ACM*, 10(1), 3–11. <http://doi.org/10.1145/363018.363038>
- 7815
- 7816 [Gor1] Gorgone, J., & Gray, P. (2000). MSIS 2000: model curriculum and guidelines for graduate degree programs in information systems. *Communications of the AIS*, 3(1es), 1.
- 7817
- 7818 [Gor2] Gorgone, J. T., Davis, G. B., Valacich, J. S., Topi, H., Feinstein, D. L., & Longenecker, H. E. (2002). IS 2002 model curriculum and guidelines for undergraduate degree programs in information systems. *Communications of the Association for Information Systems*, 11(1).
- 7819
- 7820 [Gor3] Gorgone, J. T., Gray, P., Stohr, E. A., Valacich, J. S., & Wigand, R. T. (2006). MSIS 2006: Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems. *Communications of the AIS*, 17(17), 1–58.
- 7821
- 7822 [Gru1] Gruner, Stefan (2011) "Problems for a Philosophy of Software Engineering," *Minds & Machines*. p. 25, February 2011.
- 7823
- 7824 [Gib1] Gibbs, G. Teaching students to learn: A student-centered approach. Open University Press (1981).
- 7825
- 7826 [Gos1] Gosselin, David. 2017. “Competencies vs Learning Outcomes.” Strengthen Workforce Prep in Your Program. 2017. https://serc.carleton.edu/integrate/programs/workforceprep/competencies_and_LO.html.
- 7827
- 7828 [Got1] Gottipati, Swapna, and Venky Shankararaman. 1/2018. “Competency Analytics Tool: Analyzing Curriculum Using Course Competencies.” *Education and Information Technologies* 23 (1): 41–60.
- 7829
- 7830 [Hay1] Hayashiguchi, E., O. Endou, and J. Impagliazzo. 2018. “The ‘i Competency Dictionary’ Framework for IT Engineering Education.” In 2018 IEEE World Engineering Education Conference (EDUNINE), 1–6.
- 7831
- 7832 [Hub1] Hubwieser, Peter, and Sue Sentance. 2018. “Taxonomies and Competency Models.” In *Computer Science Education: Perspectives on Teaching and Learning in School*, edited by Sue Sentance, Erik Barendsen, and Carsten Schulte, 1st ed., 221–42. London: Bloomsbury Academic.
- 7833
- 7834 [Jaf1] Jafar, M., Waguespack, L., & Babb, J. (2017). A Visual Analytics Approach to Gain Insights into the Structure of Computing Curricula, In *Proc. of 2017 EDSIG Conference*.
- 7835
- [Liu1] Liu, K. (2000). *Semiotics in Information Systems Engineering*. Cambridge University Press: Cambridge (2000).

- 7836 [Koe1] Koeppen, Karoline, Johannes Hartig, Eckhard Klieme, and Detlev Leutner. 2008. "Current Issues in Competence Modeling and
7837 Assessment." *Swiss Journal of Psychology: Official Publication of the Swiss Psychological Society Schweizerische Zeitschrift Fur
7838 Psychologie = Revue Suisse de Psychologie* 216 (2): 61–73.
- 7839 [Kem1] Kember, D., and Kwan, K.-P. Lecturers' approaches to teaching and their relationship to conceptions of good teaching. *Instructional
7840 Science* 28 (2000), 469–490. 10.1023/A:1026569608656.
- 7841 [Kro1] Kroeze, Jan H. 2010. "Ontology Goes Postmodern in ICT." In *Proceedings of the 2010 Annual Research Conference of the South
7842 African Institute of Computer Scientists and Information Technologists*, 153–59. SAICSIT '10. New York, NY, USA: ACM.
- 7843 [Leb] LeBlanc, R. J., Sobel, A., Diaz-Herrera, J. L., Hilburn, T. B., others. (2006). *Software Engineering 2004: Curriculum Guidelines for
7844 Undergraduate Degree Programs in Software Engineering*. IEEE Computer Society.
- 7845 [Lon] Longenecker, H. E., Jr., Feinstein, D. L., Fournier, R. L., Doran, M. V., & Reaugh, W. R. (1991). IS'90: The DPMA Model
7846 Curriculum for Information Systems for 4 Year Undergraduates. *Park Ridge, Illinois: Data Processing Management Association*.
- 7847 [Lun] Lunt, B. M., Ekstrom, J. J., Gorka, S., Hislop, G., Kamali, R., Lawson, E., et al. (2008). *Information Technology 2008: Curriculum
7848 Guidelines for Undergraduate Degree Programs in Information Technology*. New York, NY: ACM.
- 7849 [Lan1] Lang, James D., Susan Cruse, Francis D. McVey, and John McMasters. 1999. "Industry Expectations of New Engineers: A Survey to
7850 Assist Curriculum Designers." *Journal of Engineering Education* 88 (1): 43–51.
- 7851 [Loz1] Lozano, J. Felix, Alejandra Boni, Jordi Peris, and Andrés Hueso. 2012. "Competencies in Higher Education: A Critical Analysis from
7852 the Capabilities Approach." *Journal of Philosophy of Education* 46 (1): 132–47.
- 7853 [Lyn1] Lynch, Daniel R., Jeffrey S. Russell, Jeffrey C. Evans, and Kevin G. Sutterer. 2009. "Beyond the Cognitive: The Affective Domain,
7854 Values, and the Achievement of the Vision." *Journal of Professional Issues in Engineering Education and Practice* 135 (1): 47–56.
- 7855 [Nun1] Nunamaker, J. F., Couger, J. D., & Davis, G. B. (1982). Information systems curriculum recommendations for the 80s: undergraduate
7856 and graduate programs. *Communications of the ACM*, 25(11), 781–805. <http://doi.org/10.1145/358690.358698>
- 7857 [Pas1] Passow, Honor J. 01/2012. "Which ABET Competencies Do Engineering Graduates Find Most Important in Their Work?" *Journal of
7858 Engineering Education* 101 (1): 95–118.
- 7859 [Sab2] Sabin, Mihaela, Svetlana Peltsverger, Bill Paterson, Ming Zhang, and Hala Alrumaih. 2017. "IT2017 Report: Putting It to Work." In
7860 *Proceedings of the 18th Annual Conference on Information Technology Education*, 95–96. SIGITE '17. New York, NY, USA: ACM.
- 7861 [Sch1] Schussler, Deborah L. 2006. "Defining Dispositions: Wading Through Murky Waters." *The Teacher Educator* 41 (4): 251–68.
- 7862 [Shi1] Shiveley, James, and Thomas Misco. 2010. "But How Do I Know About Their Attitudes and Beliefs?: A Four-Step Process for
7863 Integrating and Assessing Dispositions in Teacher Education." *The Clearing House: A Journal of Educational Strategies, Issues and
7864 Ideas* 83 (1): 9–14.
- 7865 [Sic1] Sicilia, Miguel-Angel. 2010. "How Should Transversal Competence Be Introduced In Computing Education?" *ACM SIGCSE
7866 Bulletin* 41 (4): 95–98.
- 7867 [Smi1] Smith, Brad and Harry Shum (2018) *The Future Computed: Artificial Intelligence and its Role in Society*. Microsoft Corporations. 1
7868 Microsoft Way. Redmond, Washington 98052. ISBN 978-0-9997508-1-0
- 7869 [Spe1] Spector, Alfred Z. (2017, August) *Changing Nature of Computer Science and Its Impact on Undergraduate Education*. [Online].
7870 http://sites.nationalacademies.org/cs/groups/cstbsite/documents/webpage/cstb_173998.pdf
- 7871 [Ted3] Tedre, M., Simon, & Malmi, L. (2018). Changing aims of computing education: a historical survey. *Computer Science Education*,
7872 28(2), 158–186.
- 7873 [Tra1] The Royal Academy of Engineering, The Institution of Engineering and Technology, and The British Computer Society. 2009.
7874 "Engineering Values in IT." 3 Carlton House Terrace, London: The Royal Academy of Engineering.
7875 <https://www.raeng.org.uk/publications/reports/engineering-values-in-it>.
- 7876 [Top1] Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker, J. F., Sipior, J. C., & de Vreede, G. J. (2010). IS 2010: Curriculum
7877 Guidelines for Undergraduate Degree Programs in Information Systems. *Communications of the Association for Information Systems*,
7878 26(18). Retrieved from <http://aisel.aisnet.org/cais/vol26/iss1/18/>
- 7879 [Top2] Topi, Heikki, Helena Karsten, Sue A. Brown, João Alvaro Carvalho, Brian Donnellan, Jun Shen, Bernard C. Y. Tan, and Mark F.
7880 Thouin. 2017. "MSIS 2016: Global Competency Model for Graduate Degree Programs in Information Systems." *The Joint ACM/AIS
7881 MSIS 2016 Task Force*. 40(1). <https://doi.org/10.1145/3127597>. Retrieved from <http://aisel.aisnet.org/cais/vol40/iss1/18>
- 7882 [Tri1] Trigwell, K., Prosser, M., and Waterhouse, F. Relations between teachers' approaches to teaching and students' approaches to
7883 learning. *Higher Education* 37 (1999), 57–70.
- 7884 [Tre1] Trevelyan, J. 2012. "Understandings of Value in Engineering Practice." In *2012 Frontiers in Education Conference Proceedings*, 1–6.
7885 Seattle, WA, USA: IEEE.
- 7886 [Ute1] UTEC-1. <https://education.spc.org.pe/Peru/CS-UTEC/Plan%202018/CS-UTEC-poster.pdf>
- 7887 [Ute2] UTEC-2. https://education.spc.org.pe/Peru/CS-UTEC/Plan%202018/3_9_Compatibilidad_carrera_.html
- 7888 [Win1] Winterton, Jonathan, Françoise Delamare- Le Deist, and Emma Stringfellow. 2006. *Typology of Knowledge, Skills and Competences*.
7889 Clarification of the Concept and Prototype: Cedefop References. CEDEFOP Reference Series. Luxembourg: Office for Official
7890 Publications of the European Communities.
- 7891 [Usd1] USDoE (2018). "Competency-Based Learning or Personalized Learning," U.S. Department of Education. [https://www.ed.gov/oii-
7892 news/competency-based-learning-or-personalized-learning](https://www.ed.gov/oii-news/competency-based-learning-or-personalized-learning). Accessed 14 May 2019.
- 7893 [Wag2] Waguespack, L., Babb, J. (2019). Toward Visualizing Computing Curricula: The Challenge of Competency. In *Information Systems
7894 Education Journal*, 17(4) pp 51-69. <http://isedj.org/2019-17/> ISSN: 1545-679X. (A preliminary version appears in *Proceedings of
7895 EDSIGCON 2018* <<http://proc.iscap.info/2018/pdf/4633.pdf>>)
- 7896
7897
7898
7899
7900