Education and Information Technologies (2020) 25:4231–4270
https://doi.org/10.1007/s10639-020-10127-1

# Toward the visual understanding of computing curricula

Shingo Takada[1] · Ernesto Cuadros-Vargas[2] · John Impagliazzo[3] · Steven Gordon[4] · Linda Marshall[5] · Heikki Topi[6] · Gerrit van der Veer[7] · Leslie Waguespack[6]

Received: 1 October 2019 / Accepted: 30 January 2020 / Published online: 2 April 2020

**Abstract**
Various computing subdisciplines, such as computer science and software engineering, each have their own curricular guidelines. They can be very difficult to understand and compare for people such as prospective students, industry personnel, and even faculty members. This is compounded by a lack of information surrounding undergraduate computing curricular topics via visual methods. This paper describes two experimental activities where the objective is to explore the possibility of obtaining quantitative data sets necessary for visualization, one based on competencies and the other based on knowledge areas. Both activities were based on surveys. The results from the first activity showed that a consensus interpretation could be obtained for the knowledge, skills, and dispositions implied by the competency descriptions, although not as strongly for dispositions. The second activity resulted in a table of knowledge areas with minimum and maximum weights for six computing subdisciplines. Finally, this paper also shows two examples of how users can explore the various curricular guidelines through visualization.

**Keywords** Curricular visualization · Computing education · Computing competency · Global standards

## 1 Introduction

The use of digital technology as a tool is very pervasive in educational circles. At the university level, we witness the use of course assistant packages such as Blackboard and Moodle. However, use of technological tools for curricula purposes is still wanting.

✉ Shingo Takada
michigan@ics.keio.ac.jp

Extended author information available on the last page of the article

This paper explains recent activities within professional organizations to facilitate curriculum understanding for computing undergraduate (baccalaureate) programs.

### 1.1 Problem situation

In today's world, students, professional practitioners and academic personnel and teachers usually prefer to learn through visual information representations rather than acquiring information via textual methods. With the preponderance of smartphones, tablets, and other visual devices, it is only natural that dissemination of information, including curricular and learning information have similar representations. Unfortunately, little information surrounding undergraduate computing curricular topics currently exists via visual methods.

The presentation of computing curricular guidelines published by learned societies has remained stagnant and static for more than fifty years. Although PDF versions of printed copies are available, currently no automated or dynamic representation of these documents is available. To compound the situation, there is an underlying shift in the way students in computing courses should learn subject material in a marketplace where industry expects immediate performance upon hiring graduates from computing programs. The Association for Computing Machinery (ACM), the IEEE Computer Society (IEEE-CS) as well as other organizations have acknowledged not only the paucity of dynamic visualizations, but also the need to transform computing curricula from a knowledge-based approach to learning, to a competency-based (performance-based) way of learning.

To address the twofold problem, the authors have posed research questions as follows.

(1) Is it possible to generate competency-based learners from a knowledge-based setting?
(2) What would be a spectral decomposition of computing knowledge at the undergraduate level?
(3) Can we describe one or more visualization tools useful to a variety of users that can clarify different fields of computing from a competency-based viewpoint?

These underlying questions have motivated the authors to conduct two pilot experiments to obtain quantitative data sets that would be necessary for doing visualizations. They will present the results of these activities later in this work.

### 1.2 Leveraging from the past

The ACM, the IEEE-CS, and other organizations have established a project called Computing Curricula 2020 (CC2020) to update the CC2005 report (ACM et al. 2005) as the computing discipline had advanced significantly over the past fifteen years. One of the main goals of the CC2020 project is to provide intuitively expressive visualization tools to help stakeholders such as students, industry personnel, and faculty members to interpret various aspects of a specific curriculum and compare related curricula.

Another goal of the CC2020 project is clarifying the concept of competency, which is an underlying theme of the project and this paper. In general, competency refers to the *performance standards* associated with a profession or membership to a licensing organization. Competencies are what computing graduates should bring to the workplace. Any working definition of competency implies some connection of human behavior, technical skills, and knowledge. The IT2017 report (ACM and IEEE-CS 2017a) described this concept simply as:

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

within context. Further discussion on the CC2020 project and the meaning of competency appears later in this paper.

Many stakeholders may find computing curricula to be large and difficult to understand, whether they are based on conventional knowledge areas or competencies. The CC2005 overview report (ACM et al. 2005) includes static visualizations of each curricular guideline, including computer engineering (CE), computer science (CS), information systems (IS), information technology (IT), and software engineering (SE). Figure 1 shows the illustrations for CE and CS. In these illustrations, the horizontal axis represents a spectrum ranging from theory to application. The vertical axis represents a discrete layering of areas of focus spanning from computer hardware and architecture to organizational issues and systems. The CC2005 illustrations were static; that is, the CC2005 authors constructed the illustrations in advance, but users were not able to modify them. The illustrations presented in the CC2005 report have been useful for a variety of stakeholders, including faculty members, administrators, and recruiters.

It may be useful to have an interactive way for people to visualize, understand, and explore the computing curricula (Jafar et al. 2017). Toward that end, the CC2020 project is exploring how visualization tools could help interested stakeholders. Specifically, the tools should be able to allow users to pose various questions they may have concerning a specific local computing curriculum and/or established curricular guidelines. One should note that the CC2020 goal is not to have a single, all-encompassing tool, but to have a set of visualization options – a “toolbox” – to achieve the above goals.

In the rest of this paper, Section 2 provides the project background and specifically describes the CC2020 project and the questions curriculum stakeholders may want answered. Section 3 addresses related work on curricular graphics. Section 4 describes the

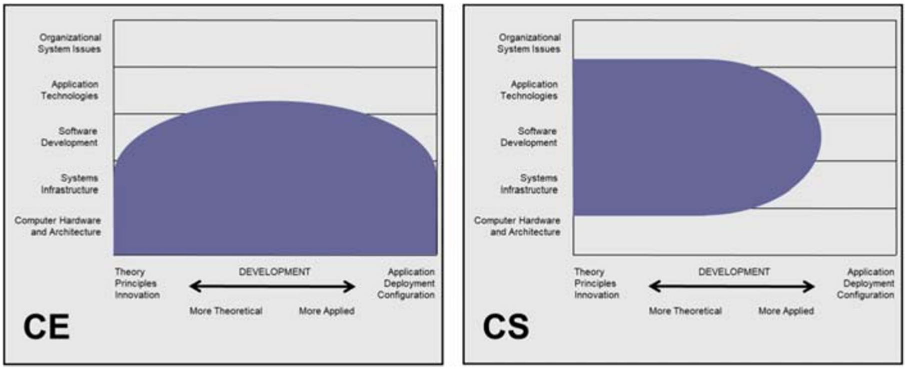


**Fig. 1** Visualizations of the CE and CS curricular guidelines in CC2005

underlying models proposed for representing curriculum in the CC2020 project. This serves as the basis for the data to be used by the toolbox. Section 5 shows examples of how users can answer questions that were posed in section 2. Section 6 presents concluding remarks.

## 2 Background

In this section, the authors describe the genesis and organization of the CC2020 project as the context of the current work. They then describe the role of competency, which is a critical component of the CC2020 project. Finally, they describe the various stakeholders of the visualizations, and what they may want to see from the visualizations.

### 2.1 The CC2020 project

In 2015, ACM considered updating the broadly influential document: *Computing Curricula 2005*, tagged CC2005 (ACM et al. 2005). ACM, the Association for Information Systems (AIS), and IEEE-CS were co-sponsors of the 2005 document. In 2016, ACM decided to proceed with the update, and established an exploratory committee to ascertain the need for a new report. ACM invited AIS and IEEE-CS to join in its development, calling the project “Computing Curricula 2020” (CC2020). Eventually, ACM and IEEE-CS became the principal sponsors of the CC2020 project with other professional organizations also joining in the effort with additional co-sponsorships.

CC2020 supports a task force of more than forty-seven academic, industry, and governmental professionals from around the world. A subset of this task force forms a steering committee of thirteen members. Currently, the task force represents seventeen countries from six continents.

The CC2020 project is examining the current state of curricular guidelines for academic programs that grant baccalaureate degrees in computing. A goal of the initiative is to produce a comprehensive report that contrasts curricular guidelines and contextualizes them in the landscape of computing education based on a framework of competency-based and knowledge-based educational guidelines. Published curricular guidelines (at this time including computer engineering, computer science, cybersecurity, information systems, information technology, and software engineering) and emerging curricular models (e.g., data science) comprise CC2020’s central domain of interest. The project also aims to provide a vision for the future of computing education.

The authors have focused on a variety of visualization models. As competency (to be described in Section 2.2) is a salient structural element of CC2020, these models include computing competency models that emphasize the meaning and use of competencies in an educational framework as well as high-level competencies to show how competencies differ from knowledge and ways to formulate competencies and illustrate their structure. The contributors also address curricular visualization models as an extension of the work presented in the CC2005 overview report (ACM et al. 2005). Ultimately, this work illustrates a comparison tool to assist in understanding computing fields and competencies within them.

Further information on the CC2020 project appears in (Impagliazzo et al. 2018a, b). The project website is under development. It includes suggestions on various ways to use the project outcomes, on identifying stakeholders who use the fruits of the project, on providing methods for quality improvement in existing programs, and on offering benchmarks for future computing programs. As the CC2020 website evolves, information will become available through textual and visual modes that provide a conceptual framework for describing some of the relationships between computing competencies, bodies of knowledge, professional profiles, educational contexts, and degree programs.

## 2.2 Need for Competency

All published baccalaureate computing curricula in CC2005 were foundationally a listing of knowledge areas, knowledge units, learning outcomes (KA-KU-LO), and topics. A knowledge area (KA) specifies "a broad category that brings together a number of knowledge units (KU)" (Topi 2017). Each KU, in turn, "can be further divided into topics". Finally, learning outcomes (LO) are "written statements of what a learner is expected to know and be able to demonstrate at the end of a learning unit (or cohesive set of units, course module, entire course, or full program)" (ACM & IEEE-CS 2017a). Thus, visual representations based on the KA-KU-LO model plausibly exhibit a degree of comparability across curriculum.

The arrival of *Information Technology Curricula 2017: Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology* (henceforth, called IT2017) (ACM & IEEE-CS 2017a) heralds a shift in specification strategy. This strategy goes beyond the basis of KA-KU-LO – specifically due to its emphasis on competency. The IT2017 project is the first of the ACM/IEEE baccalaureate curriculum projects to embrace competency as the primary characteristic of curriculum definition. *Master of Science in Information Systems 2016: Global Competency Model for Graduate Degree Programs in Information Systems* (henceforth, called MSIS2016) (ACM & AIS 2017) introduced competencies at the master's level.

One reason fueling the transition from the KA-KU-LO model to competency-based specification is the skills gap that exists between the needs of industry and the capabilities of graduates from computing programs (Radermacher et al. 2014). From any typical university, an overwhelming percent of computing graduates enter the workplace. While universities are not training grounds for industry, including arguments that industry expectations are unrealistic (Clear 2015), there is an obvious benefit in aligning baccalaureate graduates in computing with the needs of industry.

So, what is competency? The *Software Engineering Competency Model* (IEEE 2014) defines competency as the "demonstrated ability to perform work activities at a stated competency level." MSIS2016 (ACM & AIS 2017) indicates that "competencies represent a dynamic combination of cognitive and meta-cognitive skills, demonstration of knowledge and understanding, interpersonal, intellectual and practical skills, and ethical values." IT2017 (ACM & IEEE-CS 2017a, p.28) defines competency as:

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

in context, and states that "competence refers to the performance standards associated with a profession or membership to a licensing organization" and that "assessing some level of performance in the workplace is frequently used as a competence measure, which means measuring aspects of the job at which a person is competent." Other works such as Frezza et al. (2018) have adopted this definition. The approach taken in this paper also adopts this definition.

Figure 2 shows a set diagram for the meaning of competency. Descriptions of knowledge, skills, and dispositions (ACM & IEEE-CS 2017a) (Polanyi 1966) (Frezza et al. 2018) are as follows:

- Knowledge (“know-that”) designates a proficiency in core concepts and content and application of learning to new situations. A list of topics frequently represents the knowledge that a computing course covers.
- Skills (“know-how”) refer to capabilities and strategies that develop over time, through practice and interactions with others. It includes the ability to produce outcomes. Examples of skill development may be problem-based assignments and laboratory activities.
- Dispositions (“know-why”) encompass socio-emotional skills, behaviors, and attitudes that characterize the inclination to carry out tasks and the sensitivity to know when and how to engage in those tasks, i.e., how people are *disposed* to use knowledge and skills in a particular task context. This may include confidence in dealing with complexity and tolerance to ambiguity.

The prominence of competency further emphasizes that factual knowledge does not sum all the knowing to sufficiently equip a practicing professional. Competency is a familiar term in the domains of education usually classified as training and job performance assessment. Competency identifies with job recruitment, placement, and performance assessment that underpins the core of its affiliations in human resources and workforce management in the commercial and governmental arenas (Bloom and Krathwohl 1956; Dave 1970; Harrow 1972; Krathwohl et al. 1973; Wiggins et al. 2005).

Competency’s epistemological roots occur in the formal training of established labor disciplines (e.g., nursing) where the procedures and behavior employed require consistent, predictable, and disciplined application or treatments (Heath, 1998; Johns, 1995; Waguespack & Babb, 2019). They also align with the rubrics of socially acceptable conduct that circumscribe a specific profession with consequent statutory implications (e.g., licensure and legal liability). Competencies themselves are

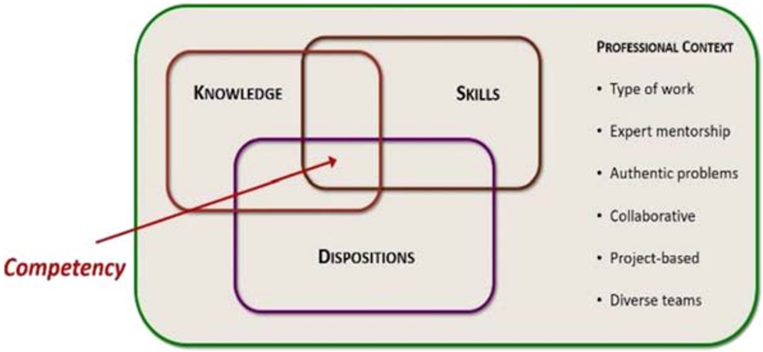


**Fig. 2** Set diagram illustrating competency

indifferent to specific pedagogy. In that sense, competency focuses on an accomplished learner's professional capacity.

### 2.3 Stakeholders and visualization - task analysis for the curriculum exploration tool

The main purpose of this paper is to provide a set of tools that can help various stakeholders intuitively understand various aspects of the curriculum. This set of tools provides (mostly visual) representations of relevant knowledge about computing curricula as may be found in the various curricular guidelines. Computing guidelines currently are available for six subdisciplines: CE, CS, IS, IT, SE, and recently, cybersecurity (CY).

The authors have identified five stakeholders for consideration. They are:

(1) Prospective students and their parents
(2) Current students
(3) Industry professionals
(4) Educators
(5) Educational authorities

The distinction between student users (1) and (2) is not absolute. Individuals in each category could sometimes have questions that are characteristic for another category.

The authors now present an overview of the different types of stakeholders, their need for knowledge about the various curricula, and a sample question intended to support curriculum exploration tools. Appendix A provides a prospective list of sample questions expected to be accommodated by these tools.

**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. They may understand that the student is interested in studying computing. But very few understand the variety of computing subdisciplines or the differences between them. A typical question by a prospective student could be:

> *I am considering a computing curriculum that fits my preferences. My candidate schools offer several computing programs. Are graduates of these programs expected to work primarily as individuals (e.g., writing code) or work with other people?*

Section 5 illustrates an interactive visualization design for answering this type of question.

**Current students** are students that are enrolled at an institution of higher education. They might consider a choice of courses from their own institute or another institute (in some cases another department when they intend to take a hybrid curriculum of multiple subdisciplines), or in another country. Alternatively, they may completely move to another educational institution. This category could also apply to students in another subdiscipline that are considering a hybrid curriculum that includes computing components. A typical question by a current student could be:

> *Which courses in curriculum X at a given university are emphasized more (at a higher level, or longer duration) compared to established guidelines for curriculum X?*

**Industry** refers to organizations that (1) are hiring graduates, (2) are collaborating with universities to choose or specialize a curriculum or need a tailor-made course, or (3) are collaborating in a curriculum by providing internships. Most importantly, industry professionals and recruiters need to understand what incoming employees have learned. Computing professionals need various specific skills. For example, employers who are looking for software developers desire people that have taken courses with a strong focus on software development, and thus they are interested in software engineering graduates. On the other hand, if the employers want people who have studied organizational impact of technology in addition to a foundation in computing, then they would prefer graduates from an information systems curriculum. Thus, understanding how particular types of curricula would fit within their employer needs would help target which type of graduates they prefer in terms of curriculum studied. One sample question that industry stakeholders might want to have answered is:

> *Our industry requires employees to have knowledge in specific knowledge areas, with relevant knowledge levels and some specific dispositions. Do the courses in curriculum X develop competences that are appropriate for continued professional education for our employees?*

Section 5 illustrates an interactive visualization design for answering this type of question.

**Computing educators** are teachers and staff within a single school or university, who are responsible for designing and implementing educational offerings (a degree program or an individual course or module as part of one or more curricula). These people may be individual university faculty members or teams that design and teach courses, design educational resources (books, massive open online courses (MOOCs), websites, presentation slide decks), manage curricula as taught in their school, or assess student entry or exit levels. Computing educators need to understand how their current or prospective curriculum fits with standard curriculum recommendations. They may think that their curriculum is computer science. However, being able to compare their curriculum with established guidelines would help them understand how well their curriculum aligns with the CS2013 curriculum guidelines (ACM and IEEE-CS 2013). A sample question that educators may like answered would be as follows.

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

**Educational authorities** are organizations that have authority over university education such as (national) 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 or certificates. A typical question that educational authorities may like answered is:

*Does this curriculum comply to the guidelines for curriculum X? What should be changed and how?*

## 3 Related work on curricular visualizations

This section describes completed work on curricular visualizations, which is the core of this paper. The authors first provide examples of work already completed on visualizing university curricula. They then describe visualization for curricular guidelines.

### 3.1 Visualization of university curricula

Many universities have visualizations of their university curricula. One type of visualization shows the relationships between courses, so that students can easily understand the prerequisites of courses, and help them choose which courses to take. Siirtola et al. (2013) used a node-link structure to visualize the topics covered by each course. Sommaruga and Catenazzi (2007) developed a 3D environment for visualizing curriculum data, such as number of credits and duration. Zucker (2009) developed a tool called ViCurriAS to enable faculty to construct a curriculum map, and allow advisors and students to see course dependencies and student progress. Cuadros-Vargas (2018a) used a graph representation (Fig. 3) to describe the complete sequence of prerequisites for a given curriculum at Universidad de Ingenieria y Tecnologia (UTEC) in Lima, Peru. In Fig. 3, the thick lines represent the critical paths which are useful for students to plan their studies, detect where the bottlenecks may occur, and to pay greater attention to courses that are most critical. The illustration allows students to see that there may be more than one critical path with the same distance.

Other works visualized the university curriculum to compare it with some type of standard. Willcox and Huang (2017) used Rhumbl,[1] a free online tool, to visualize the mapping of an undergraduate program to the Conceive Design Implement Operate (CDIO)[2] syllabus. Their visualization allows users to know which CDIO skills (such as "problem identification and formulation" or "modeling") would be introduced, taught and applied in each course.

Cuadros-Vargas (2018b) developed several visualizations to compare UTEC curriculum against CC2005 in terms of relative weight of each topic. In Figs. 4 (a) and (b), the generation of the yellow region reflected Min and Max values proposed in CC2005 (ACM et al. 2005). The black line represents the distribution for the current curricula being analyzed. These figures are particularly useful to detect visually areas in the curricula that need improvement. They can also show areas considered to be strong. Since program names do not always reflect the content of the program, such figures can be particularly useful for understanding the content of the program. Instead of discussing the name, educators can see what a program covers based on content.

Another type of work is to visualize the Skills Framework for the Information Age (SFIA)[3] skill sets using spider diagrams (von Konsky et al. 2008) or heat maps (Armstrong 2013) for various types of jobs. SFIA itself is a common reference model in a two-

[1] https://rhumbl.com/

[2] http://www.cdio.org

[3] https://www.sfia-online.org

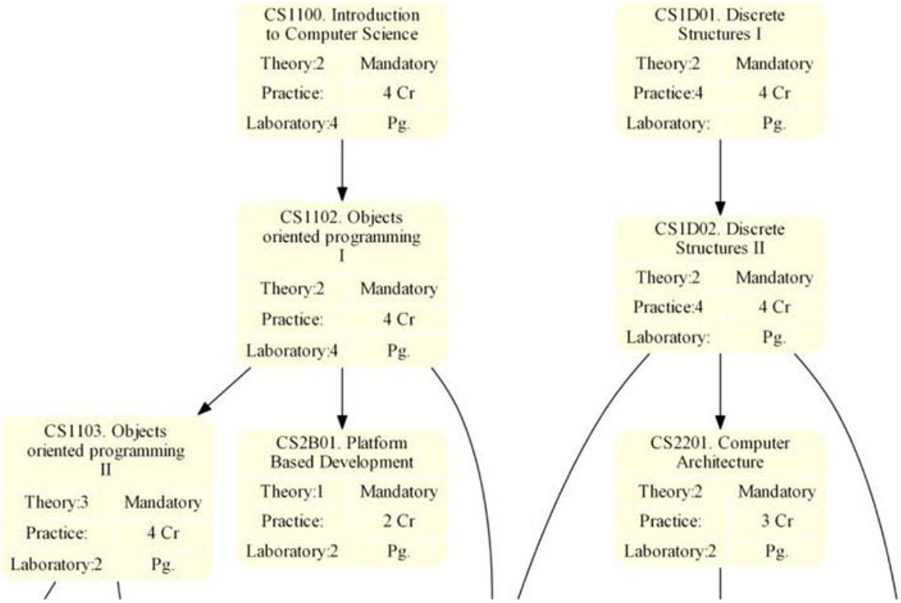
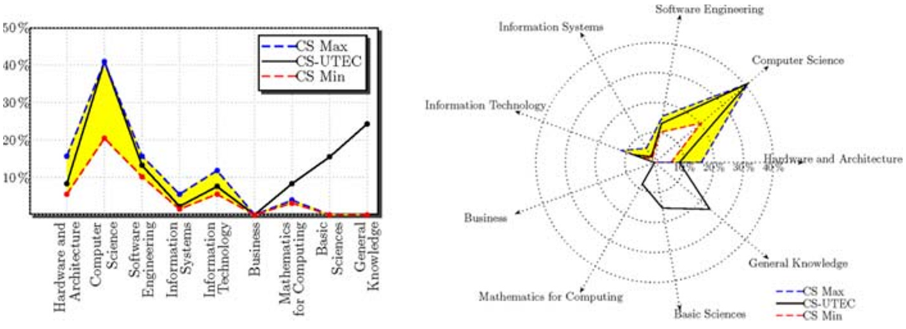


**Fig. 3** Partial Curriculum Map for UTEC. [English version of (Cuadros-Vargas 2018a)].

dimensional framework consisting of skills on one axis and seven levels of responsibility on the other. For von Konsky's approach, each spoke in the spider diagram corresponded to a skill with the distance from the center showing the responsibility level. In Armstrong's method, each cell in the heat map corresponded to a skill, with a color showing the responsibility level. Von Konsky et al. (2013) posited that such visualizations could be useful so graduates can better prepare themselves for professional practice.

## 3.2 Visualization of curricular guidelines

The CC2005 report provided graphical representations for each of CE, CS, IS, IT, and SE subdisciplines. Earlier, Fig. 1 showed the visualizations for CE and CS. In each case, the horizontal axis showed a spectrum ranging from "Theory, Principles,



(a) Line graph-based visualization (b) Radar chart-based visualization

**Fig. 4** Comparison of UTEC Curriculum with CC2005. [English version of (Cuadros-Vargas 2018b)].

Innovation" to "Application, Deployment, Configuration"; the vertical axis showed five layers: "Computer Hardware and Architecture," "Systems Infrastructure," "Software Methods and Technologies," "Application Technologies," and "Organizational Issues and Information System". The visualizations clearly showed how the focus differed between the two different subdisciplines.

Marshall (2014, 2017) modelled curricula as a graph-based structure. Figure 5 shows an example of about one-fourth of a visual model of this graph-based structure showing the core components of CS2013 (ACM & IEEE-CS 2013). This visualization centers around the CS node that links the knowledge areas (KAs), their core knowledge units (KUs) and their respective topics. The reader can easily see that it would be difficult to understand this structure *as is*, due to the size and amount of information included within it. It would be even more difficult to recognize differences between curricula modelled using the visualization of the graph-based structure. Thus, Marshall (2012, 2014) took an algorithmic approach to reduce the complexity of the graph-based curriculum models (as seen in Fig. 5) to scalar values on spokes of a radar chart as shown in Fig. 6.

Figure 6 compares CC2001 (ACM and IEEE-CS 2001) and CS2013. In this figure, the blue region (benchmark) shows what the shape would be if CS2013 and CC2001 were identical. The closer the green and red regions come to the blue region, the better the match between the curricula being compared. If a region exceeds that of the blue region on a spoke, it means that new material became part of the curriculum under comparison. A region that falls within the blue lines means that the curriculum under comparison excludes aspects that are in the curriculum to which it is being compared. For example, the green region (comparison with KA and KU equivalences) shows that there are many topics in CS2013 which are not present in CC2001. Note that equivalences are concepts which have different terminology but have the same meaning. For example, between CC2001 and CS2013, "software requirements and specifications" changed to "requirements engineering" for the KU "requirements" link. If we also include topic equivalences (red region), the difference between CC2001 and CS2013 becomes smaller. When considering what causes these differences, insight into what is included or excluded from the curriculum being compared can be gained.

## 4 Research methodology for curriculum visualizations

Data form the basis of visualization. Unfortunately, the authors could not use the curricular guidelines *as is* since the same words may have different meanings between curricula. For example, *refactoring* appears in both CS2013 (ACM & IEEE-CS 2013) and SE2014 (ACM and IEEE-CS 2015). In CS2013, the word *refactoring* is part of "Software Design" (pp. 180–181) and "Software Evolution" (pp. 183–184). In SE2014, *refactoring* is part of "Software Process" (page 35). This suggests that *refactoring* may mean different things in the two curricula.

To derive the visualized data descriptions, the authors describe two experimental activities to pre-process the base data sets (i.e., the curricula themselves). Section 4.1 describes a pilot activity used to quantify competency. The instrument used is a survey conducted to assess whether there was a consensus interpretation of knowledge, skills, and dispositions, i.e., the three components of competency. The participants included

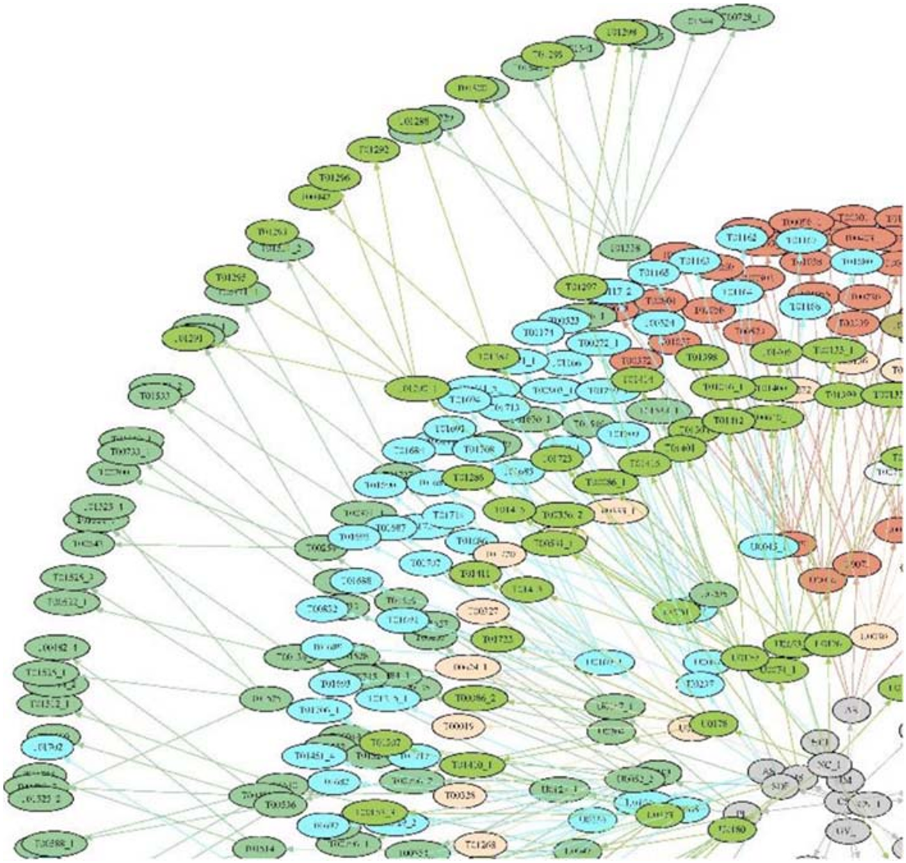


**Fig. 5** Core aspects of the CS2013 curriculum volume

members of the CC2020 steering committee who received components of the three elements of competency. Section 4.2 describes a second pilot study to quantify the stratification of computing knowledge. The instrument used was a survey that addressed knowledge components with a min-max scale of significance for each of six computing subdisciplines. The participants included members of the CC2020 steering committee who received elements of computing knowledge areas for the six computing subdisciplines. A summary of these two pilot experiments follows.

### 4.1 Expert-based data description of Competency

In order to compare multiple curricula, we need to categorize and abstract the data in some way. The data should be able to be used to visualize the major characteristics of each of the computing curriculum. However, the categorization of the competencies into these categories is not straightforward as the language used to define the competencies is not consistent either within or across the curricula. To date, no related work appears in the literature. Thus, we conducted a pilot assessment based on the IT2017 competencies (ACM & IEEE-CS 2017a). The pilot assessment took the form of a survey and was intended to provide insights on the relationships of the competencies

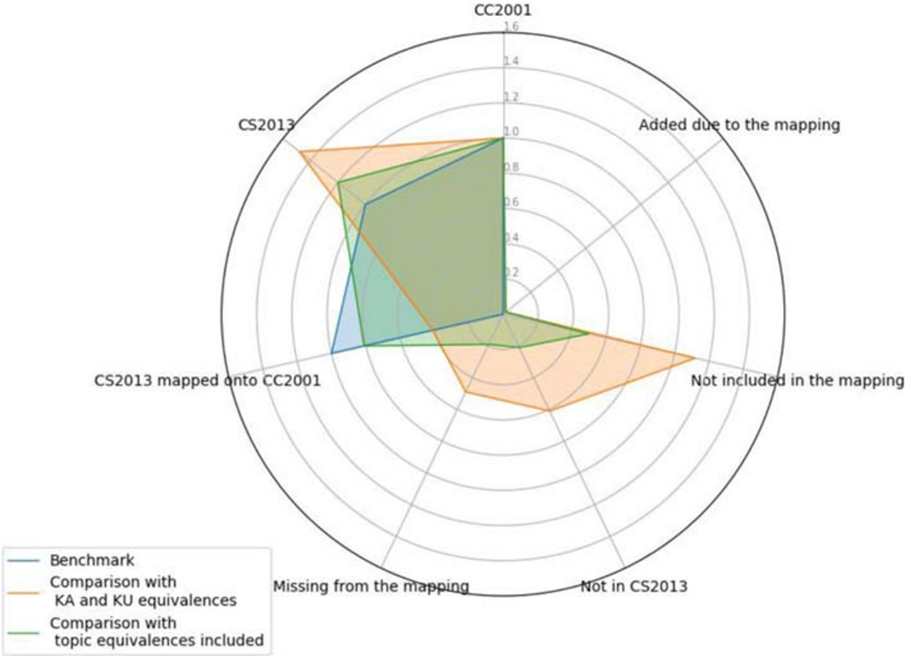


**Fig. 6** CC2001 compared with CS2013

defined in IT2017 to the knowledge, skills, and dispositions that are the basis of past curricular models. We wished to ascertain whether there was a consensus interpretation of the knowledge, skills, and dispositions implied by the competency descriptions. We chose the IT2017 competencies, as it had already been published and been in use when our work started.

### 4.1.1 Methodology

The pilot assessment used a questionnaire distributed to the entire CC2020 task force. Instead of taking an open-ended approach (asking, e.g., “What are the knowledge, skills, and dispositions in competency statement A?”), we focused on a closed question approach where the respondents can choose from a list of categories for each of knowledge, skills, and dispositions. Hence the first part was to define categories for knowledge, skills, and dispositions, from which the respondents can choose from. The authors characterized and compared the six curricular guidelines to categorize the knowledge, skills, and dispositions associated with each of the competencies. The competencies consist of noun and verb combinations where the nouns represent the knowledge and skills that are the basis of the model curricula. The verbs imply the depth of expertise associated with each of the knowledge and skill areas. The authors grouped the competencies in each curriculum guidelines according to a topic; each group of competencies also implied one or more dispositions. The authors considered a similar list in the IT2017 report as well as Bloom’s taxonomy (Bloom and Krathwohl 1956) (Krathwohl et al. 1973). This resulted in four categories for knowledge, seven for skills, and ten dispositions. Appendix B shows the categories and their definitions.

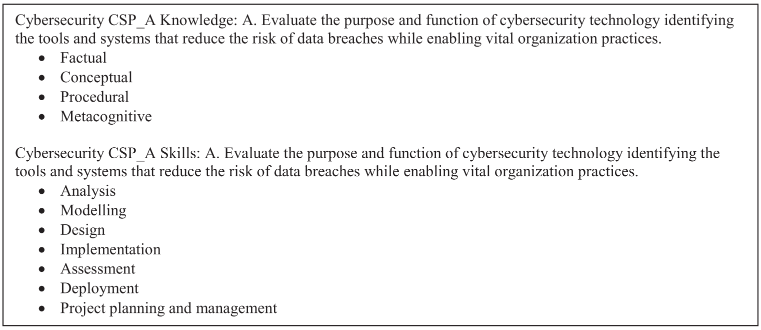
The authors then took each of the competencies from IT2017 and formulated two questions for each. The first question asked respondents to classify the knowledge associated with the competency into one of the four categories shown in Appendix B. The second question asked them to indicate from a skills table one or more skills associated with that competency. Figure 7 shows an example question pair. Appendix C shows the list of competencies that were used in the pilot. At the end of each category of competencies, an additional question asked the respondent to identify one or more of the dispositions shown in Appendix B associated with that group of competencies. Finally, a series of questions concerning the overall classification scheme was part of the survey, some of which were open-ended questions.

The survey was distributed to the 36 members of the CC2020 task force. Those individuals represent a significant sample of educators and professionals across the computing disciplines. Since there were almost 150 questions, it was unlikely that people would take the time to answer all questions. Thus, the list of questions was divided into two parts. Several questions were put into both groups. The questions were then distributed to the 36 task force members with 18 assigned to each of the two parts.

#### 4.1.2 Pilot results

Eight people responded to one part and ten people responded to the other part. Furthermore, respondents could opt to "skip" questions; thus some questions did not attain full response. The small size of the pilot sample and the way it was obtained prevented us from using inferential statistics. Nevertheless, the pilot survey provided some interesting insights into the competencies and this approach to characterize a curriculum. The generated results show percentages, but the reader should be aware that, for example, "at least 80% of the respondents" may mean four out of five people responded to one question or nine out of eleven people who responded to another question.

There was much variability in the selection of knowledge and skill areas associated with most of the questions. For the knowledge area, at least 80% of the respondents chose the same category in only seven out of the 47 competency statements (See



Cybersecurity CSP_A Knowledge: 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.

- Factual
- Conceptual
- Procedural
- Metacognitive

Cybersecurity CSP_A Skills: 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.

- Analysis
- Modelling
- Design
- Implementation
- Assessment
- Deployment
- Project planning and management

**Fig. 7** Example Knowledge and Skill Question Pair from IT2017

Appendix D.1). There was only one case where all respondents chose the same category; for the competency "SS. Show how to choose among operating system options and install at least an operating system on a computer device" all six respondents chose the category 'Procedural'. Still, the majority of questions had a strong tendency toward a single knowledge category. In 39 out of the 47 questions, at least 50% of the respondents chose the same knowledge category (See Appendix D.2). This outcome still leaves much variation across the other categories chosen.

Similarly, in the skills questions, 35 of the questions had two skill categories that in total comprised more than 50% of the responses (See Appendix D.3). For example, for the competency "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," two out of eight responses were for the 'Analysis' category and three were for the 'Assessment' category. However, there were still many questions where the skills chosen were distributed relatively equally across the seven categories.

The tendencies with the disposition question showed greater variation. Only 1 out of the 11 questions had two characteristics that totaled more than 50% of the choices available. This is probably because the competency statements rarely make explicit references to dispositions, leaving it to the judgement of the respondent whether they are implied.

The final set of summary questions related to the categorizations underlying the survey. Appendix E shows the results for those questions. Here, one can see that there was a very strong agreement that the category choices for knowledge and skills are appropriate. Almost all the respondents provided a response of yes or mostly yes to those questions. Similarly, the respondents agreed that the categorization could be useful to the description of other computing subdisciplines. Finally, the respondents felt the list of dispositions was mostly complete. There were a series of open-ended questions which asked if there were any other dispositions that were missing from the list. There was only one person who commented on the disposition list and that comment did not offer any additional items.

### 4.1.3 Discussion of pilot

The survey provides a mixed picture of the efficacy of using competencies to characterize the implied knowledge, skills, and dispositions related to the curriculum. On the positive side, the responses showed that at least 50% of the responses clustered in the same categories. Most respondents agreed that the categories for the knowledge, skills, and dispositions are appropriate. Nevertheless, there was an extensive amount of dispersion from the central tendencies. Thus, it may not be possible to use these results to help "train" a natural language processing tool to help characterize the similarities and differences among computer curricula based on the categorizations given in the pilot.

The most likely explanation for the wide range of responses is that the language and scope of the competencies are prone to different interpretations of the required knowledge and skills. Indeed, the competencies intend to be a general guideline to those preparing a curriculum and thus not constructed in a way that would prescribe a single specific interpretation. Furthermore, there have been reports that reaching agreement on

Bloom's level for even a constrained domain of introductory programming exercises can be challenging (Whalley et al. 2006).

The dispersion of responses with respect to dispositions points to a broader problem. Most of the competencies do not explicitly address dispositions. Dispositions are not only important to industry but are also clearly needed for long-term professional success, and even for personal growth and development of character and citizenship. In that regard, it may be necessary for future curricular documents to indicate explicitly the scope and breadth of dispositions in computing curricular graduates.

Additionally, even though no respondent offered additional dispositions, this does not mean that no other characteristics exist. Indeed, a wide variety of dispositions have been proposed (Perkins et al. 1993) (Clear 2017). Along with knowledge and skills, the CC2020 taskforce is actively discussing what dispositions should be within the scope of the CC2020 project.

Note that the pilot respondents were from the CC2020 project task force. Thus they are more likely to respond similarly compared to the population-at-large. Although the number of total respondents itself was 18, and thus the sample size is limited from a structured population, the 40% response rate provides credibility with statistical confidence. Thus, the results shown in Appendix D are important as they demonstrate that a consensus of categorization is achievable. The authors plan on conducting a formal survey targeting a broad international community after applying the improvements based on the pilot such as limiting the number of questions per respondent to increase the response rate as well as reviewing and revising the current categorizations. We are envisioning the results of the formal survey to help create a training set for a natural language processing program that applies the results across all the curricula. The results of the formal survey may also be used to help define a more consistent set of principles for future curriculum reviews that enable more straightforward interpretations of the curricular guidelines.

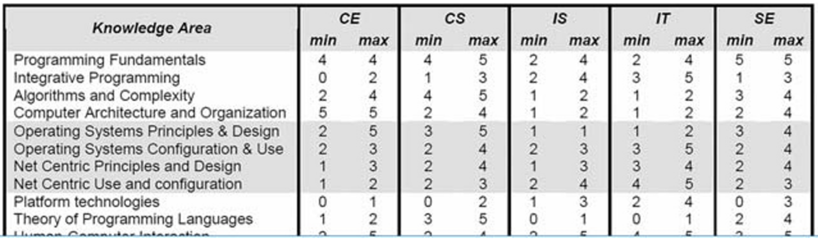
### 4.2 Expert-based delineations of knowledge areas

One of the widely used features in the CC2005 report were tables concerned with knowledge areas, each presenting a different numeric comparison between the five computing disciplines included in CC2005. As more than 10 years had passed since CC2005 was published, the task force envisioned that there would be changes in the knowledge areas. This section thus first describes the knowledge area tables in CC2005, and then our attempt in updating the knowledge areas.

#### 4.2.1 Previous work: Knowledge areas in CC2005

Table 1 is an excerpt of Table 3.1 in (ACM et al. 2005). It provided a comparison of comparative weights of computing knowledge areas, giving two numeric evaluations (minimum and maximum) for each of the computing knowledge areas. For example, Table 1 indicated that computer engineering has a minimum weight of 2 and maximum weight of 5 for *Operating Systems Principles & Design*, whereas information systems has a minimum of 1 and a maximum of 1 for the same knowledge area. For *E-business*, the corresponding values for CE were 0 and 0 and for IS 4 and 5. Collectively, the weights formed a reasonable mechanism for comparing the relative importance of

**Table 1** Comparative weight of computing topics.



| Knowledge Area | CE | | CS | | IS | | IT | | SE | |
|---|---|---|---|---|---|---|---|---|---|---|
| | min | max | min | max | min | max | min | max | min | max |
| Programming Fundamentals | 4 | 4 | 4 | 5 | 2 | 4 | 2 | 4 | 5 | 5 |
| Integrative Programming | 0 | 2 | 1 | 3 | 2 | 4 | 3 | 5 | 1 | 3 |
| Algorithms and Complexity | 2 | 4 | 4 | 5 | 1 | 2 | 1 | 2 | 3 | 4 |
| Computer Architecture and Organization | 5 | 5 | 2 | 4 | 1 | 2 | 1 | 2 | 2 | 4 |
| Operating Systems Principles & Design | 2 | 5 | 3 | 5 | 1 | 1 | 1 | 2 | 3 | 4 |
| Operating Systems Configuration & Use | 2 | 3 | 2 | 4 | 2 | 3 | 3 | 5 | 2 | 4 |
| Net Centric Principles and Design | 1 | 3 | 2 | 4 | 1 | 3 | 3 | 4 | 2 | 4 |
| Net Centric Use and configuration | 1 | 2 | 2 | 3 | 2 | 4 | 4 | 5 | 2 | 3 |
| Platform technologies | 0 | 1 | 0 | 2 | 1 | 3 | 2 | 4 | 0 | 3 |
| Theory of Programming Languages | 1 | 2 | 3 | 5 | 0 | 1 | 0 | 1 | 2 | 4 |

((ACM et al. 2005) page 24, Table 3.1 [excerpts])

different knowledge areas for different subdisciplines. Another table (Table 3.2 in (ACM et al. 2005)) provided similar weights for non-computing topics (including organizational/business topics that are particularly important for IS and electronics topics particularly significant for CE).

The knowledge areas included in these tables were created by first collecting all knowledge area level elements from the curriculum documents that existed in 2005, and then resolving potential naming conflicts between the subdisciplines. For the resulting set of knowledge areas, the CC2005 task force collectively determined the comparative weights based on a comprehensive evaluation of the relative emphasis on each knowledge area in each of the curriculum recommendations. This evaluation was not only based on a specific numeric value (e.g., number of core hours) but it also took into account the nature of the coverage during the time dedicated to the knowledge area.

### 4.2.2 Methodology: Updating knowledge areas for CC2020

The CC2020 steering committee decided to create a new version of these tables. There were some differences in how they derived the knowledge categories and the values in them compared to the CC2005 process. In CC2005, the work to create these tables took place in face-to-face meeting discussions among the task force members. In CC2020, the process was more elaborate: it had multiple stages and was partially distributed. The stages were as follows.

1. At a face-to-face meeting, a group within the steering committee created an aggregated collection of computing knowledge areas from the underlying curriculum recommendations (CE2016 (ACM and IEEE-CS 2016), CS2013 (ACM et al. 2013), CSEC2017 (ACM and IEEE-CS 2017b), IS2010 (ACM and AIS 2010), IT2017 (ACM et al. 2017a), and SE2014 (ACM & IEEE-CS 2015)). They first extracted the knowledge areas from the curriculum documents and then identified overlapping knowledge areas using both high-level definitions of the knowledge areas and an evaluation of the knowledge units included in the knowledge areas in various reports.
2. After the face-to-face discussion, they developed an online survey that asked survey participants to identify the minimum and maximum weights for each of the new CC2020 computing knowledge areas based on the participant's understanding of the relative emphasis of each of the knowledge areas for each of the subdisciplines of computing (minimum reflecting the lowest relative weight acceptable for the subdiscipline and the maximum reflecting the highest weight one

could reasonably expect for a program in a subdiscipline). The knowledge areas were ordered alphabetically. The participants had an opportunity to consult high-level definitions for the knowledge areas.

3. They administered the survey described above to the members of the steering committee to obtain a baseline set of values corresponding to those in CC2005.
4. Three expert members of the steering committee worked to develop a sequencing of the knowledge areas that would offer scale-like properties and a sense of continuity between the areas. The concept of semiotic ladder (Stamper 1991) provided a theoretical foundation for this process. The semiotic ladder consists of six categories: physical, empirics, syntactics, semantics, pragmatics, and social world. In addition, the experts' understanding informed the clustering by using dependencies between the knowledge areas and their interpretation of the level of abstractness of the areas. The experts first placed each knowledge areas into one of the six clusters, after which they named the clusters and further ordered the knowledge areas within the clusters to achieve a better continuity for the pseudo-scale that emerged from this process.

#### 4.2.3 Results

Table 2 shows the resulting table, consisting of the 36 knowledge areas each of which is grouped into one of six clusters (knowledge categories). The table also shows the minimum and maximum weights for each of the knowledge areas in each of the six computing subdisciplines. For each knowledge area, the highest values are highlighted in yellow and the lowest values in light blue. Note that the table also includes the new discipline for cybersecurity (CY) as a new column compared to the CC2005.

#### 4.2.4 Discussion

The final set of 36 knowledge areas included seven new or significantly restructured and two with substantive name changes. Seven knowledge areas from CC2005 did not appear any more in CC2020. Table 3 includes the details of the changes.

Some of the new knowledge areas are concerned with topics that did not exist (or just started to exist) in 2005, such as cloud computing and internet of things, and this is to be expected as the computing discipline evolves. What is especially interesting is that there are also areas that existed in 2005 but were not included in CC2005, such as software development fundamentals and systems fundamentals.

It is also interesting to note the areas that were not included in the final set of 36 CC2020 areas. This does not mean that knowledge areas such as digital media development or scientific computing are no longer important. Instead, it indicates a shift of focus within the undergraduate curricula.

The first version of the knowledge area table (after stage 1) was organized in an alphabetical order, and this was used when developing and administering the survey (stages 2 and 3). This sequencing of the knowledge areas was not, however, considered to be helpful for the purposes of visualization or any other use that requires scale properties, and this was the primary reason for stage 4. Table 2 reflects the resulting sequencing, which includes the six knowledge categories that roughly correspond to those of the semiotic ladder. It is important to note that the clustering and further sequencing concluded

**Table 2** Relative Weights of Computing Knowledge Areas

| Knowledge Category | Knowledge Area | CE | | CS | | CY | | IS | | IT | | SE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max |
| **1. Users and Organizations** | 1.1. Social Issues and Professional Practice | 2 | 5 | 2 | 4 | 2 | 4 | 3 | 5 | 2 | 4 | 3 | 5 |
| | 1.2. Security Policy and Management | 1 | 3 | 2 | 3 | 4 | 5 | 2 | 3 | 2 | 4 | 2 | 4 |
| | 1.3. IS Management and Leadership | 0 | 2 | 0 | 2 | 1 | 2 | 4 | 5 | 1 | 2 | 1 | 2 |
| | 1.4. Enterprise Architecture | 0 | 1 | 0 | 1 | 1 | 2 | 3 | 5 | 1 | 3 | 1 | 3 |
| | 1.5. Project Management | 1 | 3 | 2 | 3 | 1 | 2 | 4 | 5 | 2 | 3 | 2 | 4 |
| | 1.6. User Experience Design | 1 | 3 | 2 | 4 | 1 | 3 | 2 | 4 | 3 | 4 | 3 | 5 |
| **2. Systems Modeling** | 2.1. Security Issues and Principles | 2 | 3 | 2 | 3 | 4 | 5 | 2 | 4 | 3 | 4 | 2 | 4 |
| | 2.2. Systems Analysis & Design | 1 | 2 | 1 | 2 | 1 | 2 | 4 | 5 | 1 | 3 | 2 | 4 |
| | 2.3. Requirements Analysis and Specification | 1 | 2 | 1 | 2 | 0 | 2 | 2 | 4 | 1 | 3 | 3 | 5 |
| | 2.4. Data and Information Management | 1 | 2 | 2 | 4 | 2 | 3 | 3 | 5 | 2 | 3 | 2 | 4 |
| **3. Systems Architecture and Infrastructure** | 3.1. Virtual Systems and Services | 1 | 3 | 1 | 3 | 1 | 2 | 1 | 2 | 3 | 4 | 1 | 3 |
| | 3.2. Intelligent Systems (AI) | 1 | 3 | 3 | 5 | 1 | 2 | 1 | 2 | 1 | 2 | 0 | 1 |
| | 3.3. Internet of Things | 2 | 4 | 0 | 2 | 1 | 3 | 1 | 3 | 2 | 4 | 1 | 3 |
| | 3.4. Parallel and Distributed Computing | 2 | 4 | 2 | 4 | 1 | 2 | 1 | 3 | 1 | 3 | 2 | 3 |
| | 3.5. Computer Networks | 2 | 4 | 2 | 4 | 2 | 4 | 1 | 3 | 3 | 4 | 2 | 2 |
| | 3.6. Embedded Systems | 3 | 5 | 0 | 2 | 1 | 3 | 0 | 1 | 0 | 1 | 0 | 3 |
| | 3.7. Integrated Systems Technology | 1 | 2 | 0 | 2 | 0 | 2 | 1 | 3 | 3 | 4 | 1 | 3 |
| | 3.8. Platform Technologies | 0 | 1 | 1 | 2 | 1 | 2 | 1 | 3 | 2 | 4 | 0 | 2 |
| | 3.9. Security Technology and Implementation | 2 | 3 | 2 | 4 | 4 | 5 | 1 | 3 | 2 | 4 | 2 | 4 |
| **4. Software Development** | 4.1. Software Quality | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 3 | 1 | 2 | 3 | 5 |
| | 4.2. Software Development and Management | 1 | 2 | 1 | 3 | 1 | 2 | 2 | 3 | 2 | 3 | 3 | 5 |
| | 4.3. Software Verification and Validation | 1 | 3 | 1 | 3 | 1 | 2 | 1 | 3 | 1 | 2 | 4 | 5 |
| | 4.4. Software Process | 1 | 2 | 2 | 3 | 0 | 2 | 1 | 3 | 1 | 2 | 3 | 5 |
| | 4.5. Software Modeling and Analysis | 1 | 3 | 1 | 3 | 1 | 2 | 2 | 4 | 1 | 3 | 4 | 5 |
| | 4.6. Software Design | 2 | 4 | 2 | 4 | 1 | 3 | 1 | 3 | 1 | 2 | 4 | 5 |
| | 4.7. Platform-Based Development | 0 | 2 | 2 | 4 | 0 | 1 | 1 | 3 | 2 | 4 | 1 | 3 |
| **5. Software Fundamentals** | 5.1. Graphics and Visualization | 1 | 2 | 2 | 4 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 2 |
| | 5.2. Operating Systems | 2 | 4 | 3 | 5 | 2 | 3 | 1 | 2 | 1 | 3 | 1 | 3 |
| | 5.3. Algorithms and Complexity | 2 | 4 | 4 | 5 | 1 | 3 | 1 | 3 | 1 | 2 | 2 | 4 |
| | 5.4. Programming Languages | 2 | 3 | 3 | 5 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 3 |
| | 5.5. Software Development Fundamentals | 2 | 4 | 4 | 5 | 2 | 3 | 1 | 3 | 2 | 4 | 3 | 5 |
| | 5.6. Systems Fundamentals | 2 | 3 | 2 | 3 | 1 | 2 | 2 | 3 | 1 | 3 | 2 | 3 |
| **6. Hardware** | 6.1. Architecture and Organization | 4 | 5 | 3 | 4 | 1 | 3 | 1 | 2 | 1 | 2 | 1 | 3 |
| | 6.2. Digital Design | 4 | 5 | 1 | 2 | 0 | 2 | 0 | 1 | 0 | 1 | 0 | 2 |
| | 6.3. Circuits and Electronics | 4 | 5 | 1 | 2 | 0 | 1 | 0 | 1 | 1 | 2 | 0 | 1 |
| | 6.4. Signal Processing | 3 | 4 | 0 | 1 | 0 | 2 | 0 | 1 | 0 | 1 | 0 | 1 |

without access to the values linking the knowledge areas to the computing subdisciplines. Therefore, the way the maximum disciplinary row values align with the clusters provides evidence regarding the robustness of the clustering process.

The administration of the survey to the steering committee members did not reveal any substantive problems with the knowledge area. The steering committee decided that it would be important to obtain a broader perspective on the relative weights. Thus as part of future work, the members of the full task force will complete the survey with one important change: during this new round, each respondent will assign weights only for those subdisciplines that the respondent had at the beginning of the survey declared their own.

**Table 3** Differences between knowledge areas in CC2005 and CC2020

| Knowledge areas in CC2005 but not in CC2020 | Knowledge areas in CC2020 but not in CC2005 | Significantly renamed knowledge areas |
|---|---|---|
| • Computer Systems Engineering<br>• Digital Media Development<br>• E-business<br>• Engineering Economics for SW<br>• Engineering Foundations for SW<br>• Scientific computing (numerical methods)<br>• Systems administration/Technical support | • Computing Infrastructure/Cloud Computing<br>• Enterprise Architecture<br>• Internet of Things<br>• Platform-Based Development<br>• Software Development Fundamentals<br>• Systems Fundamentals<br>• Virtual Systems and Services | • Information Systems Development ➔ Systems Analysis & Design<br>• Human Computer Interaction ➔ User Experience Design |

## 5 Understanding curriculum based on visualization

A visualization-based tool is currently under development as a web-based application to help users answer questions posed in Section 2.3. This section shows two example cases of how one can use this tool.

The knowledge areas follow the 36 areas given in Section 4.2.3. These may be explored to find competence descriptions that are related to them in a specific curriculum. The description may include one of the following skill levels (Krathwohl et al. 1973) recommended for a knowledge area in each curriculum, where each next level includes all previous levels. They include the following:

| | | |
|---|---|---|
| 1. Remember | 2. Understand | 3. Apply. |
| 4. Analyze | 5. Evaluate | 6. Create. |

The tool also allows to consider the dispositions described for each competency. Note that the concept of disposition is still evolving and may change. Currently, we consider the following dispositions. See Appendix F for their descriptions.

| | | | | |
|---|---|---|---|---|
| Proactive | Self-directed | Passionate | Purpose-driven | Professional |
| Responsible | Adaptable | Collaborative | Responsive | Meticulous |

The general interactive visualizations allow the user different actions as follows.

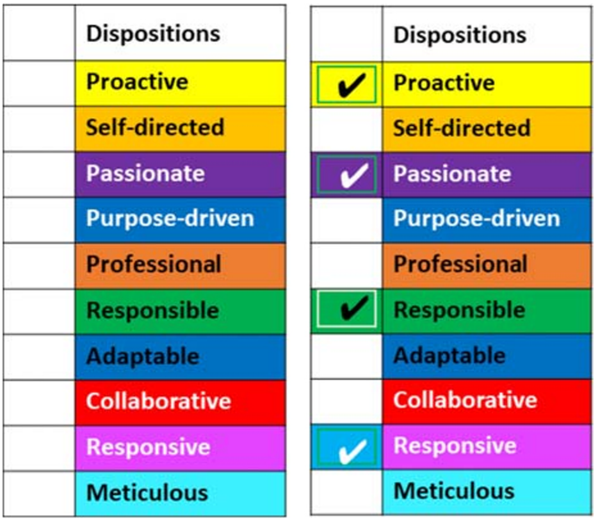
1. Hovering over a concept (such as with a mouse) provides an explanation of the concepts, and examples if relevant;
2. Clicking on a concept results in a chosen mark for this concept, clicking again reverses this action;
3. Clicking on a confirm button (not represented in the illustrations below) confirms all kept choices on the current screen and results in displaying the next screen in the dialogue.
4. The user may always go back to a previous phase to reconsider any selection.

The tool provides different interfaces and interactive visualizations for the different stakeholders. The explanations and examples indicated in hovering action #1 above adjusts to the type of stakeholder and, if relevant, supports understanding of the concepts in relation to the generally accepted labels and terminology in the academic education culture and standards as understood in the user's country, if required.

### 5.1 Case 1: A question from a prospective student

A student is interested in entering undergraduate education in computing, and wants to know what type of curriculum would best fit her interests. She might have some ideas about dispositions that are relevant in her future curriculum, and/or have a preliminary view on domains that would provide her with future job opportunities. She might start by checking promising dispositions (or, alternatively, she could start by choosing the knowledge categories and areas – we show only the first scenario but the alternative would lead to the same results). She would see a list of dispositions (Fig. 8(a)), from which she would choose, resulting in the interface showing the chosen dispositions as shown in Fig. 8(b). Note that the dispositions are indicated by color, as there is no order dimension.

The student may also indicate which knowledge categories and knowledge areas seem interesting for her. Figures 9 and 10 show a possible process. She first chose three categories: *Users and Organizations*, *Systems Modeling*, and *Software Fundamentals*. In Fig. 9, the ellipse of these three categories are highlighted with red borders. If needed, the student could indicate which individual knowledge areas are most relevant. Figure 10(a) shows the knowledge areas for each of the chosen three categories. The student chose the knowledge area *User Experience Design* for *Users and Organizations* category, and *Systems Analysis and Design* and *Requirements Analysis and Specification* for *Systems Modeling* category; again the ellipse of the chosen knowledge areas are



|  | Dispositions |
|---|---|
|  | Proactive |
|  | Self-directed |
|  | Passionate |
|  | Purpose-driven |
|  | Professional |
|  | Responsible |
|  | Adaptable |
|  | Collaborative |
|  | Responsive |
|  | Meticulous |

(a) Before choosing

|  | Dispositions |
|---|---|
| ✓ | Proactive |
|  | Self-directed |
| ✓ | Passionate |
|  | Purpose-driven |
|  | Professional |
| ✓ | Responsible |
|  | Adaptable |
|  | Collaborative |
| ✓ | Responsive |
|  | Meticulous |

(b)After choosing

**Fig. 8** Choosing dispositions by a prospective student

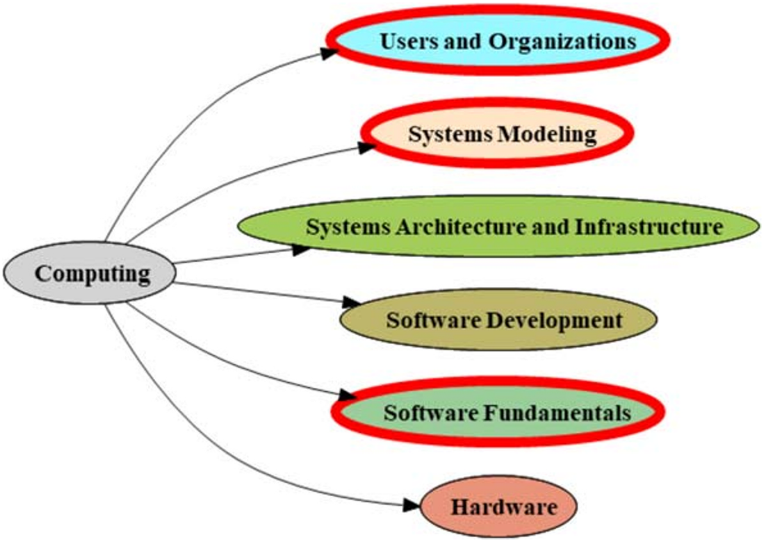
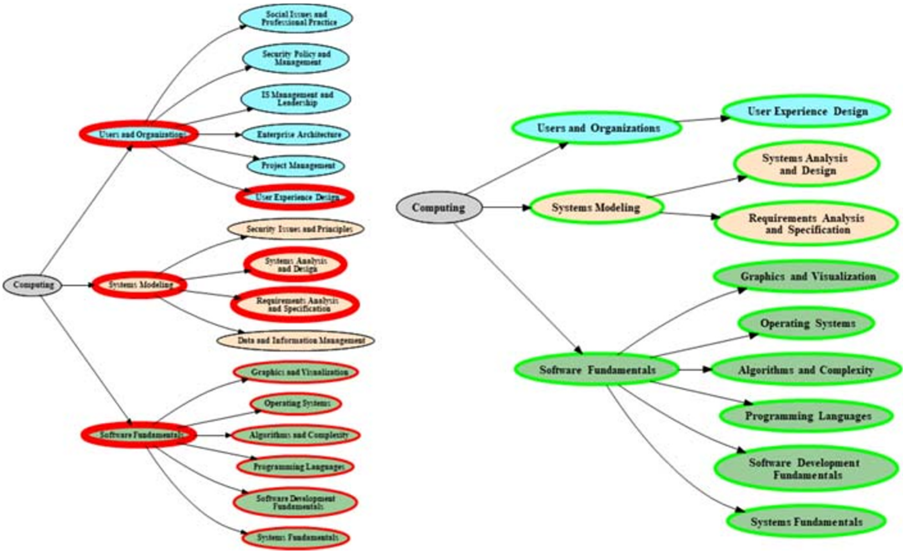


**Fig. 9** The student’s choice of computing categories

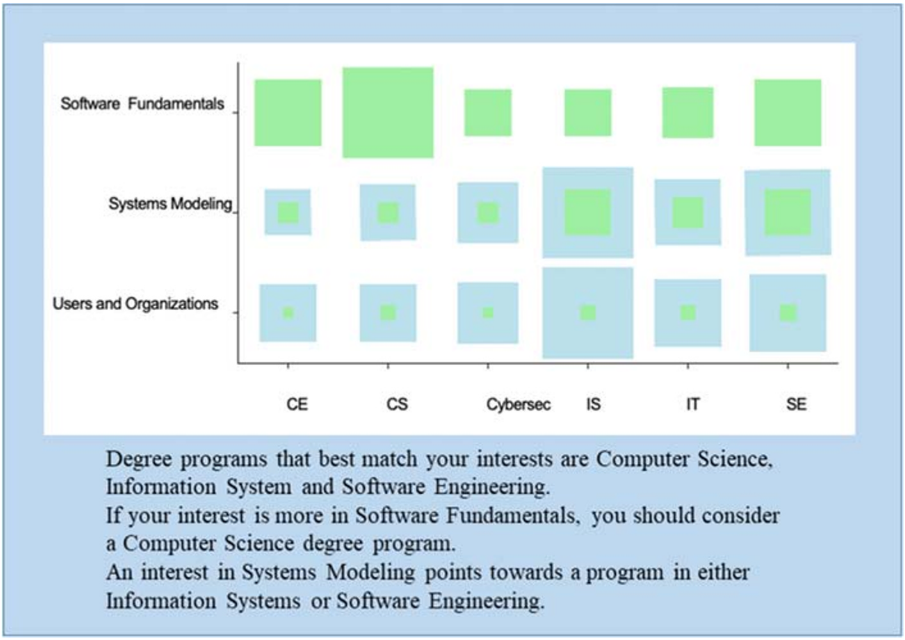
highlighted with red borders. The student did not want to make a detailed choice in the category of *Software Fundamentals*. The resulting choices are shown in Fig. 10(b).

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 Fig. 11, the intended knowledge categories (which have been partly


Social Issues and Professional Practice
Security Policy and Management
IS Management and Leadership
Users and Organizations
Enterprise Architecture
Project Management
User Experience Design
Security Issues and Principles
Systems Analysis and Design
Computing
Systems Modeling
Requirements Analysis and Specification
Data and Information Management
Graphics and Visualization
Operating Systems
Software Fundamentals
Algorithms and Complexity
Programming Languages
Software Development Fundamentals
Systems Fundamentals
Users and Organizations
User Experience Design
Systems Analysis and Design
Computing
Systems Modeling
Requirements Analysis and Specification
Graphics and Visualization
Operating Systems
Software Fundamentals
Algorithms and Complexity
Programming Languages
Software Development Fundamentals
Systems Fundamentals

(a) Choosing knowledge areas (b) Final result

**Fig. 10** Detailed choice of knowledge areas



Degree programs that best match your interests are Computer Science, Information System and Software Engineering.
If your interest is more in Software Fundamentals, you should consider a Computer Science degree program.
An interest in Systems Modeling points towards a program in either Information Systems or Software Engineering.

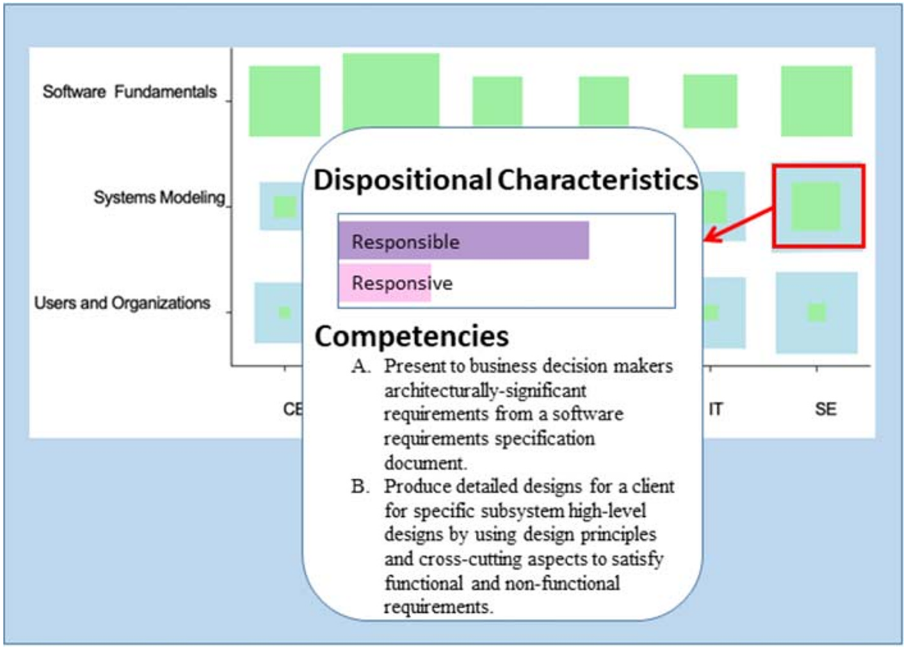
**Fig. 11** Mapping of chosen knowledge categories to the six curricular guidelines

specified into knowledge areas) are mapped for each of the six 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 Table 2. Since the student is more interested in software modeling, based on the message given in Fig. 11, the student decides to explore details regarding SE and her favored knowledge categories. By hovering over a square (Fig. 12), the corresponding competencies are listed. Also displayed are the dispositions linked to the competencies along with the relative level computed from the student choices.

### 5.2 Case 2: A 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 Figs. 9 and 10 in Case 1, she decides to choose *Hardware*, *Software Fundamentals*, and *Software Development* as categories that seem relevant, and removes the other three categories.



Dispositional Characteristics

Responsible

Responsive

Competencies

A. Present to business decision makers architecturally-significant requirements from a software requirements specification document.

B. 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.

**Fig. 12** Disposition and competency details

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 Fig. 13.

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 Fig. 14, the skill level is specified by using a slider, and the disposition is specified by choosing from a menu.

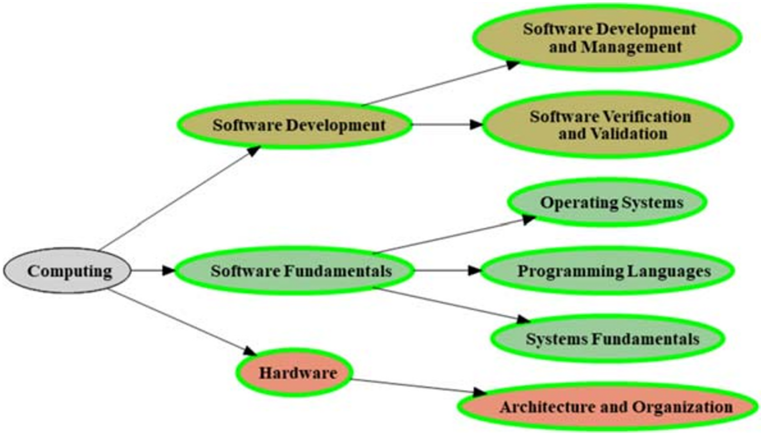


**Fig. 13** Result of knowledge areas selection

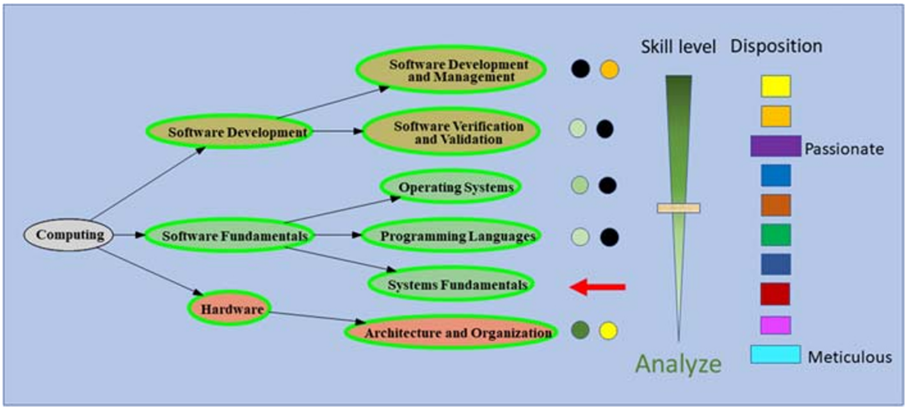


**Fig. 14** Detailing skill and disposition

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 15 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.

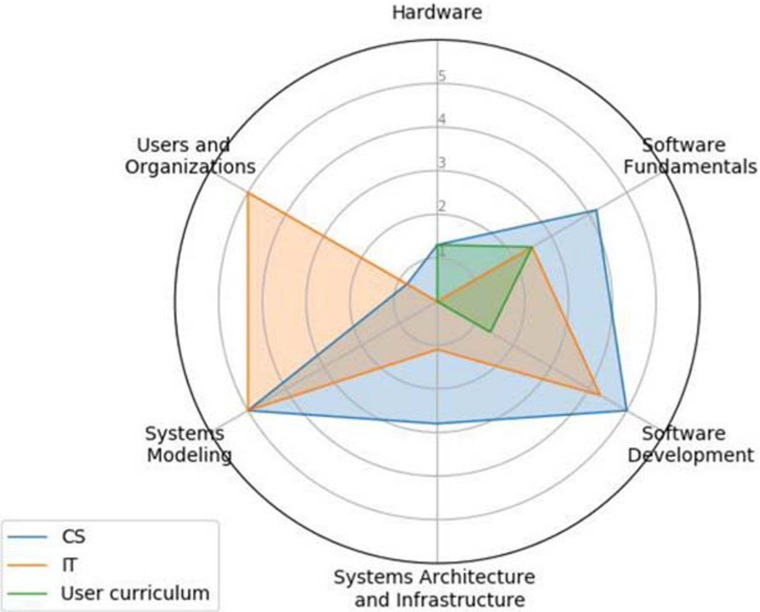


**Fig. 15** Comparison of CS and IT based on knowledge level

## 6 Conclusion

We described work within the CC2020 project concerned with visualization of curricular guidelines. We focused on two approaches for building the data representations of the curricular guidelines; one based on expert-defined competencies and one based on expert-defined knowledge areas. We also showed two examples of how users can explore the various curricular guidelines through visualization. The use of visualization requires more work within the CC2020 project and beyond.

With respect to the first research question: “Is it possible to generate competency-based learners from a knowledge-based setting?” the result is affirmative, although qualified. Based on the results of the first experiment, we saw that there was a significant association for knowledge and skills. The disposition component of competency seemed confusing to the respondents. Hence the association was not as strong as hoped.

With respect to the second research question: “What would be a spectral decomposition of computing knowledge at the undergraduate level?” the result is affirmative. There was no problem with this aspect of the study. Respondents were able to contrast the elements of computing knowledge elements and using min-max levels, they were able to identify the effect according to each of the six computing disciplines.

With respect to the third research question: “Can we describe one or more visualization tools useful to a variety of users that can clarify different fields of computing from a competency-based viewpoint?” the result is affirmative. Based on the results of the two pilot experiments, there were enough data to formulate preliminary diagrams based on competency that included knowledge, skills, and dispositions.

In summary, this research activity resulted in a positive experience to demonstrate the visualizations of computing curricula. The research showed that it was possible to contrast one curriculum to another. It also provided a mechanism by which stakeholders (students, practitioners, academics) might use visual tools to explore a variety of possible applications such as deciding which area of computing to study, the expectation of industry practitioners to hire computing graduates, or for academics to modify current curricula or develop new curricula for the rapidly changing field of computing.

## 7 Future work

The authors envision future work requiring three important activities. The first activity concerns the addition of further possible visualizations. Section 5 discussed several visualizations developed by the authors. These visualizations are not a complete and final set. The authors and others plan to consider other possible visualizations such as dynamic heat maps and three-dimensional figures.

The second activity concerns a stable and accurate representation of curricular guidelines as this is the basis for any and all visualizations. With advances in technology and new disciplines appearing, it is unlikely to attain a completely stable representation. Indeed, during the writing of this paper, an initial draft for a computing companion to data science, data engineering, and data analytics have appeared (ACM 2019). Thus, it is important to find an ongoing way to improve and update the representation. A sub-group of the authors are currently conducting an exploratory effort of mining existing descriptive language of published guidelines through screen-scraping and machine learning to extract a vocabulary appropriate for the dimensions that can facilitate visualization through direct computation. We also believe that this mining-based approach is promising to address the issue of improving and updating the representation.

The culmination of the above leads to the third activity: the creation of a tool made available to the public. Such a tool is currently under development. The steering committee intends to have it available through the CC2020 website by the end of 2020. Once achieved, the website would enable various stakeholders to answer questions such as those posed in section 2.3 of this paper.

**Acknowledgements** The authors would like to thank the following organizations for supporting this project: ACM, IEEE-CS, AIS, the Association for Information Technology Professionals Education Special Interest Group (AITP/EDSIG), the ACM Special Interest Group for Computer Human Interaction (SIGCHI), the Information Processing Society of Japan (IPSJ), the Chinese Computing Federation (CCF), the Latin American Computing Conference (Conferencia Latinoamericana de Informática - CLEI), ACM India, and NSPARC (Mississippi State University).

## Appendix A

### List of example questions from the various stakeholders of the interactive visualization tool.

**Prospective students,** or their parents, are considering a curriculum in computing (currently one of CE, CS, IS, IT, SE, and Cybersecurity). Example questions are as follows:

- Does this program of study (or what program or courses) make me fit for specific requirements as stated in an industry’s job descriptions?
- Does this program of study fit my interests?
- What background would I need in mathematics to be successful in this curriculum?
- Are graduates of this program expected to work primarily as individuals (e.g. doing coding) or work with other people?
- What kinds of contributions do professionals in this curriculum make to society?
- What competencies does this program (course) provide?
- For which jobs (job titles) in the local area would this program (courses) satisfy competency requirements?

**Current students** are students that are enrolled at a university level educational institute. Example questions are as follows:

- On which course(s) does the curriculum place strongest emphasis (high level, or long duration, compared to the guidelines)?
- I’m interested in computing, but not sure what I actually want to do (work) after graduation. Which curriculum (or does this curriculum) seem to provide a broad preparation in computing skills and knowledge?
- Are there differences between universities that offer the same curriculum?
- Does a selection from a study, or a specific course, bring me up to speed regarding my current (professional) needs?

**Industry** indicates industry bodies that are hiring certified students, are collaborating with universities to choose or specialize a curriculum or need a tailor made-course, or that are collaborating in a curriculum by providing internships. Industry, in general, has rather clear ideas about what knowledge at what level is needed, as well as what human dispositions are required to apply the knowledge.

Industry mainly asks only one type of question. It primarily focuses on whether specific competencies for knowledge areas in courses offered by a certain university in curriculum XYZ are appropriate for recruiting new employees, or for continued professional education for actual employees.

**Educators**' questions usually address a curriculum or a course for which they are responsible, which might either be intended to be equivalent to one of our current set (CE, CS, Cybersecurity, IS, IT, or SE), a combination of some of these, or a hybrid that contains parts of non-computing content. Following are examples:

- What type of courses; what type of end competences, should be in my curriculum?
- What is suggested for my course on knowledge area(s) X or X&Y; could I adopt an existing course from elsewhere to fill a gap / provide an alternative / in my curriculum?
- Could I accept students from another educational institute to take my course; to embark at level X in my curriculum?
- Is there a particular area where my curriculum is different from the guidelines? Where I might consider changes?
- Is there a particular area where my curriculum seems to contain unique skill content, and do I want this?
- When accepting students from other universities/curriculum, which courses can they skip, i.e., which courses can be considered as "already taken"?
- What types of applied methods should be included in my curriculum?
- What is the minimum number of course hours in my curriculum to satisfy educating for this particular specification of competency(s)?
- Which competencies in program X are not satisfied by program Y?

**Educational authorities.** Education authorities (and educators) that want to have their curriculum comparable to the IEEE-ACM-AIS guidelines are suggested to add labels to their curriculum content that reflect the knowledge areas. In practice, this might well be an indication of which courses relate to one knowledge area, to a combination of several, or be a hybrid of a knowledge area and another content label. and with the knowledge area labels there might be competence descriptions indicating knowledge level and/or dispositions. For example, a course or curriculum part named "Multimedia" might well be indicated to have a partial content of Graphics, User Experience Design, and Visual Art (not part of our computing curricula).

Education authority questions (Most of these questions would appear to provide useful background data that would be used in a broader analysis effort to reach a conclusion. Therefore, these questions would not be directly answered but, would provide input to the answer.):

- Does this curriculum comply to the guidelines for curriculum X? What should be changed and how?

- Could we accept students from a specified curriculum X to finish in curriculum Y?
- I want to rank universities/departments based on curriculum. Do universities that say they teach X actually teach Y?
- What subset (if any) of program Y is satisfied by competency found in program X?
- Is any coursework in program X superfluous to satisfying program Y?
- Is there coursework in programs X and Y that is redundant?

## Appendix B

### Category Definitions for Competency (Knowledge, Skills, and Dispositions) used in Pilot.

**A. Level of Knowledge**

1. *Factual* – basic elements students must know to be acquainted with a discipline to solve problems in it (simple professional facts or separate concepts)
2. *Conceptual* – interrelationships among basic elements within a larger structure that enable them to function together (semantics of professional knowledge)
3. *Procedural* - how to do something, methods of inquiry, and criteria for using skills, algorithms, techniques and methods (how to use professional techniques and tools)
4. *Metacognitive* – knowledge of cognition in general as well as awareness and knowledge of one’s own cognition (how to learn and where to find professional knowledge)

**B. Skills in a professional type of activities**

1. Analysis of a problem or question
2. Modeling the design /problem space
3. Design of a potential solution
4. Implementation of the solution
5. Assessment of the solution
6. Deployment of the solution in the intended actual content
7. Project planning and management

**C. Dispositions**

1. *Collaborative Attitude* – Having the character to work jointly with others or together especially in an intellectual endeavor
2. *Communication Clarity* – Having oral and written ability to process and to exchange information between individuals lucidly through common conduits of writing or oral presentation
3. *Creative Thinking* – Possessing the mental quality of producing by a course of action or behavior something rather than imitating it
4. *Institutional Temperament* – Possessing a personal mental attitude, mood, or behavior within industry or government in harmony with a common goal
5. *Leadership Behavior* – Having the human capacity to guide someone or something along a path toward a common goal
6. *Mentoring Colleagues* – Possessing the human ability to become a person charged with the instruction and guidance of another in the workplace
7. *Organizational Aptitude* – Having the personal skills to coordinate and carry out activities efficiently
8. *Relationship Development* – Having the personality to develop a human state between those being mutually or reciprocally interested dealing with social or commercial matters
9. *Self-Motivation* – Having the personality to act or influence oneself with a reason for achieving some goal or ambition
10. *Time Management* – Possessing the capacity to control and make decisions about using one’s time efficiently

## Appendix C

**List of IT 2017 essential competencies used in the pilot survey.**

**Information Management** (ACM & IEEE-CS 2017a, p.56)

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.

B. Design and implement a physical model based on appropriate organization rules for a given scenario including the impact of normalization and indexes.

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.

D. Analyze ways data fragmentation, replication, and allocation affect database performance in an enterprise environment.

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.

**Integrated Systems Technology** (ACM & IEEE-CS 2017a, p.56)

F. 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.

G. Show how a commonly used intersystem communication protocol works, including its advantages and disadvantages.

H. Design, debug and test a script that includes selection, repetition and parameter passing.

I. 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.

**System Paradigms** (ACM & IEEE-CS 2017a, p.58)

J. Justify the way IT systems within an organization can represent stakeholders using different architectures and the ways these architectures relate to a system lifecycle.

K. 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.

L. Evaluate integration choices for middleware platforms and demonstrate how these choices affect testing and evaluation within the development of an IT system.

M. 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.
N. Develop and implement procedures and employ technologies to achieve administrative policies within a corporate environment.
O. Organize personnel and information technology resources into appropriate administrative domains in a technical center.
P. Use appropriate and emerging technologies to improve performance of systems and discover the cause of performance problems in a system.

**Global Professional Practice** (ACM & IEEE-CS 2017a, p.55)
Q. Analyze the importance of communication skills in a team environment and determine how these skills contribute to the optimization of organization goals.
R. Evaluate the specific skills necessary for maintaining continued employment in an IT career that involves system development in an environmental context.
S. Develop IT policies within an organization that include privacy, legal, and ethical considerations as they relate to a corporate setting.
T. 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.

**Cybersecurity Principles** (ACM & IEEE-CS 2017a, p.55)
U. 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.
V. Implement systems, apply tools, and use concepts to minimize the risk to an organization's cyberspace to address cybersecurity threats.
W. 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.
X. 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.

**User Experience Design** (ACM & IEEE-CS 2017a, p.59)
Y. 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.

Z. 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 techniques (e.g., heuristic evaluation), aiming at universal access and inclusiveness, and showing a responsive design attitude, considering assistive technologies and culture sensitive design.

AA. For evaluation of user-centered design, articulate evaluation criteria and compliance to relevant standards

BB. In design and analysis, apply knowledge from related disciplines including human information processing, anthropology and ethnography, and ergonomics/human factors.

CC. Apply experience design for a service domain related to several disciplines, focusing on multiple stakeholders and collaborating in an interdisciplinary design team.

**Networking** (ACM & IEEE-CS 2017a, p.57)

DD. Analyze and compare the characteristics of various communication protocols and how they support application requirements within a telecommunication system.

EE. Analyze and compare several networking topologies in terms of robustness, expandability, and throughput used within a cloud enterprise.

FF. Describe different network standards, components, and requirements of network protocols within a distributed computing setting.

GG. Produce managerial policies to address server breakdown issues within a banking system.

HH. Explain different main issues related to network management.

**Software Fundamentals** (ACM & IEEE-CS 2017a, p.58)

II. Use multiple levels of abstraction and select appropriate data structures to create a new program that is socially relevant and requires teamwork.

JJ. 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.

KK. Develop algorithms to solve a computational problem and explain how programs implement algorithms in terms of instruction processing, program execution, and running processes.

LL. 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.

**Web and Mobile Systems** (ACM & IEEE-CS 2017a, p.59)

MM. Design a responsive web application utilizing a web framework and presentation technologies in support of a diverse online community.

NN. Develop a mobile app that is usable, efficient, and secure on more than one device.

OO. Analyze a web or mobile system and correct security vulnerabilities.

PP. Implement storage, transfer, and retrieval of digital media in a web application with appropriate file, database, or streaming formats.

QQ. Describe the major components of a web system and how they function together, including the web server, database, analytics, and front end.

**Platform Technologies** (ACM & IEEE-CS 2017a, p.57)

RR. Describe how the historical development of hardware and operating system computing platforms produced the computing systems we have today.

SS. Show how to choose among operating system options, and install at least an operating system on a computer device.

TT. 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.

UU. 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.

## Appendix D

**Pilot assessment results.**

### D. 1 Competency and category where 80% or more of the respondents chose the same category for the knowledge area.

Domain and competency item are given according to the listing given in Appendix B. The ratio “X/Y” denotes that X out of Y respondents chose that category.

| Domain | Competency | Category | Ratio |
|---|---|---|---|
| System Paradigm | J | Conceptual | 4/5 |
| Cybersecurity | V | Procedural | 5/6 |
| User Experience Design | AA | Conceptual | 4/5 |
| Web & Mobile Systems | MM | Procedural | 5/6 |
| | PP | Procedural | 5/6 |
| Platform Technologies | RR | Factual | 5/6 |
| | SS | Procedural | 6/6 |

### D. 2 Competency and category where 50% or more of the respondents chose the same category for the knowledge area.

The following lists those other than the ones that were given in Appendix D.1.

| Domain | Competency | Category | Ratio |
|---|---|---|---|
| Information Management | A | Conceptual | 3/5 |
| | B | Procedural | 3/5 |
| | C | Procedural | 3/5 |
| | D | Conceptual | 3/5 |
| Integrated Systems Tech | F | Factual | 3/6 |
| | H | Procedural | 4/6 |
| | I | Conceptual | 4/6 |
| System Paradigms | L | Procedural | 3/5 |
| | N | Procedural | 3/5 |
| | O | Procedural | 3/5 |
| | P | Procedural | 3/5 |
| Global Professional Prac. | S | Procedural | 5/7 |
| | T | Procedural | 3/6 |
| Cybersecurity | W | Procedural | 3/5 |
| | X | Procedural | 3/5 |
| User Experience Design | CC | Conceptual | 3/5 |
| Networking | DD | Conceptual | 6/11 |
| | EE | Conceptual | 8/11 |
| | FF | Factual | 6/11 |
| | HH | Conceptual | 6/11 |
| Software Fundamentals | II | Procedural | 3/6 |
| | JJ | Conceptual | 3/6 |
| | KK | Procedural | 4/6 |
| | LL | Procedural | 4/6 |
| Web & Mobile Systems | NN | Procedural | 3/6 |
| | OO | Procedural | 4/6 |
| | QQ | Conceptual | 3/6 |
| Platform Technologies | UU | Factual | 3/6 |

### D. 3 Competency and category where the top two responses totaled 50% or more for the skill area.

| Domain | Competency | Category #1 (Num. Responses) | Category #2 (Num. Responses) | Total Num. Responses |
|---|---|---|---|---|
| Information Management | A | Analysis (2) | Assessment (3) | 8 |
| | B | Design (4) | Implementation (4) | 13 |
| | C | Design (4) | Implementation (3) | 8 |
| | D | Analysis (4) | Assessment (3) | 10 |
| | E | Proj. Plan. & Man.(4) | Implementation (3) | 11 |
| Integrated Systems Tech | F | Implementation (4) | Assessment (2) | 12 |
| | G | Implementation (3) | Assessment (3) | 11 |
| | H | Design (4) | Implementation (4) | 14 |
| | I | Modelling (3) | Implementation (3) | 12 |
| System Paradigms | J | Analysis (4) | Assessment (3) | 12 |
| | L | Assessment (4) | Implementation (3) | 14 |
| | N | Implementation (4) | Design (3) | 12 |
| | O | Deployment (3) | Proj. Plan. & Man. (3) | 9 |
| | P | Assessment (4) | Implementation (3) | 11 |
| Global Professional Prac. | Q | Analysis (4) | Proj. Plan. & Man. (4) | 12 |
| | R | Assessment (3) | Proj. Plan. & Man. (3) | 11 |
| Cybersecurity | U | Analysis (5) | Analysis (5) | 12 |
| | V | Implementation (5) | Deployment (2) | 9 |
| | X | Implementation (2) | Proj. Plan. & Man. (3) | 11 |
| User Experience Design | Y | Design (4) | Implementation (2) | 9 |
| | Z | Analysis (3) | Modelling (3) | 11 |
| | AA | Assessment (4) | Implementation (2) | 9 |
| | BB | Design (3) | Implementation (3) | 10 |
| Networking | DD | Analysis (9) | Assessment (4) | 19 |
| | EE | Analysis (8) | Assessment (5) | 23 |
| | FF | Analysis (6) | Modelling (3) | 17 |
| Software Fundamentals | JJ | Design (5) | Modelling (2) | 14 |
| | KK | Design (5) | Modelling (4) | 15 |
| Web & Mobile Systems | OO | Analysis (5) | Implementation (3) | 14 |
| | PP | Implementation (5) | Design (4) | 13 |
| | QQ | Analysis (4) | Modelling (3) | 10 |
| Platform Technologies | RR | Analysis (5) | Modelling (2) | 10 |
| | SS | Implementation (4) | Deployment (3) | 10 |
| | TT | Analysis (3) | Modelling (2) | 10 |
| | UU | Modelling (3) | Design (2) | 9 |

# Appendix E

## Questions from Pilot addressing the various category choices.

| Summary Questions | Yes/Mostly Yes # | Total # |
|---|---|---|
| Do you find the category choices for "knowledge" areas provided to be intuitively appropriate? | 10 | 12 |
| Do you find the category choices for knowledge provided to be distinctive? | 12 | 13 |
| Do you think the category choices would be equally useful across "knowledge" aspects of other computing disciplines? | 10 | 12 |
| Do you find the category choices for "skills" areas provided to be intuitively appropriate? | 11 | 12 |
| Do you find the category choices for skills provided to be distinctive? | 9 | 12 |
| Do you think the category choices would be equally useful across "skills" aspects of other computing disciplines? | 10 | 12 |
| With respect to the "dispositions" associated with the major IT categories, do you find the choices provided to be appropriate? | 11 | 12 |
| With respect to the "dispositions" associated with the major IT categories, Do you find the choices provided to be complete? | 10 | 12 |
| Do you believe the categorization of competencies into "knowledge", "skills", and "dispositions" a useful method for summarizing and ultimately comparing the requirements of different computing disciplines? | 9 | 12 |

## Appendix F

### Description of dispositions in Section 5.

| Dispositions | Description |
|---|---|
| Proactive | 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. (With Initiative (Nwokeji, Stachel, & Holmes, 2019) / Self-Starter (Clear, 2017)) |
| Self-directed | Demonstrates determination to sustain efforts to continue tasks. Direction from others is not required to continue a task toward its desired ends. (Self-motivated (Clear, 2017) / Self-Directed (Nwokeji et al., 2019)) |
| Passionate | 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. (With Passion (Nwokeji et al., 2019), (Clear, 2017) / Conviction (Gray, 2015)) |
| Purpose-driven | 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. (Purposefully engaged / Purposefulness (Nwokeji et al., 2019), (Clear, 2017)) |
| Professional | Reflecting qualities connected with trained and skilled people: Acting honestly, with integrity, commitment, determination and dedication to what is required to achieve a task. (With Professionalism / Work ethic (Nwokeji et al., 2019)) |
| Responsible | 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. (With Judgement / Discretion (Nwokeji et al., 2019) / Responsible (Clear, 2017) / Rectitude (Gray, 2015)) |
| Adaptable | Ability or willingness to adjust approach in response to changing conditions or needs. (Adaptable (Nwokeji et al., 2019) / Flexible (Clear, 2017) / Agile (Weber, 2017)) |
| Collaborative | 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. (Collaborative (Weber, 2017) / Team Player (Clear, 2017) / Influencing (Nwokeji et al., 2019)) |
| Responsive | Reacting quickly and positively. Respecting the timing needs for communication and actions needed to achieve the goals of the work. (Responsive (Weber, 2017) / Respectful (Clear, 2017)) |
| Meticulous | Achieves thoroughness and accuracy when accomplishing a task through concern for relevant details. (Attentive to Detail (Weber, 2017), (Nwokeji et al., 2019)) |

## References

ACM. (2019). *Computing Competencies for Undergraduate Data Science Curricula (Initial Draft)*. http://www.cs.williams.edu/~andrea/DSReportInitialFull.pdf.

ACM & AIS. (2010). *IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems*. https://www.acm.org/binaries/content/assets/education/curricula-recommendations/is-2010-acm-final.pdf.

ACM & AIS. (2017). *MSIS 2016: Global Competency Model for Graduate Degree Programs in Information Systems*. https://www.acm.org/binaries/content/assets/education/msis2016.pdf.

ACM & IEEE-CS. (2001). *Computing Curricula 2001: Computer Science*. https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2001.pdf.

ACM & IEEE-CS. (2013). *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf.

ACM & IEEE-CS. (2015). *Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. https://www.acm.org/binaries/content/assets/education/se2014.pdf.

ACM & IEEE-CS. (2016). *Computer Engineering Curricula 2016: Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering*. https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf.

ACM & IEEE-CS. (2017a). *Information Technology Curricula 2017: Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology*. https://www.acm.org/binaries/content/assets/education/curricula-recommendations/it2017.pdf.

ACM & IEEE-CS. (2017b). *Cybersecurity Curricula 2017: Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity*. https://www.acm.org/binaries/content/assets/education/curricula-recommendations/csec2017.pdf.

ACM, AIS, & IEEE-CS. (2005). *Computing Curricula 2005: The Overview Report covering undergraduate degree programs in Computer Engineering, Computer Science, Information Systems, Information Technology, and Software Engineering (CC2005)*. https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2005-march06final.pdf.

Armstrong, C. J. (2013). An approach to visualising information security knowledge. In *WISE 6, 7, and 8. IFIP AICT, 406*, 148–155.

Bloom, B. S., & Krathwohl, D. R. (1956). Taxonomy of educational objectives: The classification of educational goals, by a committee of college and university examiners. In *Handbook I: Cognitive domain*. New York: Longman.

Clear, T. (2015). IT industry employers expectations: our graduates deserve better!. In ACM Inroads 6(2):20-22.

Clear, T. (2017). THINKING ISSUES: Meeting employers expectations of devops roles: Can dispositions be taught?. In ACM Inroads. 8(2):19–21.

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.

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.

Dave, R.H. (1970). *Psychomotor levels in Developing and Writing Behavioral Objectives*, pp.20-21. R.J. Armstrong, ed. Tucson, Arizona: Educational innovators press.

Frezza, S., Daniels, M., Pears, A., Cajander, A., Kann, V., Kapoor, A., McDermott, R., Peters, A., Sabin, M., & Wallace, C. Modelling competencies for computing education beyond 2020: a research based approach to defining competencies in the computing disciplines. In *Proc. Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE* 2018 Companion). ACM. pp.148-174.

Gray, J. (2015). Virtue Ethics: Examining Influences on the Ethical Commitment of Information System Workers in Trusted Positions. Doctoral dissertation. Nova Southeastern University. https://nsuworks.nova.edu/gscis_etd/364/

Heath, H. (1998). "Reflection and patterns of knowing in nursing," *Journal of Advanced Nursing*, Vol. 27, pp. 1054-1059.

Harrow, A. (1972). *A taxonomy of psychomotor domain: A guide for developing behavioral objectives*. New York, USA: David McKay Co., Inc..

IEEE. (2014). *Software Engineering Competency Model: Version 1.0*. IEEE.

Impagliazzo, J., et al. (2018a). Developing an overview of computing/Engineering curricula via the CC2020 project. In *Proc. of the IEEE EduNine Conference*. IEEE Education Society.

Impagliazzo, J., et al. (2018b). Developing an overview of computing/Engineering curricula via the CC2020 project. In *Proc. of the IEEE EduNine Conference*. IEEE Education Society.

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*.

Johns, C. (1995). "Framing learning through reflection within Carper's fundamental ways of knowing in nursing," *Journal of Advanced Nursing*, Vol. 22, pp. 226-234.

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.

Marshall, L. (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. In *Computer Science Education Research Conference (CSERC 2012)*. ACM. Pp.29-34.

Marshall, L. (2014). *A graph-based framework for comparing curricula*. Ph.D. thesis, University of Pretoria, South Africa.

Marshall, L. (2017). A topic-level comparison of the ACM/IEEE CS curriculum volumes. *In Communications in Computer and Information Science. Springer., 730*, 309–324.

Nwokeji, J., 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)*. 1–9.

Perkins, D. N., Jay, E., & Tishman, S. (1993). Beyond abilities: A dispositional theory of thinking. *Merrill-Palmer Quarterly., 39*(1), 1–21.

Polanyi, M. (1966). *The tacit dimension*. University of Chicago Press.

Radermacher, A., Walia, G., and Knudson, D. (2014). Investigating the skill gap between graduating students and industry expectations. In *Companion Proc. of the 36th International Conference on Software Engineering (ICSE Companion 2014)*. Pp.291-300.

Siirtola, H., Räihä, K., & Surakka, V. (2013). Interactive Curriculum Visualization. In *In 2013 17th International Conference on Information Visualisation* (pp. 108–117).

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.

Stamper, R. K. (1991) The semiotic framework for information systems research. Information systems research. *Contemporary approaches and emergent traditions*, pp. 515-528.

Topi, H. (2017) Information systems in CC2020: Comparing key structural elements of curriculum recommendations in computing. In *Proc. of AIS SIGED 2017 Conference*.

von Konsky, B. R., Hay, D., & Hart, B. (2008). Skill set visualisation for software engineering job positions at varying levels of autonomy and responsibility. In *19th Australian Conference on Sofware Engineering (ASWEC) Industry Experience Reports*.

von Konsky, B. R., Jones, A., & Miller, C. (2013). Embedding professional skills in the ICT curriculum. In *30th ascilite Conference*. Pp. 883-887.

Waguespack, L., Babb, J. (2019). Toward visualizing computing curricula: The challenge of Competency. In *Information Systems Education Journal*, 17(4) pp 51-69. http://isedj.org/2019-17/ ISSN: 1545-679X. (a preliminary version appears in *Proceedings of EDSIGCON 2018* <http://proc.iscap.info/2018/pdf/4633.pdf>).

Weber, H. (2017). The New Virtues of Engineering and the Need for Change in the Engineering Curriculum. https://www.researchgate.net/publication/325924314_The_New_Virtues_of_Engineering_and_the_Need_for_Change_in_the_Engineering_Curriculum Accessed August 21, 2019.

Whalley, J.L., Lister, R., Thompson, E., Clear, T., Robbins, P., Kumar, P.K.A, & Prasad, C. (2006). An Australasian study of reading and comprehension skills in novice programmers, using the bloom and SOLO taxonomies. In *Proc. of 8th Australasian Conference on Computing Education (ACE '06)*, pp.243-252.

Wiggins, G., McTighe, J., & Ebrary, I. (2005). *Understanding by design Expanded* (2nd ed.). Alexandria, VA: Association for Supervision and Curriculum Development.

Willcox, K., & Huang, L. (2017). Mapping the CDIO curriculum with network models. CDIO. In *13th International CDIO Conference*.

Zucker, R. (2009). ViCurriAS: A curriculum visualization tool for faculty, advisors, and students. *In J. Comput. Sci. Coll., 25*(2), 138–145.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

**Shingo Takada[1] · Ernesto Cuadros-Vargas[2] · John Impagliazzo[3] · Steven Gordon[4] · Linda Marshall[5] · Heikki Topi[6] · Gerrit van der Veer[7] · Leslie Waguespack[6]**

Ernesto Cuadros-Vargas
ecuadros@utec.edu.pe

John Impagliazzo
john.impagliazzo@hofstra.edu

Steven Gordon
Gordon.1@osu.edu

Linda Marshall
lmarshall@cs.up.ac.za

Heikki Topi
htopi@bentley.edu

Gerrit van der Veer
gerrit@acm.org

Leslie Waguespack
LWaguespack@bentley.edu

[1] Keio University, Yokohama 223-8522, Japan

[2] Universidad de Ingeniería y Tecnología, 15063 Lima, Peru

[3] Emeritus, Hofstra University, Hempstead, NY 11549, USA

[4] Emeritus, Ohio State University, Columbus, OH 43210, USA

[5] University of Pretoria, Pretoria 0002, South Africa

[6] Bentley University, Waltham, MA 02452, USA

[7] Vrije Universiteit Amsterdam, Amsterdam, Netherlands