Association for Information Systems
AIS Electronic Library (AISeL)

2017 Proceedings | SIGED: IAIM Conference

2017

# Information Systems in CC2020: Comparing Key Structural Elements of Curriculum Recommendations in Computing

Heikki Topi
*Bentley University*, htopi@bentley.edu

# INFORMATION SYSTEMS IN CC2020: COMPARING KEY STRUCTURAL ELEMENTS OF CURRICULUM RECOMMENDATIONS IN COMPUTING

Heikki Topi
Department of Computer Information Systems
Bentley University
htopi@bentley.edu

**Abstract:**

This paper describes the characteristics of the Computing Curricula 2020 process, discusses the reasons why it is essential for information systems to be involved, and explores the core structures of existing computing curriculum recommendations, particularly from the learning outcome and competency perspective. The two main categories are the knowledge area – knowledge unit structure used by CE, CS, and SE and the competency structure used by IT and MSIS. Finding a way to express the competency expectations of all degree program types in computing at the same level of abstraction will be a key to the success of the CC2020 project. The upcoming process to develop a new IS undergraduate recommendation will also benefit from CC2020 work and contribute to it.

Keywords: Computing curricula, learning outcomes, competencies

## I. INTRODUCTION

In mid-2000s, ACM, IEEE-CS, and AIS collectively published a report entitled *Computing Curricula 2005: Overview report* (later CC2005) [Shackelford et al., 2005]. It covered undergraduate degree programs in Computer Engineering (CE), Computer Science (CS), Information Systems (IS), Information Technology (IT), and Software Engineering (SE). CC2005 was specifically intended to describe the identity of various undergraduate degree types in computing and analyze the similarities and differences between the degrees. In addition, the task force specifically stated that the report wants to “help you determine which of the programs are most suited to particular goals and circumstances” (Shackelford et al., 2005). In addition, CC2005 was also intended to provide background material for guidance given to students regarding potential career paths best supported by specific computing degree types.

CC2005 was a successful initiative, and its main outcome, the Overview Report, has been cited more than 250 times according to Google Scholar. At least as important is the report’s use by numerous universities, colleges, schools, and departments to make sense of the complex landscape of computing and computing education. Specific data regarding the extent of this use is difficult to find but anecdotal evidence suggests that particularly the figures and tables of the document have served well as background in many internal discussions.

CC2005 is, however, more than 10 years old and, therefore, ACM and IEEE-CS together with collaborating partners AIS and AITP EDSIG have decided to launch a project to create a set of new deliverables that will not only serve updated in roles similar to those of CC2005 but also provide a much deeper look into the future of computing education. CC2020 has also been a global project from the very beginning, representing a major change from the North American focus of CC2005.

The purpose of this paper is to given an overview of the CC2020 project from information systems perspective and to focus on a significant area of intended new focus of the CC2020 project: graduate competencies. Even though its well-known graphical views of computing disciplines focus on “what students in each of the disciplines typically do after graduation, not on all topics a student might study,” at the detailed level, CC2005 compared computing degree programs primarily based on the topics that these programs cover in the model curricula (Tables 3.1 and 3.2). CC2020 will focus significantly more on the expected competencies of the program graduates. To build foundation for this work, this paper will review the ways in which different computing curriculum recommendations have dealt with various outcomes of the degree program (graduate competencies and program or course learning outcomes) vs. the extent to which various topics are covered in the program. This comparison will analyze not only the five traditional undergraduate degree program curriculum recommendations (CE2016, CS2013, IS2010, IT2017, and SE2014), but also MSIS2016 (master's level recommendation for programs in information systems)—the only graduate level recommendation jointly developed by two academic/professional societies (ACM and AIS).

In the next section, this paper will first have a closer look at the reasons why it is important for information systems as a discipline to be part of the global conversation regarding computing degree programs and the ways they are related to each other. Next section will provide a review of the types of degree outcomes featured in the computing curriculum documents. Following that, the degree outcome type framework will be used to compare the approaches chosen by various computing degree recommendations (including the IS recommendations IS2010 and MSIS2016). The paper will end with a discussion that explores the broader impact of competency-related plans on CC2020 and the lessons future revisions of the undergraduate level IS curriculum may learn regarding outcomes from the CC2020 work and from earlier curriculum recommendations.

## II. COMPUTING CURRICULA PROJECTS AND THE ROLE OF INFORMATION SYSTEMS IN THEM

There are a variety of reasons why it is useful for the information systems discipline to participate actively in the Computing Curricula projects and influence their outcomes through contributing actively (see also [Topi, forthcoming]).

First, not having information systems as an active participant in a project that intends to review computing as a whole would leave the analysis with a gaping hole. As CC2005 has already demonstrated, IS brings to computing and particularly computing education distinctive capabilities without the overall picture would be incomplete, leaving a significant deficiency.

Second, these projects give information system important visibility as the computing discipline that is most closely connected to the processes through which organizations transform themselves and their core processes through intelligent use of information technology. CC2005 articulated information systems as a discipline that serves organizational needs (Figure 2.1), focuses on "integrating information technology solutions and business processes to meet the information needs of businesses and other enterprises, enabling them to achieve their objectives in an effective, efficient way" (p. 14), and is "concerned with the relationship between information systems and the organizations that they serve" (p. 19)—thus articulating the unique contributions that IS and IS graduates provide. The CC2020 project gives the information systems discipline an opportunity to describe its distinctive contributions in the current context.

Third, projects such as CC2005 and CC2020 give the information systems discipline a good opportunity to consider its identity in relation to other computing disciplines and its main past and potential future contributions to computing education. CC2020 will create an excellent opportunity for analysis and possible reconciliation of multiple perspectives on information systems, given that both AIS and AITP EDSIG are represented in the core of the project.

Fourth, IS has been part of the computing education landscape since late 1960s [see, e.g., Couger, 1973 for an early review], and the IS community has had an integral role in shaping the way in which computing professionals are interacting with and contributing to modern organizations. It is essential that the information systems community will have a role in these conversations also in the future.

## III. KEY CHARACTERISTICS OF THE CC2020 PROJECT

At this time, the identified guiding principles for the CC2020 project are as follows:

1) "Computing" is an appropriate umbrella term for the broad discipline that CC2020 is interested in, and it covers an area of study and domain of knowledge that is much broader than any of the subdisciplines of computing (such as computer science) alone. CC2020 explicitly uses the term of computing with this broader meaning.

2) As CC2005, CC2020 will provide support for distinguishing different types of computing programs to prospective students and their parents, those responsible for program development and management, those hiring computing graduates, and those making decisions regarding educational policy.
3) CC2020 will not only provide a review of the past based on existing documents but it intends to take a forward-looking view, capturing relevant future trends and interpreting their meaning for computing education.
4) CC2020 is a global project and as such, it will have to use language that is expressed in a way that can effectively be used to cross the chasms between various educational systems.

According to the current plan, the final outcomes of CC2020 will include a narrative document ("report"), a competency model for computing as a whole, and a tool that enables meaningful and efficient comparison between computing disciplines. As discussed above, the main focus of this paper will be on the background for the competency model.

## IV. CLASSIFICATION OF CORE UNITS TYPE TERMINOLOGY IN EDUCATIONAL GUIDANCE DOCUMENTS

The core elements of educational guidance documents can be at the highest level separated into two categories: those that focus on the content of educational experiences (at some level of abstraction) and those that focus on the outcomes of educational experiences. Moving forward towards a unified competency model in CC2020 requires understanding of both.

The approaches where the focus is on content in most cases have *knowledge areas* and *knowledge units* (and sometimes topics) as their core elements. Knowledge areas (KA) are typically highest level elements, each of which specifies a broad category ("topical area of study" according CS2013, p. 14) that brings together a number of knowledge units (KU). KUs, in turn, can be further divided into topics. KU level may also include additional information, such as categorization of topics into different levels of importance (e.g., core/elective) and articulation of learning objectives related to the KU. The hierarchical structure of KAs, KUs, and topics (where used) together form the body of knowledge of the discipline.

It is important to realize that knowledge area and knowledge unit specifications alone typically are not sufficient for providing guidance for the development of a curriculum. For example, one of the CS2013 curriculum areas is Information Management (IM). This term obviously can be interpreted in a rich variety of ways and thus, it is clear that further guidance is needed. The knowledge unit level gives us a much better articulation of the subjects that collectively form the Information Management knowledge area: *Information Management Concepts, Database Systems, Data Modeling, Indexing, Relational Databases, Query Languages, Transaction Processing, Distributed Databases, Physical Database Design, Data Mining, Information Storage and Retrieval, and Multimedia Systems.* These, however, would still not be sufficient for building an actual curriculum: most, if not, all are quite familiar as labels for both computer science and information systems faculty members, but many of them are understood differently in these separate disciplinary contexts.

Therefore, only by moving to the topic level we will have a sufficient understanding of the knowledge units to start to build a curriculum. Continuing our example from CS2013, if we have a closer look at the IM/Data Modeling knowledge unit, we find the following topics: *Data modeling, Conceptual models, Spreadsheet models, Relational data models, Object-oriented models, and Semi-structured data model*. Even though this still leaves a lot of room for interpretation, it gives significantly better guidance regarding what to include in a curriculum. The CS2013 recommendation specifies the number of classroom hours that a program is expected to spend on each knowledge unit. In the case of Information Management/Data Modeling, this total number is four, indicating that if a program wants to provide more than awareness-level coverage regarding any of the six topics, it is essential to choose the one or two on which the wants to specialize and cover others at a very high level.

Still, the example above says very little about what the students are expected to be able to do after completing the program (or after a specific module that focuses on, say, a knowledge unit). This brings us to the other aspect of educational guidance: outcome specifications. The most important forms of outcome specifications are *learning outcomes* and *competencies*.

*Learning outcomes* (or student learning outcomes) are typically specified at a course or module level, and they describe what a student will be expected to achieve after completing the course/module, specifying them often as a combination of skills and knowledge. For example, IT2017 [IT2017JointTaskGroup, 2017] defines learning outcomes as follows based on [Kennedy, 2006]:

> *"Learning outcomes 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)."*

For example, in the CS2013 curriculum the Information Management/Data Modeling knowledge unit includes the following learning outcomes [CS2013JointTaskForce, 2013, p. 114] and several others that have been excluded here):

- *"Compare and contrast appropriate data models, including internal structures, for different types of data. [Assessment]*
- *Describe concepts in modeling notation (e.g., Entity-Relationship Diagrams or UML) and how they would be used. [Familiarity]*
- *Define the fundamental terminology used in the relational data model. [Familiarity]*
- *Describe the basic principles of the relational data model. [Familiarity]*
- *Apply the modeling concepts and notation of the relational data model. [Usage]"*

It is important that the curriculum links each learning outcome to an achievement level (such as those specified by Bloom's Taxonomy [Anderson and Krathwohl, 2001] or those articulated in CS2013 [Familiarity, Usage, and Assessment]). Whichever way the achievement levels are specified, it is essential to recognize that a learning outcome specification without an articulation of the levels is meaningless.

*Competencies* are typically articulated at the program level, and they describe the knowledge, skills, and attitudes/dispositions students are expected to have at the time of graduation. Hartel and Foegeding [2006] define competency as "A general statement detailing the desired knowledge and skills of student graduating from [our] course or program." In comparison, the same authors define outcome as "A very specific statement that describes exactly what a student will be able to do in some measurable way." These definitions help in understanding the difference between competencies and learning outcomes but they leave out an essential dimension of competencies: attitudes and/or dispositions.

OECD [OECD, 2005] conducted a major project in the context of the PISA initiative that focused on the "definition and selection of key competencies." In this context, they stated the following regarding competencies:

> *"A competency is more than just knowledge and skills. It involves the ability to meet complex demands, by drawing on and mobilizing psychosocial resources (including skills and attitudes) in a particular context. For example, the ability to communicate effectively is a competency that may draw on an individual's knowledge of language, practical IT skills and attitudes towards those with whom he or she is communicating." (p. 4)*

An important element of this definition is the inclusion of psychosocial resources and the acknowledgement that context is important. The OECD report continues further later:

> "*coping with today's challenges calls for better development of individuals' abilities to tackle complex mental asks, going well beyond the basic reproduction of accumulated knowledge. Key competencies involve a mobilization of cognitive and practical skills, creative abilities and other psychosocial resources such as attitudes, motivation, and values.*" (p. 8).

Fully aligned with the OECD report, IT2017 presents a model that includes knowledge, skills, and dispositions ("personal qualities (socio-emotional skills, behaviors, attitudes) associated with success in college and career", p. 26) as the core elements of competencies.

As is the case with learning outcomes, competencies are meaningless unless they are combined with an expected level of achievement. There is no uniformity in the literature regarding the specification of achievement levels. As an example, we can take the four levels used in MSIS2016 [Topi et al., 2017, p. 9]: Awareness, Novice, Supporting (role), and Independent (contributor). MSIS2016 also recognizes that there are higher levels (such as Expert), but states that it is not realistic to expect students to attain these levels during a degree program. Often, the number of competencies specified at the program level is so high that a multi-level hierarchical structure is needed to organize them.

## V. USE OF DIFFERENT OUTCOME TYPES IN COMPUTING CURRICULA

Equipped with the conceptual background specified above, we have now the tools to compare and contrast the approaches that various computing curriculum recommendations have taken to specifying outcomes. This analysis will be helpful in the process of designing the characteristics of the CC2020 competency model and describing the similarities and differences between the degree programs in computing.

### CE2016

In the same way as in CS2013 (discussed above as an example), CE2016 [CE2016JointTaskForce, 2016] uses knowledge areas and knowledge units as its fundamental structural elements. CE2016 describes knowledge units as elements that "represent individual themes within an area." (p. 22). Interestingly, CE2016 does not include topics, but it specifies learning outcomes as the "lowest level of the hierarchy." CE2016 includes 12 computing engineering knowledge areas and within them altogether 135 knowledge units. Some of the knowledge units are associated only with core learning outcomes and some both with core and elective outcomes.

For example, the CE curriculum has a knowledge area Software Design, and within it a knowledge unit Database systems. Within this knowledge unit, there are four core learning outcomes:

- *"Explain how use of database systems evolved from programming with simple collections of data files;*
- *Describe the major components of a modern database system;*
- *Describe the functionality provided by languages such as SQL; and*
- *Give examples of interactions with database systems that are relevant to computer engineering*" [CS2016, p. 103].

Interestingly, CE2016 does not specify attainment levels for the learning outcomes, but these levels can at least to a certain extent be determined based on the verbs used to describe the learning outcomes. For example, the learning outcomes specified for the Database systems knowledge unit clearly indicated that the CE curriculum does not expect CE graduates to be able to design databases; all verbs suggest that the students are expected to attain awareness level knowledge of Database systems. Obviously, the situation is very different for other knowledge units.

### CS2013

As discussed and described with an example above, the foundational structure of the most recent computer science curriculum (CS2013) is based on knowledge areas divided into knowledge units. Each knowledge unit, in turn, is specified further by articulating its topics (a clear extension of the knowledge area – knowledge unit hierarchy) and learning outcomes. Overall, CS2013 presents 1110 outcomes, slightly over 50% of which are in the core, and about the same percentage (but, of course, not the same outcomes) are at the Familiarity level. The CS2013 learning outcomes have not been further categorized into higher-level categories, and there is no specific mechanism to deal with learning outcomes that might need knowledge components from multiple knowledge units.

## IS2010

Unlike all the other ACM/AIS/IEEE-CS curriculum recommendations, the latest undergraduate level curriculum recommendation for information systems uses a course as its organizing element, following the structure established in IS 2002 [Gorgone et al., 2002] and IS'97 [Davis et al., 1997). The recommendation includes seven required courses and a large selection of electives. The description of each course consists of a set of detailed learning objectives (in practice, learning outcomes) and a set of topics to be covered in the course. The learning outcomes are detailed and typically presented at least at the same level as the learning outcomes specified in CS2013. For example, the Data and Information Management course (IS2010.2) includes learning outcomes such as:

6. *"Use at least one conceptual data modeling technique (such as entity-relationship modeling) to capture the information requirements for an enterprise domain.*
7. *Link to each other the results of data/information modeling and process modeling.*
8. *Design high-quality relational databases.*
9. *Understand the purpose and principles of normalizing a relational database structure.*
10. *Design a relational database so that it is at least in 3NF."*

These same learning outcomes would be associated with topics such as Conceptual data model, Logical data model, and Physical data model, which all have a number of subtopics.

As a separate structure, IS2010 includes a two-level (knowledge areas divided into knowledge units) representation of an IS Body of Knowledge, but only for information systems specific knowledge areas (seven of them). For example, there is a Data and Information Management knowledge area that includes (among others) knowledge units *Data and Information Management at Conceptual and Logical Levels* and *Physical Database Implementation*. It does not, in practice, provide any information beyond the knowledge area level regarding general computing knowledge areas (which are specified in CS2008/CS2013), individual foundational knowledge areas, or domain-related knowledge areas.

IS2010 does, however, include significant additional coverage of program level graduate capabilities. The curriculum specifies a set of *High-level IS Capabilities* which, in turn, are a foundation for *IS Specific Knowledge & Skills*, *Foundational Knowledge & Skills*, and *Domain Fundamentals*. These form the conceptual foundation for curriculum topics that, as discussed above, are organized into courses.

The High-level IS Capabilities as specified in IS2010 (p. 362) are broad and stable:

- *Improving Organizational Processes,*
- *Exploiting Opportunities Created by Technology Innovations,*
- *Understanding and Addressing Information Requirements,*
- *Designing and Managing Enterprise Architecture,*
- *Identifying and Evaluating Solution and Sourcing Alternatives,*
- *Securing Data and Infrastructure, and*
- *Understanding, Managing, and Controlling IT Risks.*

## IT2017

As already mentioned above, IT2017 was one of the first two computing curriculum documents that chose the competency-based approach as the fundamental structural mechanism. In IT2017, the highest-level organizational unit is "IT domain," which in this case does not refer to an area of goal-oriented activity that IT supports and enables (as is the case in the information systems documents). Instead, IT domain is a category of competencies—on p. 11, the IT2017 report states that "The task group established desired competencies and allowed IT domains to follow from the competencies." IT2017 specifies 10 essential IT domains and nine supplemental domains. Each IT domain is further divided into subdomains and described with scope statements.

Using again data and information management as an example, the IT domain on Information Management (ITE-IMA) consists of the following competencies [IT2017, p. 54]:

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

The subdomains, in turn, are as follows [IT2017, p. 54]:

1. *Perspectives and impact*
2. *Data-information concepts*
3. *Data modeling*
4. *Database query languages*
5. *Data organization architecture*
6. *Special-purpose databases*
7. *Managing the database environment.*

Interestingly, IT2017 does not specify attainment levels for the competencies; instead, it identifies a level of *learning engagement* for each of the subdomains. There are three possible levels: L1 indicates a "minimal degree of engagement" and L2 and L3 indicate "medium and large degrees of learning engagement." The document gives examples of practices appropriate L2 and L3, such as "investigative laboratory activities, prototyping of computational artifacts, authentic projects, public professional presentations, and other authentic performances" [IT2017], citing [Wiggins, 2011].

Finally, Appendix B of IT2017 provides suggested IT performances, which "may be used to develop course learning outcomes for a given IT course" or as "possible assessments to measure student performance." [IT2017, p. 11]. A closer evaluation of the IT performances reveals a highly useful list of performance specifications that mostly are, indeed, at the course module learning outcome level, and provide additional clarification regarding the nature of the domains.

## MSIS2016

MSIS2016 differs from the other reports included in this review because it focuses on the master's level. It is, however, useful to include it because it—in the same way as IT2017—presents a competency hierarchy as the fundamental underlying structure for the document. MSIS2016 starts with the specification of three realms: IS competencies, foundational individual competencies, and domain competencies. Most of the document focuses on IS competencies because they are unique for an information systems degree whereas the other two categories are shared with many other degree types. MSIS2016 acknowledges, however, the essential role of the other two in the preparation of an IS professional.

MSIS2016 identifies nine IS competency areas, divided further into 88 competency categories, each of which presents multiple sample competencies. Staying with our interest area of data and information management, one of the MSIS2016 competency areas is Data, Information, and Content Management. Within this competency area there are 15 competency categories, examples of which are specified below. For competency category 12, the representation below includes also the four sample competencies specified in the report [MSIS2016, p. 70-71]:

12. *"Capturing and structuring data and information requirements using appropriate conceptual modeling techniques*
    a. *Interview and observe users to identify their data needs.*
    b. *Evaluate domain activities to understand how they can be improved with effective use of data.*
    c. *Structure domain data requirements using Enhanced Entity-Relationship (EER) modeling and alternative models such as hierarchical, network, cube, etc. as appropriate to the domain.*
    d. *Evaluate appropriateness of different data structures for representing and storing data.*
13. *Developing a logical level representation of data based on a conceptual model*
14. *Implementing a database solution to serve systems consisting of multiple applications*
15. *Using a contemporary data manipulation and retrieval language effectively"*

MSIS2016 includes two additional central elements related to the competencies: competency levels and professional profiles. As discussed above, MSIS2016 specifies four levels of attainment for the competencies (Awareness, Novice, Supporting [role], and Independent [contributor]), with the acknowledgement of at least one higher level (Expert), which is not, however, feasible as an expectation for a master's degree program. Competency levels are closely related to professional profiles in that the attainment expectations associated with a competency category depend on the desired professional profile. MSIS2016 focuses on six profiles: Business Information Manager, Project Manager, Business Analyst, Systems Analyst, Enterprise Architect, and IT Consultant [based on CEN, 2012]. With examples, the document demonstrates how the competency expectations vary depending on the profile.

## SE2014

The core structure in SE2014 [SE2014JointTaskForce, 2015] is entitled the Software Engineering Education Knowledge (SEEK). It consists of 10 *knowledge areas*, each of which includes three to six *knowledge units*. Each knowledge unit is further divided into up to 15 *topics*. As was the case in the context of computer engineering, it is difficult to use data management as an example because the area is covered only in two contexts:

- as topic *Database design* (DES.dd.2) within knowledge unit *Detailed design*, in turn within knowledge area *Software design* and
- as topic *Database fundamentals* (CMP.cf.10) within knowledge unit *Computer science foundations* within knowledge area *Computing essentials*.

Unlike CS2013, there are no specific learning outcomes associated with knowledge units or topics in SE2014. SE2014 does, however, present seven high-level student outcomes categorized as follows [SE2014, pp. 20-21]:

1. *"[Professional Knowledge]*
2. *[Technical Knowledge]*
3. *[Teamwork]*
4. *[End-User Awareness]*
5. *[Design Solutions in Context]*
6. *[Perform Trade-Offs]*
7. *[Continuing Professional Development]"*

Later in the document, SE2014 emphasizes the importance of the "outcomes or learning objectives" as a foundation for curriculum design (p. 39): "The student learning outcomes (see Chapter 3) should be used as a basis for designing and assessing software engineering curricula in general. These can be further specified for the design of individual courses." Thus, SE2014 also presents a strong call for anchoring the curriculum into high-level outcome expectations.

## Summary

As is clear based on Table 1 below, the key documents providing educational guidance for degree programs in computing vary significantly in terms of

- their core conceptual structure,
- the role of body of knowledge in the document,
- whether or not the document specifies detailed learning outcomes,
- whether or not the document specifies a competency framework and, in general, is based on a competency-driven approach, and
- whether or not the document identifies multiple degree options based on a variety of professional profiles.

The only element of these reports that all of them share is some type of a specification of high-level graduate characteristics.

Table 1: Comparison of Computing Curricula

| | **CE2016** | **CS2013** | **IS2010** | **IT2017** | **MSIS2016** | **SE2014** |
|---|---|---|---|---|---|---|
| **Core structure** | Knowledge area – Knowledge unit – Learning outcome | Knowledge area – Knowledge unit – Topic/ Learning outcome | Course – Topic/ Learning outcome | IT Domain – Competency | Competency area – Competency category – Competency | Knowledge area – Knowledge unit – Topic |
| **Body of knowledge** | Core structure | Core structure | Supporting material | No | No | Core structure |
| **Detailed learning outcomes** | Associated with Knowledge units | Associated with Knowledge units | Associated with Courses | "Performances" associated with subdomains are closely related | No | No |
| **High-level graduate characteristics** | Yes | Yes | Yes | Yes | Yes | Yes |
| **Competency framework** | No | No | No | Yes | Yes | No |
| **Professional profiles** | No | No | Yes | No | Yes | No |
| Note: The original idea for this table emerged from collaborative work at the August 2017 CC2020 Task Force meeting. The collaborative process and contributions of CC2020 are greatly appreciated. | | | | | | |

# VI. DISCUSSION

## Finding a Common Foundation for CC2020

The goals of the CC2020 process discussed at the beginning of this paper are clearly very significant, and it will be an important achievement if CC2020 is able to offer an integrated competency model that can be applied to all computing programs. This is the goal towards which the task force is currently working on. It will also be very useful if CC2020 is able to provide a tool that allows administrators, faculty, student, and parents to compare different computing programs using a well-defined framework and terminology that works with all programs. Both of these goals can, however, be achieved only if the task force is somehow able to integrate the disparate core structural approaches represented in the existing curriculum models.

In practice, the CC2020 task force will have to choose between a body of knowledge approach (followed by CE, CS, and SE) and a competency approach (IT and MSIS). Based on the positive experiences from IT2017 and MSIS2016 and the broad use of industry-led competency models such as e-CF 3.0 and SFIA, the task force has decided to use competencies as the common currency. Fortunately, it will not be necessary to create the competencies for CS and CE entirely from scratch: both CS2013 and CE2016 include extensive specifications of knowledge unit –level learning outcomes. These are, of course, at a different level of abstraction from the competencies specified in IT and MSIS, but at least they are focused on desired student outcomes instead of topics to be taught.

In addition, all recommendations include some type of specification of high-level graduate characteristics, which will also provide guidance regarding the appropriate competencies needed for the competency model of computing and the degree program comparison tool.

It will not be a trivial task to determine the appropriate level of abstraction and connectedness to the expected work context for the joint computing competency model, but it is likely that such an undertaking will be very helpful in strengthening our ability to communicate the reasons why a student should select a specific type of a degree program and the graduates' preparedness for specific types of jobs after graduation. Fortunately, there are existing research programs that will provide a good starting point for this work also from the computer science perspective (such as those reported in [Bröker, Kastens, & Magenheim, 2015] and [Bröker & Magenheim, 2014].

## Guidance from industry

As discussed both in IT2017 and MSIS2016, during the recent years significant efforts strongly affiliated with industry and government have developed competency models for the broader computing industry. Space constraints prevent us from discussing those at a detailed level, but it is essential that they be considered in any project that evaluates and proposes computing competencies. The most important ones of these are SFIA (Skills Framework for the Information Age; www.sfia-online.org) originally by the British Computing Society and now by the SFIA Foundation, the pan-European e-CF (e-competency framework; www.ecompetences.eu/e-cf-3-0-download), and Japanese iCD (i-Competency Dictionary; https://www.ipa.go.jp/english/humandev/icd.html).

## Lessons learned for IS202X

For the information systems education community, one of the important open questions is the approach chosen for the next version of undergraduate level guidance for degree programs in information systems (IS202X). All IS documents since IS'97 have been based on a course-driven curriculum recommendation model. This model has served the IS community well but it also has significant limitations, particularly when the intention is to serve the global IS community with the same process and document. One of the reasons MSIS2016 makes a strong case for following the competency-based approach is that expected graduate competencies are shared much more widely around the world than specific ways to offer and organize educational programs. For this same reason, the next undergraduate level IS recommendation project should at least consider the competency-based approach.

There are other reasons to do so: The competency-based approach

- is clearly centered on students and their learning instead of faculty members and what they teach (or degree programs and the courses they offer and require)
- goes far beyond knowledge and recognizes the essential role of not only knowledge and skills but also attitudes or dispositions as highly relevant outcomes of degree programs
- forms a common language for natural conversations with prospective employers, who are looking for graduates with job-related competencies instead of only an understanding of what the students have been taught, and
- would allow the new undergraduate process to build directly on the work done in MSIS2016—many of the competency categories and areas are likely to be shared between the levels even though the expected attainment levels will vary.

## VII. CONCLUSION

In this paper, we have described the key characteristics of the Computing Curricula 2020 process, discussed the reasons why it is essential for information systems to be involved in this process, and explored the core structures of existing computing curriculum recommendation documents, particularly from the student outcome and competency perspective. In an analysis of the existing documents we have discovered that the approaches chosen by different recommendations vary significantly; the two main categories are the body of knowledge (knowledge area – knowledge unit) structure used by CE, CS, and SE and the competency structure used by IT and MSIS. Finding a way to express the competency expectations of all degree program types in computing with the same terminology and at the same level of abstraction will be a key to the success of the CC2020 project. The upcoming process to develop of a new IS undergraduate recommendation will also be able to benefit from this work and contribute to it.

## VIII. REFERENCES

Anderson, L. W., and D.R. Krathwohl (2001). *A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of educational objectives.* New York: Longman.

Bröker, K., & J. Magenheim (2014). "Are there competences every computer scientist should have? Presented at the *2014 IEEE Global Engineering Education Conference (EDUCON)*, pp. 999-1002.

Bröker, K., U. Kastens, and J. Magenheim (2015). Competences of Undergraduate Computer Science Students (pp. 77–96). Presented at the *KEYCIT 2015*.

CE2016JointTaskForce (2016). *Computer Engineering Curricula 2016*. ACM and IEEE-CS. https://www.acm.org/education/curricula-recommendations (current Aug 19, 2017).

CEN (2012). *European ICT Professional Profiles, European Committee For Standardization, CEN/WS ICT, ICT Skills Workshop (IT Profiles and Curricula)*. CEN.

Couger, J. D. (1973). Curriculum recommendations for undergraduate programs in information systems. *Communications of the ACM*, *16*(12), pp. 727–749.

CS2013JointTaskForce. (2013). *Computer Science Curricula 2013*, https://www.acm.org/education/curricula-recommendations (current Aug 19, 2017)

Davis, G. B. et al. (1997). IS '97: model curriculum and guidelines for undergraduate degree programs in information systems. *ACM SIGMIS Database*, *28*(1), 1–94.

Gorgone, J. T. et al. (2002). IS 2002 model curriculum and guidelines for undergraduate degree programs in information systems. *Communications of the Association for Information Systems*, *11*(1). Retrieved from http://aisel.aisnet.org/cais/vol11/iss1/1.

Hartel, R. W., and E.A. Foegeding (2006). Learning: Objectives, Competencies, or Outcomes? *Journal of Food Science Education*, *3*(4), 69–70.

IT2017JointTaskGroup. (2017). *Information Technology Curricula 2017*. ACM and IEEE-CS, https://www.acm.org/education/curricula-recommendations (current Aug 19, 2017).

Kennedy, D. (2006). *Writing and using learning outcomes: a practical guide.* University College Cork.

OECD (2005). *The Definition and Selection of Key Competencies*. Retrieved from https://www.oecd.org/pisa/35070367.pdf (current Oct 10, 2017).

SE2014JointTaskForce. (2015). *Software Engineering 2014*. ACM and IEEE-CS. https://www.acm.org/education/curricula-recommendations (current Aug 19, 2017).

Shackelford, R. et al. (2005). *Computing Curricula 2005: The Overview Report*. ACM, AIS, and IEEE-CS. https://www.acm.org/education/curricula-recommendations (current Aug 19, 2017).

Topi, H. (2017, forthcoming). Role of Information Systems in the CC2020 Initiative. *ACM Inroads*, 8(4).

Topi, H. et al. (2017). MSIS 2016: Global Competency Model for Graduate Degree Programs in Information Systems. *Communications of the Association for Information Systems*, *40*(1). Retrieved from http://aisel.aisnet.org/cais/vol40/iss1/18.

Wiggins, G. (2011). A True Test: Toward More Authentic and Equitable Assessment. *Phi Delta Kappan*, *92*(7), pp. 81–93.

## ABOUT THE AUTHOR

**Heikki Topi** is Professor of Computer Information Systems at Bentley University. He has been actively involved in leadership roles in national computing curriculum development and evaluation efforts since early 2000s (including IS 2002, CC2005 Overview Report, co-chair of IS 2010 and MSIS 2016, and currently CC2020) and written extensively on topics related to IS education and usability of search and large scale enterprise systems. He is coauthor of *Modern Database Management* and co-editor of IS/IT volume of CRC/Chapman & Hall's *Computing Handbook*. He currently serves as AIS Vice President of Education.