Journal of Information Systems Education

Volume 30
Issue 4
Fall 2019

Invited Paper

IS2010: A Retrospective Review and Recommendation

Paul M. Leidig, Roger C. Ferguson, and John H. Reynolds

Recommended Citation: Leidig, P. M., Ferguson, R. C., & Reynolds, J. H. (2019). Invited Paper: IS2010: A Retrospective Review and Recommendation. *Journal of Information Systems Education,* 30(4), 298-302.

Article Link: http://jise.org/Volume30/n4/JISEv30n4p298.html

Initial Submission:	3 May 2019
Accepted:	8 October 2019
Abstract Posted Online:	n/a
Published:	12 December 2019

ISSN: 2574-3872 (Online) 1055-3096 (Print)

IS2010: A Retrospective Review and Recommendation

Paul M. Leidig
Roger C. Ferguson
John H. Reynolds
Grand Valley State University
Allendale, MI 49401, USA
paul.leidig@gvsu.edu, roger.ferguson@gvsu.edu, john.reynolds@gvsu.edu

ABSTRACT

The Information System (IS) model curriculum has been advanced by the Association of Computing Machinery (ACM), the Association for Information Systems (AIS), and other associations since the 1970s. The IS2002 and IS2010 curriculum models have been positively received by both academic institutions and industry alike. Each of these models used design principles to help guide the development process; these principles included concepts such as maximizing program flexibility or course sequencing to improve depth of knowledge. The most current undergraduate model (IS2010) that guides university curriculum is nearly 10 years old. In today's IS field, a curriculum should address previous design concepts; plus the environment in which IS professionals work has become even more dynamic and multifaceted. Given these challenges, a new IS model curriculum would benefit by including more programming concepts along with course sequencing. The authors propose a two-course sequence in areas such as computer programming to increase depth of knowledge and keep some program flexibility. Further, the authors recommend: (1) require a minimum amount of programming, (2) require technical infrastructure coverage, (3) allow for specialization, (4) specify sequencing to provide depth, and (5) identify both core competencies and course structure. These recommendations are not a criticism but build a new model with the strengths and knowledge gained in the past 10 years.

Keywords: Curriculum design & development, Model curricula, IS education, IS2010

1. INTRODUCTION

Since the 1970s, the development of the undergraduate Information Systems (IS) model curriculum has been advanced by the Association of Computing Machinery (ACM), the Association for Information Systems (AIS), and various collaborating associations (Couger, 1973; Nunamaker, Couger, and Davis, 1982; ACM, 1983; Davis et al., 1997; Gorgone et al., 2002; Topi et al., 2010). Though the history of the development of the IS model curriculum has been steady, roughly a new model every 10 or so years, there has been no shortage of controversies regarding the structure and content of the IS model curriculum (Longenecker, Feinstein, and Clark, 2013). These controversies include such things as how flexible an IS model curriculum should be or to what extent a model should mandate the courses that an undergraduate must take to graduate with an IS degree. Other issues related to curriculum development dealt with course sequencing or prerequisite structures. Perhaps the most serious point of contention has been over the necessity of technical depth, such as programming, being a part of the IS model curriculum.

The importance of getting an IS model curriculum that fits the needs of the IS community is even more critical today given (1) the increased number of IS jobs that have been created in the past 10 years, both in terms of overall number and in terms of skillsets needed, and (2) the severe shortage of skilled talent. Recently, studies have compared curricula and degree programs in Russia and Portugal (Pereira, Aleksandr, and Popova, 2018) and German-speaking countries (Jung and Lehrer, 2017) using both local and international models to try to solve this problem.

The two most recent IS models, IS2002 and IS2010, have accomplished much in terms of satisfying industry and university needs. For the IS2010 curriculum model, the following were the main goals of this model (Topi et al., 2010, pp. 8-11):

- 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.

To accomplish these main goals, the designers of the 2010 IS curriculum model had to be flexible to allow universities to offer an IS program that fits the goals of each university, IS department, and surrounding IS community. Further, the design of the IS model curriculum had to match industry needs as well.

The principles guiding the IS2010 model curriculum design were as follows:

1. The model curriculum should represent a consensus from the Information Systems community.
2. The model curriculum should be designed to help Information Systems programs produce competent and confident entry-level graduates well-suited to workplace responsibilities or further studies of Information Systems.
3. The model curriculum should guide but not prescribe. Using the model curriculum guidelines, faculty can design their own courses, and schools can design their own programs.
4. The model curriculum should be based on sound educational methodologies and make appropriate recommendations for consideration by Information Systems faculty.
5. The model curriculum should be flexible and adaptable to most Information Systems programs.
6. The model curriculum is not restricted to a specific domain; all Information Systems programs are, however, linked to some domain.
7. The model curriculum has a core of content that is common to all Information Systems programs internationally.
8. The model curriculum has career targets that require both core and elective content.
9. The model curriculum does not focus on specific issues related to pedagogy. This is not a reflection of our understanding of the importance of pedagogical decisions; we simply believe that these highly significant issues are outside the scope of this document.

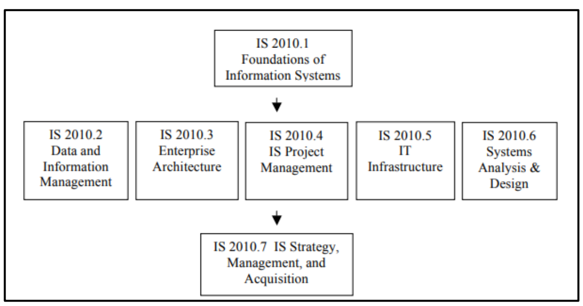

Figure 1. IS2010 Core Courses

The IS2010 model curriculum (Figure 1) provided a minimalist core and allowed universities and IS departments a great deal of flexibility. It has been a model that has served the IS community, universities, and industry for the past 10 years.

2. CURRENT SITUATION

Even with a successful model curriculum, the consequences of the IS2010 model curriculum have been cited in many publications (Longenecker, Feinstein, and Babb, 2013; Babb et al., 2014). The flexibility of the IS2010 model, allowing universities to create IS programs that fit a local faculty and environment, was an excellent goal. This flexibility allowed IS programs to be developed that enhanced student appeal to the major. Unfortunately, this flexibility has caused significant inconsistency in the requirements for an IS degree.

Another consequence of the IS2010 model has to do with course sequencing. The model curriculum removed previous course prerequisite paths. This design attribute allowed easier scheduling of courses, again giving IS departments flexibility. Unfortunately, this flexibility also allowed students to graduate without reaching higher levels of Bloom's taxonomy. Courses without prerequisites typically start at the lower levels of Bloom's taxonomy and do not have the time to reach higher levels. Perhaps the most controversial issues of the IS2010 model curriculum deal with the elimination of a requirement for programming and a reduction in the emphasis on technical foundations in general (Babb et al., 2014).

There has been a growing divergence of IS programs between those focused on a general business view of the role of an information system and those with a more technical focus of designing and building information systems. Programs in the former group, often housed within a business unit, tend to minimize or eliminate the requirement of programming. Programs that are housed in either an autonomous computing unit or in a business unit that offers a technical concentration in information systems, include programming as a required part of an IS degree (Reynolds, Ferguson, and Leidig, 2016).

As indicated earlier, there has been no shortage of dissatisfaction with the current IS model curriculum. While the 2010 model curriculum emphasized (1) program flexibility over program consistency and (2) curricular breadth and flexibility over student knowledge depth, the previous IS2002 model curriculum was criticized for a lack of program requirement flexibility, and yet was not criticized for program inconsistency nor depth of student technical knowledge. In the decade since the last model curriculum was developed, there have been significant changes in the IS field. Perhaps the most consistent recurring complaint regarding the IS2010 model is that current graduates' technical skills do not appear to meet current industry needs. The IS discipline would be well-served with a minimum standard and consistent curriculum that articulates a foundation of breadth and depth that educational institutions can use to develop and offer programs that meet current industry stakeholder needs.

The IS discipline faces a continuous stream of technological developments, new trends, and buzzwords. This dynamic situation poses a significant challenge for faculty and administrators responsible for delivering a relevant and consistent curriculum preparing graduates as IS professionals. The need for effective information systems and information technology to keep up with the rapid pace of technological change demands a current and updated IS curriculum.

The academic discipline of information systems has been in existence for over a half-century, and has been offered by a variety of names, such as information systems, management information systems, computer information systems, etc. As the field of information systems grew and expanded, the variety of programs also expanded, using a plethora of program names resulting in an even more diverse set of academic offerings. Additionally, these diverse programs were in a variety of administrated alignments, including in business schools with MIS/BIS programs, computing/technical schools, information schools (iSchools), informatics, and many other titles. At the same time, the broader computing programs similar to IS were expanding as well, including computer science, software engineering, information technology, cybersecurity, and data science, to name a few. Therefore, any model IS curriculum faces the challenge of addressing the needs of all, or at least most, of these programs. The challenge is to identify the common core of what constitutes an IS program while leaving flexibility to address local needs (Gammack, 2011).

3. A NEW MODEL CURRICULUM

Beginning with the first model curriculum, each succeeding IS model curriculum has framed the requirements in three basic knowledge areas: (1) IS technology, (2) IS systems concepts and processes, and (3) organizational functions and management. Thus, any development of a new model is expected to update the current professional skills needed by industry employers in each of these three areas. When a published model fails to fully address one of these areas, industry and academic leaders will seek a return to full coverage of all areas. The general course requirements in the IS2010 model are illustrated in Figure 1. The figure shows the suggested course content and the course prerequisite sequences. This model demonstrates the minimum content areas and generally flat hierarchical structure. As both industry and academic representatives articulate the needs of information systems graduates, this model needs modification to expand, or change, these requirements.

3.1. The Need for Technical Foundations

While one might argue that the corporate CEO has no need to be an accomplished stenographer/typist, it would be foolish to say that anyone today doesn't need to know the fundamentals of keyboarding, regardless of their place in the organization. Similarly, whatever the focus or specification of an individual student or program, a student cannot be adequately prepared without the fundamentals of computing technology. Consider that the fundamentals in *Information Systems Management* (McNurlin, Sprague and Bui, 2009) have not really changed, nor have the fundamental technical concepts of computing, in spite of the rapid social and technological change today.

Over the past decade, we have witnessed numerous new applications of technology in organizations. Agile software development, SCRUM, human-centered-design, and other techniques have changed the face of programming and heightened the demand for these skills from IS programs. The Internet of Things (IoT), big data and data science, along with increased use of artificial intelligence, AI agents, machine learning, data mining, and many other systems used in analytics for modern information systems, have led to a renewed interest in a broad coverage of new technologies in those programs. These are only a few examples that illustrate the vast and growing array of technological applications that have become standard parts of today's IS program requirements.

At the same time as the role of IS within organizations has been increasingly critical, the environment in which IS professionals need to work has become even more dynamic and multifaceted. With the digitization of work, improved automated processes, and evidence-based decision making, IS professionals are called upon to provide an ever-increasing complex system. All of this builds on the ever present need for graduates to exhibit complex problem-solving, social, and communication skills.

3.2. The Need for a Sequencing Compromise

With the increase in professional programs in the traditional liberal arts university, there has been a movement away from the traditional two years of general education followed by two years of courses in a major. Professional programs generally follow a parallel approach to general education and a specific major by having students take the first class in their major their first semester. By following this model, professional programs can build more depth into their programs. Consequently, IS2002 continued the highly sequential curriculum of IS'97 by specifying a prerequisite structure that focused on achieving a high level of scaffolding in the curriculum.

Professional programs (e.g., colleges of business) generally followed this approach for their majors but found it difficult to vertically integrate a second area such as information systems, preferring to offer those courses in the last two years. IS2002 was criticized for its rigid structure, and IS2010 responded to this by flattening the curriculum, thereby limiting the depth.

This becomes an issue for schools that want to build more depth by having their students take computing classes in all four years of their program. Without sequencing, schools have to choose whether to offer classes at a level for students in the first two years or the last two years. Since those teaching the classes are likely to have both groups in the same class, the former leads to under-challenged upper division students, while the latter may encourage new students to drop the program.

The compromise suggested below (Figure 2) has multiple two-course sequences inserted between an introductory course and a senior sequence followed by a capstone course. The two-course sequence in programming and the two-course sequence in infrastructure form a technical foundation that *should* be completed before the more in-depth two-course sequence of database and systems analysis & design. The second upper-division, two-course sequence is elective and leaves room for a sequence in an area of the student's choosing. Finally, the project management sequence is in the last year followed by a capstone course. Regardless of a particular program's structure, an applied capstone project provides an opportunity to apply prior content.

Schools who wish to put most of their computing courses in the last two years may not wish to sequence between the two-course groups, but would be able to leave the two-course sequences intact. Schools who prefer to vertically integrate the major should implement the sequencing between the two-course groups to avoid any of the aforementioned issues.

4. RECOMMENDATIONS

The Association for Computing Machinery (ACM) has been involved in every model curriculum published, beginning in the 1970s. Along with the Association for Information Systems (AIS), beginning in the 1990s, and other organizations, these organizations are in the early stages of developing a new model curriculum (IS2020). We recommend that any new model curriculum for the next decade should include the following critical updates missing in the current guide:

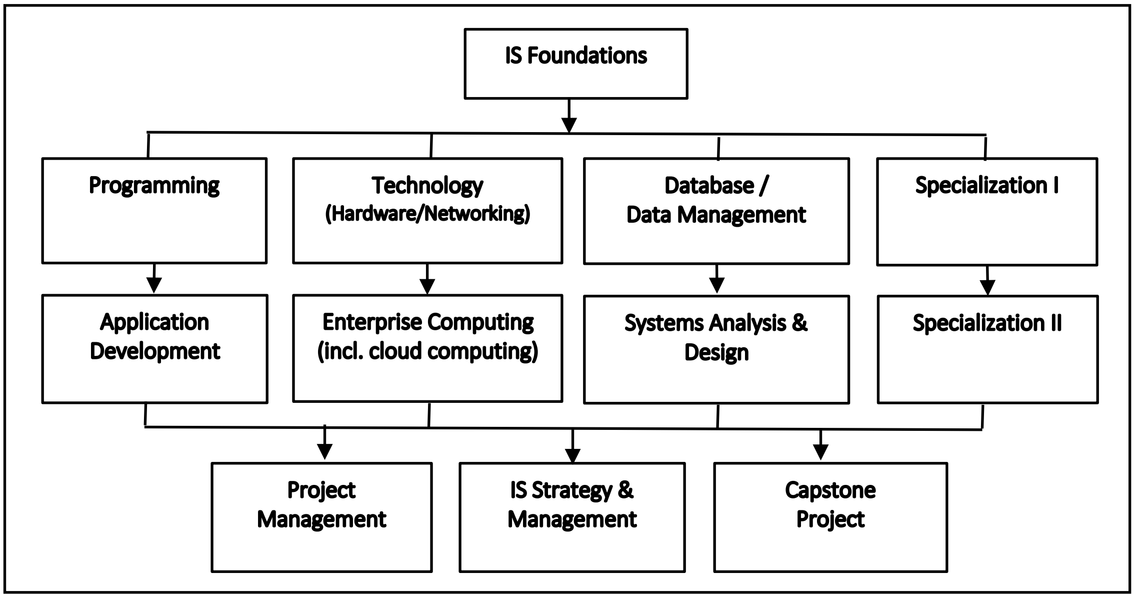

Figure 2. Sample Curriculum Design

1. Require a minimum amount of programming
2. Require technical infrastructure coverage
3. Allow for specialization in the curriculum
4. Specify sequencing to provide depth
5. Identify both core competencies and course structure

The diagram above (Figure 2) shows the overall conceptual design of the proposal as a sample of how this model curriculum might be structured. In this model, some of the fundamental technical content from IS2002 is required and inserted after the IS2010 Foundation of Information Systems and before the upper division IS courses.

This sample, "new" model curriculum reinstates the programming and infrastructure from previous models and provides a structure for sequencing and added depth in courses. Whatever the focus or specialization of an individual student or program, a student cannot be adequately prepared without the fundamentals of computing technology. Consider that the fundamentals of information systems espoused in earlier curriculum guides (1970s-2002) have not really changed, nor have the fundamental technical concepts of computing, in spite of the rapid social and technological change observed today.

Historically, most computing curriculum models have been designed to specify a typically hierarchical knowledge units (KU) structure that provides a body of knowledge (BoK). While this curriculum structure provides guidance on delivering measurable content knowledge, it does not provide much information about what graduates are expected to be able to do with that knowledge upon graduation. Several recent curriculum guides have provided a set of graduate competencies. Competencies include the knowledge units, but also include skills that should be learned and demonstrated, along with dispositions, or character traits, that graduates should exhibit. Regardless of which knowledge units the framers of a new model recommend, we strongly recommend that any new model include both appropriate competencies for all flavors of information systems programs. These competencies should include a core set of knowledge units, the demonstrable skills that are necessary, and a description of desired graduate dispositions. This hybrid approach provides both the desired learned competencies that can then be measured and assessed, and it also could provide a more tangible model curriculum structure that could be used to create program requirements.

5. REFERENCES

ACM. (1983). *ACM Recommendations for Information Systems*, *Volume II*, New York, NY: ACM Committee on Computer Curricula of ACM Education Board.

Babb, J., Longenecker, H., Jr., Baugh, J., & Feinstein, D. (2014). Confronting the Issues of Programming In Information Systems Curricula: The Goal is Success. *Information Systems Education Journal*, 12(1), 42-72.

Couger, J. (1973). Curriculum Recommendations for Undergraduate Programs in Information Systems. *Communications of the ACM*, 16(12), 727-749.

Davis, G., Gorgone, J., Couger, J., Feinstein, D., & Longenecker, H., Jr. (1997). *IS '97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems*. New York, NY: ACM and Park Ridge, IL: AITP (formerly DPMA).

Gammack, J. G. (2011). *The Book of Informatics Revised.* Melbourne, Australia: Cengage Learning.

Gorgone, J., Davis, G., Valacich, J., Topi, H., Feinstein, D., & Longenecker, H., Jr. (2003). IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. *Communications of the Association for Information Systems*, 11(1), i-53.

Jung, R. & Lehrer, C. (2017). Guidelines for Education in Business and Information Systems Engineering at Tertiary Institutions. *Business Information Systems Engineering*, 59(3), 189-203.

Longenecker, H., Jr., Feinstein, D., & Babb, J. (2013). Is There a Need For a Computer Information Systems Model Curriculum? Proceedings of the Information Systems Educators Conference, San Antonio, TX.
Longenecker, H., Jr., Feinstein, D., & Clark, J. (2013). Information Systems Curricula: A Fifty Year Journey. *Information Systems Education Journal*, 11(6), 71-95.
McNurlin, B., Sprague, R., & Bui, T. (2009). *Information Systems Mangement. (8 ed.)*. New York, NY: Pearson.
Nunamaker, J., Couger, J., & Davis, G. (1982). Information Systems Curriculum Recommendations for the 80s: Undergraduate and Graduate Programs. *Communications of the ACM*, 25(11), 781-805.
Pereira J., Aleksandr, I., & Popova, E. (2018). Evaluation of Information Systems Curriculum in Portugal and Russia. In Rocha, Á., Adeli, H., Reis, L. P., & Costanzo, S. (eds.) *Trends and Advances in Information Systems and Technologies*, WorldCIST'18: *Advances in Intelligent Systems and Computing*.
Reynolds, J., Ferguson, R., & Leidig, P. (2016). A Tale of Two Curricula: The Case for Prerequisites in the Model Curriculum. *Information Systems Education Journal*, 14(5), 17-24.
Topi, H., Valacich, J., Wright, R., Kaiser, K., Nunamaker, J. F., Sipior, J., & de Vreede, G. (2010). IS 2010: Curriculum Guidelines for Undergraduate Degree Programs in Anformation Systems. *Communications of the Association for Information Systems*, 26(1).

AUTHOR BIOGRAPHIES

Paul M. Leidig is a professor and Director of the School of Computing and Information Systems at Grand Valley State University. Leidig holds several positions on computing boards and organizations, including the Computing Sciences Accreditation Board (CSAB), the Association of Computing Machinery (ACM) Education Council, and a past Commissioner on the Computing Accreditation Commission of ABET. Leidig currently serves as chair of the CSAB Criteria Committee, co-chair of the ACM Data Science Taskforce, co-chair of the Information Systems 2020 Curriculum Taskforce, and the Computing Curricula 2020 task force report group. He is a past president of the Association of Information Technology Professionals (AITP) Education Special Interest Group (EDSIG) and was named an EDSIG Fellow. He began his career over 40 years ago as the data processing director of a regional hospital. He teaches courses on the management of information systems and information systems policy, and has authored several textbooks on database systems and applications. Leidig received his Ph.D. from Virginia Commonwealth University, M.B.A from James Madison University, and B.S. from Eastern Mennonite University. His work has been published in *Journal of Information Systems Education* and *Information Systems Education Journal*. His work has also been included in numerous proceedings, including: ACM SIGCSE, ACM ITiCSE, ACM SIGITE, AMCIS, ICIS, AITP ISECON, and EDSIGCON.

Roger C. Ferguson is a professor in the School of Computing and Information Systems at Grand Valley State University. He received his Ph.D. in software engineering from Wayne State University, an M.S. in computer science from Central Michigan University, and a B.S. in electrical engineering from Michigan Technological University. He started his career as a software developer at ABW Corporation in Ann Arbor, MI, and worked as a programmer/analyst/researcher at ITI Corporation in Detroit, MI. He has recently worked on research in the area of curriculum development for information system majors. His work has been published in *Journal of Information Systems Education* and *Information Systems Education Journal*. His work has also been included in proceedings including AITP ISECON and EDSIGCON.

John H. Reynolds is an associate professor in the School of Computing and Information Systems at Grand Valley State University. He received his Ph.D. in higher education/computing education from the University of North Texas. He has been developing and evaluating educational assessment instruments and curriculum in information systems for over 20 years. He served as Associate Director of Certification for the Institute for the Certification of Computing Professionals (ICCP) and co-created the Center for Computing Education Research. He has served as Secretary and Board member for the Association of Information Technology Professionals (AITP) Education Special Interest Group (EDSIG) and was named an EDSIG Fellow. He founded and is president of the Midwest Collegiate Computing Conference (MWC3) Foundation, a 501(c)3 non-profit whose purpose is to hold an annual student conference and promote regional collaboration between employers, universities, and students. His work has been published in *Journal of Information Systems Education* and *Information Systems Education Journal*. His work has also been included in proceedings such as AITP ISECON and EDSIGCON.

Information Systems & Computing
Academic Professionals

EDSIG
Serving Information Systems Educators

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 2574-3872