4

A common core for noninformatics majors

Focus group
Monique Grandbastien (F; chair), Richard LeBlanc (USA; rapporteur), Hubert Christiaen (B), Charles Duchâteau (B), Peter Juliff (AUS), Hajime Ohiwa (J), Harriett Taylor (USA), Eric van Ammers (NL), Piet van der Kamp (NL) and Roland Vollmar (FRG)

Abstract

Virtually all students in modern higher education programs should be exposed to informatics, not only in its applications but also in its core concepts. In this focus group paper we propose an introductory informatics course with six important themes, covering the full breadth of the discipline. The course can be implemented in various ways, depending on the audience. In any case there should be 'construction experience' and problem solving activity. This can be achieved through the conventional programming approach, but equally through alternatives that are more attractive for noninformatics majors.

Keywords

Informatics, other disciplines, noninformatics majors, curriculum (core)

1 QUESTIONS FOR CONSIDERATION

The question regarding a common informatics core for noninformatics majors can be translated in the following practical questions:

- what is the conceptual core of informatics?
- should their be a separate course or approach for science and engineering majors?
- who should do the teaching and what training should students have?

Informatics in Higher Education F. Mulder & T. van Weerts (Eds.)

- when is the right time for an informatics course?
- what tools should be used for practical work?
- what can students be expected to know before they begin an introductory informatics course?

All questions in the list above depend on the first question: What is the conceptual core of informatics? Therefore we have concentrated on developing a proposed conceptual core for an introductory informatics course. We strongly advocate that this core knowledge should be learned by both nonmajors and majors in informatics, since these fundamental informatics concepts are relevant to virtually all students in the modern academic world.

Thus, we recommend that most (or all) higher education programs include an introductory course in informatics.

2 PROPOSED INTRODUCTORY INFORMATICS COURSE

The core of the introductory course, proposed in this focus group paper, is intentionally described in general terms in order to cope with the many contexts in which it could be implemented. Among the various constraints that may have to be faced in implementing such a course, are:

- the inability to rely on any prerequisites;
- the limited duration of the course;
- limitations on the quantity and nature of practical experiences that can be included.

The proposed core consists of the following themes:

- information processing as a formal process;
- functional model of a computer system;
- concepts of computer-based communication;
- problem specification, modelling, representation, limitations and validation;
- potentials and limitations, fundamental as well as practical, of computer-based technologies;
- the computing perspective.

Information processing as a formal process

Students should develop an understanding of how informatics deals with formal processing (by programs) of digitalized information (data). A program provides a formal set of rules for manipulating information. Though information in the real world may take many physical forms, it must be coded as discrete values in order to allow formal processing by a program.

Functional model of a computer system
A student should be able to describe a functional (conceptual) model of a computer system which embraces the following.

There is a hardware component, including:

- a processor which executes instructions as directed by the software;
- an internal memory which is transient and represents an area where data may reside temporarily to be processed by instructions (a program);
- an internal permanent storage medium (hard disk) for permanent storage of software and data;
- a screen, keyboard, and other devices for user interaction.

and there is a software component, including:

- system software, such as the operating system, which controls and coordinates all of the activities of the computer;
- application software, which carries out tasks specifically chosen by users.

Concepts of computer-based communication
Since computer-based communication has become a pervasive way of interaction, it is important that students understand the conceptual workings of such communication systems, including how information goes from an input device to a computer, from a computer to an output device, and from a computer to (potentially) many other computers through networks. Simple notions of coding and decoding and of the need for standards to make communication systems work should be included.

Problem specification, modelling, representation, limitations and validation
Students should achieve understanding of the fundamentals of problem specification and modelling. These include both the designation of some set of tangible and observable aspects of the real world together with their relationships in order to describe a problem (specification) and the recognition of the information needed to describe the mechanisms held responsible for these relationships (modelling). Finally, issues of model representation and the limitations on model fidelity that are a consequence of discrete modelling should be included, along with the need to validate the meaningfulness of models.

Potentials and limitations of computer-based technologies
Computer-based technologies are founded in principles that induce limits to their applicability. The concept of an algorithm (rules for formal processing) embodies the range of potential applications and also provides a basis to understand the limitations of the technology, as expressed in our knowledge about complexity and computability. Students should be familiar with these concepts and be able to relate them to applications in their own disciplines. They should also understand where limitations on the performance of computer systems are of a more practical nature, such as the lack of success so far in creating computer systems that can

mimic human perceptual capabilities (vision, speech understanding) in any general way. The issue of correctness should also be introduced, as it relates to specification, modelling, algorithm design and implementation.

The computing perspective
The design of algorithms is a basic methodology not only for solving problems, but also for obtaining a basic scientific understanding of various phenomena, problems and processes. Recent attempts to develop information processing models of physical, chemical, biological, economic and social aspects of the world have shown such modelling to be an important new paradigm to attack problems that could not be handled adequately before.

Students should obtain experience with construction of algorithmic models so that they can understand and utilize this computing-inspired way of thinking about the world, just as previous generations shared a common understanding of mechanistic models.

3 CONSIDERATIONS FOR REALIZATION

Abstraction
The use of abstraction for complexity management is a pervasive technique in informatics. It is expected that students will be exposed to it and will apply abstraction repeatedly in the process of mastering the concepts described above.

Programming or not?
The most common form of introductory informatics course today is one in which students learn to program. A course that focuses primarily on students developing programming skills in a particular language will most likely succeed in transmitting major aspects of this core. However, it is by no means obvious that it will necessarily transmit the entire core. Conversely, we think it quite possible that a course based on this core might include no programming at all. We do believe though that such a course should include a 'construction experience', that is students should be required to get experience constructing formal problem solutions by doing one or more of: writing algorithms, creating spreadsheets, programming in visual languages, etc.

Implementations depending on the audience
As suggested by the previous point, we believe that courses based on this core can have a wide variety of implementations. The level of detail to which these concepts are developed and the methods used in the course might depend significantly on the audience (such as, informatics majors alone, informatics and other technical majors, engineering majors, nontechnical majors, business majors, or a mixed class). It is particularly important that the construction experience involves tools and problems appropriate for the composition of the class.

Attention for implications of various kind
There are numerous ways to include considerations of social, ethical and human-computer interaction issues related to information technology. For example, a discussion about why some applications are harder to use than others can call attention directly to interface design issues. That might lead to consideration of the implications of substitution of an automated system with a restricted interface (e.g. voice-mail) for a more flexible and responsive person (e.g. a receptionist). An introductory course should take good advantage of this opportunity to increase students' awareness of the relationship between these issues and technical concerns more naturally associated with an informatics course.

4 CONCLUSION

Informatics for all students and optional specializations
Fundamental informatics concepts are relevant to most (if not all) students in the modern academic world. Understanding of these concepts would provide a valuable foundation even for those students who will only be users of computer technology, particularly as modern applications provide increasing capabilities for user-level customization. Many students, however, will want to go beyond knowing only these fundamentals. Appropriate paths might include deeper development of modelling, design and programming skills in courses taught within an informatics department or development of discipline-specific informatics skills through courses in their own majors.

Informatics to be included in secondary school programs
There are many different situations regarding informatics courses in secondary schools. Because of the lack of consistency among these courses and their uneven availability to students, most introductory courses in higher education must begin from scratch. We recommend that secondary education moves rapidly toward inclusion of some informatics basics and experiences, based on our recommended core, on which higher education courses could rely.

Who should do the teaching?
Finally the question remains on who should teach the introductory informatics courses. For the introductory course as outlined above, instructors must clearly have informatics knowledge; specialized expertise in a particular noninformatics discipline is not enough. In addition, instructors must have sufficient conceptual background to help the students abstract the core concepts from particular examples.