A common core of concepts for informatics majors

Focus group
Doris Lidtke (USA; chair), Paul Myers (USA; rapporteur), Lillian Cassel (USA), Gopal Gupta (AUS), Anneke Hacquebard (NL), Fred Mulder (NL), Sigrid Schubert (FRG), Diane Schwartz (USA), Nobumasa Takahashi (J), Henk van Leeuwen (NL) and Stanislaw Waligorski (PL)

Abstract
The core concepts of informatics can be located in the intersection of the core concepts of the constituting and amalgamating fields of computer science, information systems, software engineering and computer engineering. In this focus group paper we present a core of both concepts and skills in eleven categories, ranging from information modelling, formalism and system design to social/ethical implications and (inter)personal skills.

Keywords
Informatics, informatics majors, curriculum (core)

1 INTRODUCTION

The broad field of informatics has increasingly been defined to be an amalgamation of the fields of computer science (CS), information systems (IS), software engineering (SE), and computer engineering (CE). Informatics may be seen then as computing in a very general sense. Yet though the term is in fairly wide use, there has been given no attention to those core concepts of informatics that would be essential for the development of relevant curricula. By its very definition, we see that the core concepts of informatics are in the intersection of the core concepts of CS, IS, SE and CE, respectively.

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

The following is an attempt to identify such a core. Of course it is worth pointing out that these core concepts are not identical, for instance, to the ACM Curriculum '91 (influenced by Denning's paper, 'Computing as a discipline'). For though the terms used there seem to be broad, the emphasis is actually on computer science alone. Here we share with Curriculum'91,however, the emphasis on core concepts and experiences that should be incorporated into the curriculum as a whole. The topics below are thus not intended to represent individual courses in any particular sequence.

2 CORE CONCEPTS AND SKILLS

The headings below are the broadly stated concepts and issues proposed as constituting the common informatics core in a very general sense. For each heading there are sufficient details and examples for clarifying and explaining the concepts. These indicate the sorts of knowledge and skills needed as a basis on which to build the 'whole person' who will become an informatician.

Representation of information

Perhaps the most fundamental concern of the field is information (hence the very term *informatics*). And whatever the domain of discourse, to enable manipulating (computing) and communication requires that the information be represented symbolically.

Examples are: symbols; various natural languages; sound, colour, etc.; bits and bytes; encryption and compression.

Information modelling

Manipulating information (or its representation) is important precisely because we obtain understanding of some situations or phenomena. Whenever anyone writes a program or designs a system to solve a problem, he or she is modelling the world (or a specific domain).

Critical to the discussion of information modelling is an appreciation of the complexity of the phenomena to be modelled. Such attention will reveal certain inadequacies and difficulties in the very paradigm of informatics: data collection, ambiguity, effects of policy and social parameters, information loss, etc. Other relevant topics include: abstract data types, object orientation, databases.

Formalism

Information modelling requires underlying formalism, i.e. the manipulation of *forms* that represent information about a domain.

Tools and symbol systems of particular importance to informatics include: discrete mathematics, logic, artificial language(s) and awareness that computing systems are vast compared with previous modes of mathematical modelling.

System design
The various fields of computing have always been involved with the construction of systems. Perhaps due to increasing awareness of engineering methodologies, more recent attention has been given to elements of good design (e.g. the emergence of the discipline of software engineering).

Most relevant in this area is the system life-cycle: without advocating a particular life-cycle model, attention should be given to requirements analysis, design, implementation, testing and validation as well as maintenance. Other important topics are: documentation, human-computer interaction, security, quality.

Algorithmics
It is a truism that computing is accomplished by algorithms and data structures. So awareness of key concepts in the design and analysis of algorithms is core to any computing discipline.

Pertinent knowledge and skills include: complexity (time and space), comparing algorithms, selection and design of algorithms, data structures.

Software development
Hands-on experiences in the methods of programming are taken to be core experiences, though they may be interpreted quite broadly and not strictly limited to standard programming languages.

Topics here include those very programming methods (e.g. programming paradigms, case tools, database systems, spreadsheets), modularity, reuse.

Capabilities and limitations of computing
It goes without saying that all computing practitioners need a keen appreciation of the capabilities of the computing paradigm: they will be adding to those capabilities during the course of a career. In addition the 'whole person' practitioner should be at least aware of limitations of the paradigm, from both theoretical and pragmatic points of view.

Examples include breadth and scope of feasible applications (perhaps achieved through student research projects and reports), noncomputability (the halting problem), infeasibility (the travelling salesman problem). We make a note here that there is no need to know these last two examples very deeply; they are often presented to educated lay readers without appeals to such deeper concepts as Church's thesis or NP-completeness.

Social and ethical implications
While often acknowledged as important for the computing professional, there never seems to be sufficient time in a curriculum to tackle concepts in these areas.

Possibly through seminars, case studies, role-playing, field trips and guest speakers, formal attention must be given to such topics as privacy, intellectual property, professional issues, access equity, codes of ethics, social change.

Computer systems and architectures
Core knowledge includes awareness and demonstrated abilities concerning the diverse parts of a computer system such as hardware, systems software, networks, embedded systems, tools, comparison and evaluation. Also included are important computer applications such as spreadsheets, databases, world wide web.

Personal and interpersonal skills
It has been stated that the era of the solo asocial programmer has come to an end. Through a maturing of the field, as well as the awesome complexity of the problems to be solved, effective team work has become crucial in the construction of the resulting extremely complex systems.

So skills are needed in such areas as communication, team work, critical thinking, leadership, working with users, interdisciplinary environments, written specifications and documentation, dealing with ambiguity.

Broad perspectives
Some knowledge and understanding may not be directly relevant to the design of a computing system yet is still considered as core to informatics. This would be similar to an acknowledgement that a solid grounding in the liberal arts is core to any educated person. One should be aware that informatics has a very rich and lengthy history (far predating electronics and involving some of the giants of the world's intellectual heritage) and exceedingly deep philosophical implications.

Relevant topics for this awareness might include: history, philosophy, artificial intelligence and cognitive science, theoretical foundations (see also capabilities and limitations of computing above).

3 CONCLUSION

The listing of the preceding core topics, concepts, and experiences (not constituting specific courses in a curriculum) has been influenced by the personal knowledge and experiences of the focus group members, representing the various subfields of informatics. But this list is not simply the reflection of current practice that is common to all those subfields. Indeed, in places we advocate topics that may not be part of the current curriculum or knowledge base of one or more classes of informaticians.

Thus we assume that some of the preceding may be quite controversial. But controversy stimulates further deliberation and discussion, and is the enemy of complacency. As ours is still a young, growing, and rapidly changing field, we might all agree that lazy complacency is to be avoided at all costs.