29

Teaching informatics as a subject

Peter Hubwieser
Fakultät für Informatik der Technischen Universität München
D-80290 München
E-mail: hubwiese@informatik.tu-muenchen.de

Steffen Friedrich
Fakultät Informatik der Technischen Universität Dresden
D-01062 Dresden
E-mail: sf2@inf.tu-dresden.de

1 INTRODUCTION

The following theses concern the teaching of informatics, which is synonymous with the teaching of the basic concepts of Information Technologies. We are not discussing the use of Information and Communication Technologies (ICT) across curriculum.

2 WHY SHOULD WE TEACH INFORMATICS?

Modern information and communication (particularly Internet and multimedia) technologies (ICT) are more and more dominating our private spheres as well as our social and working environments. They are changing the way we think, the way we talk and the way we watch our world. Shortly, they will be affecting the foundations of our society.

Therefore the schools are facing the task of enabling their students:

1. to make use of the technologies in economical and efficient ways;
2. to control and to judge the consequences of using technology.

Capacity Building for IT in Education in Developing Countries
G. Marshall & M. Ruohonen (Eds.)

The teaching of user skills meets some serious obstacles as there are:

1. a broad variety of different user interfaces and different systems often expect very different sequences of actions to perform the same work routine. It is not possible to train on all possible systems;
2. the design and operation of the systems is changing very fast; what fits today is not working tomorrow. Nobody can foresee which user interfaces will be the fashion in the coming year.

To control and judge the consequences of the rapid evolution of IT means to understand, at least in principle, the way IT systems and operations are constructed and the way they work because such understanding is equivalent to the ability to assess their inherent possibilities (Friedrich, 1995).

Thus we have shown that we have to teach the basic concepts of modern ICT. As a result the students should learn:

1. to build proper mental models of the systems that explain their peculiarities (Brauer and Brauer, 1989);
2. to look at the systems as universal complex tools that feel no emotions and suspect no mysteries.

This has to be done by teachers who are carefully and specifically educated in Informatics, which means that they have completed a special course of studies at university (Friedrich, 1996). This is impossible to guarantee in the case that Informatics is taught in other subjects.

3 WHAT SHOULD WE TEACH IN INFORMATICS ?

The raw material of all ICT is information. Every treatment of information follows the same process pattern:

1. The first requirement of any operation is a suitable representation of the information - a proper model of a real life situation or a proposed system is formed and described by adequate means; data structures are constructed and realized, and a message is encoded following the rules of an communication protocol.
2. Once there is such a representation, it is possible to change it (information processing) or exchange it (communication); a system model is improved by refinement or abstraction; data structures are changed by an algorithm that has to be constructed and encoded to make it runnable; datagrams are exchanged.
3. After (ex-) changing the representation has to be interpreted in order to produce information that is entirely new or that is not yet known at this particular place or by this particular subject; the system model is interpreted by a programmer; the state of a data structure causes some output on the screen; electronic mail is read.

The described pattern induces the range of contents we propose to teach - techniques and concepts of representation, processing and interpretation of representations.

Not every subject is teachable in schools: following Bruner (Bruner, 1960; Bruner, 1966; Schwill, 1997) we have to observe at least three principles:

- generality - the concepts should be applicable or observable in multiple ways and different areas;
- durability - what is taught today should be relevant at least for some years in the future;
- teachability - of course we have to take into account the students' abstraction level and specific abilities related to their particular age.

The result of these reflections is a set of concepts that form the skeleton of our proposed curriculum (for details see also Hubwieser and Broy 1996; Hubwieser, Broy and, Brauer, 1997):

1. Modeling techniques - precise descriptions of complex systems by diagrams concerning decomposition, subsystems, data flows, message sequences, states and transitions, causal relationships;
2. typical data structures as objects of information processing - atomic types, sequences, records and variant records
3. typical applications of the structures - representation of arrays, relations, graphs, trees and typical data structures of standard software;
4. fundamental strategies of problem solving - 'divide et conquer', top-down, bottom-up, algorithmic, rule-based, object-oriented strategies;
5. algorithms and their description through programming languages - atomic structures of algorithms handling sequences and trees, searching in networks (graphs), fundamentals of programming languages;
6. local and global networks - description by graph, typical topologies;
7. basic concepts of communication - addressing techniques , protocol stacks, services of computer networks;
8. synchronisation of parallel processes - common data structures (semaphores), monitoring, handling of deadlocks, problems of consistency;
9. assessment of representations, systems and solutions - borders of computability, basics of cost-benefit analysis, manipulation of data;
10. problems data protection - legal situation, management of data access, encryption techniques.

4 HOW SHOULD WE TEACH INFORMATICS ?

We base our methodical approach on the psychological concepts of cognitive flexibility (Spiro and Jehng, 1990) and cognitive networks (Anderson, 1985). Above all we demand to arrange the lessons around larger project sequences.

We propose to follow the process of modeling and simulation, using adapted methods of modern software engineering (Broy, 1995; Broy, Facchi, Grosu, Hettler, Hussman, Nazareth, Regensburger, Slotosch and Stoelen, 1993; Booch, 1994; Jacobson, Christerson, Jonsson and Övergard, 1991; Rumbaugh, Blaha, Premerlani, Eddy and Lorensen, 1991):

1. The teacher selects a problem to be solved or a real life system to be simulated. It should be as close to the students' experience of life as possible in order to illustrate the concepts. The complexity has to be high enough to prevent solutions by naive, intuitive means. This will motivate the students to learn and use new techniques.

2. The starting point of the real project forms an informal description of the problem. Hereby the students are forced to make their own reflections on the problem.
3. This step represents the most important learning process - we construct a (as much as possible formalized) model of the situation, using adequate description techniques.
4. Based on that model we construct a realization of our solution, a simulation of the system, aiming to illustrate and check our model. According to the students' interests this will be the most popular part of the project.
5. At the conclusion of the project we review all our work - the correspondence between informal description; model and simulation is inspected; the cost-benefit analysis is performed; the consequences of our solution are judged; and we think about possible improvements.

5 REFERENCES

Anderson, J.R. (1985) *Cognitive Psychology and Its Implications*. Freeman & Co., New York.

Brauer, W. and Brauer, U. (1989) Better Tools - Less Education?, in (ed. G.X. Ritter): *Information Processing 89*. Elsevier Science Publishers B.V. (North-Holland). IFIP, Amsterdam.

Booch, G. (1994) *Object-Oriented Analysis and Design*. Benjamin Cummings, Redwood City, CA.

Broy, M. (1995) Mathematics of Software Engineering. Invited talk at MPC 95. In Möller, B. (Hrsg.): Mathematics of Program Construction, July 1995, Kloster Irsee, *Lecture Notes of Computer Science* **947**, 18-47, Springer.

Broy, M., Facchi, C., Grosu, R., Hettler, R., Hussmann, H., Nazareth, D., Regensburger, F., Slotosch, O., Stoelen, K. (1993) The Requirement and Design Specification Language SPECTRUM. An Informal Introduction (Version 1.0). TUM-I 9311, Technische Universität München.

Bruner, J.S. (1960) *The Process of Education*. Harvard University Press, Cambridge, MA.

Bruner, J.S. (1966) *Toward a Theory of Instruction*. Harvard University Press, Cambridge, MA.

Friedrich S. (1995) Informatik-Didaktik, ein Fachgebiet im Aufbruc. *Innovative Konzepte für die Ausbildung*. Springer.

Friedrich, S. (1996) Ausbildung gefragt. *LOG IN*, **16**.

Hubwieser, P., Broy, M. and Brauer, W. (1997) A New Approach to Teaching Information Technologies: Shifting Emphasis from Technology to Information, in *Information Technology: Supporting Change through Teacher Education* (eds. D. Passey and B. Samways),Chapman & Hall, London.

Hubwieser, P. and Broy, M. (1996) Der informationszentrierte Ansatz, ein Vorschlag für eine zeitgemäfle Form des Informatikunterrichtes am Gymnasium. Institut für Informatik der Technischen Universität München, Technischer Bericht TUM-I9624, Mai 1996.

Jacobson, I., Christerson, M., Jonsson, P., Övergard, G. (1991) *Object-Oriented Software Engineering - A Use Case Driven Approach*. Addison-Wesley, Reading, MA.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W. (1991) *Object-Oriented Modeling and Design.* Prentice-Hall, Englewood Cliffs, NJ.

Schwill, A. (1997) Computer Science Education Based on Fundamental Ideas, in *Information Technology: Supporting Change through Teacher Education* (eds. D. Passey and B. Samways), Chapman & Hall, London.

Spiro, R.J. and Jehng, J.C. (1990) Cognitive Flexibility and Hypertext, in *Cognition, Education and Multimedia: Exploring Ideas in High Technology* (eds. Nix, D. and Spiro, R.J.), Lawrence Erlbaum & Associates, Hillsdale NJ.