Introducing Computer Science in an Undergraduate Program Based on Problem-Based Learning by Means of Declarative System Modelling and Simulation

Bengt Lennartsson, bln@ida.liu.se
Patrik Hägglund, patha@ida.liu.se
Department of Computer and Information Science
Linköpings Universitet
S-581 83 LINKÖPING, Sweden

Abstract This paper presents the objective for and experience from using Dymola as "the first programming language" in an engineering curriculum in information technology. This curriculum is based entirely on the Problem Based Learning model, PBL, and on integration of the contents from different disciplines and different department. The curriculum started in the fall 1995, so there is still less than one year of experience.

1. The general objective for the curriculum

The Information Technology Program at Linköping University is a new 4.5 year engineering program with three specific objectives:

1. To investigate and demonstrate the potential of Problem Based Learning as a general strategy for engineering education and training
2. To recruit 40% or more female students (around 5% is a common fraction in other engineering programs in Sweden)
3. To achieve integration of the contents over subject and department boundaries

In order to attract a higher fraction of female students the marketing material was carefully designed (images tested to appeal to women in particular, use of the principle of Noah; on every occasion where the program was discussed or presented there should always be one male and one female representative present), and the contents of the program was analyzed. One particular concern was about the introduction of computer science and programming such that no favor should be given to the (male) hacker and video game champion type of student.

2. The initial design of the introduction to programming

The design of the programming contents of the first year was based on the constraint that the girls who had applied and been accepted to the program should not be punished by contents or by the organization of the curriculum. The stuff presented should ideally be new to everyone, also for the (male) hackers.

At the department we were also a few persons who preferred an introduction to programming based upon a declarative view of computer programs. That is a system where the relational or equational constraints can be fed directly to the computer system, and where the algorithms for finding solution were hidden for the user, would be advantageous.

The integration over subject and department boundaries implied that the introduction to programming should be integrated with linear algebra and with elementary physics (harmonic oscillations, elastic collisions, etc.) Hence Dymola met the requirements and was a quite natural choice for the tool to be used by the students. If the models used in physics were of ordinary differential equation type, tate space models of the dynamic systems could fairly easily be defined in Dymola, and animation of the solution could be implemented without any particular prior knowledge of traditional programming.

3. The Problem Based Learning Strategy

The corner stones of the problem based learning model, PBL, are [1]:

1. Work based on group work in the tutorial group, and the seven step process (or sometimes nine steps; starting from a "situation", a "scenario", or a "case" via clarification of words and concepts, brainstorming, and structuring and grouping, on or more relevant problems are identified and designed, and specific learning goals are formulated (steps 1-5). Then it is time to follow the paths towards the goal and to evaluate the result (steps 6-7).

2. Work based upon scenarios from the real world. The starting point for the seven steps excursion above is a situation understandable by the students and relevant as a starting point or a template for the definition of the learning goals.
3. The path from the scenarios to the knowledge goals decided by the students themselves individually, but it is in general prescribed that the goal should be the same for all students.

Several assessments of PBL based curricula has been done, but there is no quite clear result about the outcome. A rather common interpretation is that the students from PBL programs are better in understanding totality and wholistic aspects, but they may be weaker in the details of a particular subject. It is in general assumed that PBL students are better motivated, remembers their findings better, and in particular they are happier and enjoying their studies much more than students from traditional programs, despite they have to work more and harder.

4. The outline of the Dymola ”course”

During the first semester the students met with three themes:

1. “Learn how to learn”
2. “The tools for the engineer”
3. “The engineer and the society”

Theme 1 was a five weeks introduction to library use, internet features, and to the PBL model as such (tutorial group, the seven steps, etc.)

The second theme, The tools for the engineer, has contents from mathematics (linear algebra), physic: classical mechanics and oscillations, and computer science: abstraction and evaluation of expressions; ([2] was one of the text books recommended in this area), regular expressions, context-free grammars, and finite state machines ([3] was the reference book for the students). The glue between computer science and physics was the definition of dynamic systems by means of systems of differential equations, the state space view, and solution and animation of the system by means of Dymola. The scenarios presented to the students were defined in terms of phenomena from physics: single and double pendulum, oscillating beams, particles moving in gravitational fields, etc. The students should do observations and measurements on “the real world objects” in the lab, try to describe the relations and dependencies in mathematical formula, and animate models by means of Dymola.

In Theme 3 the students were doing field studies in industrial organizations, with a focus on communication means and communication patterns in such organizations.

5. Implementation of the strategy

It was quite clear that we did not reach the goal the first year. The computers were installed late. The Dymola distribution for UNIX was still a bit shaky and buggy, and the students were not able to take the responsibility themselves to the degree we had (unrealistically) expected. Hence, the clue of theme 3, particle motion and collision in a rotating system, was not covered during the time slot available.

However the students were able to model a double pendulum in Dymola and to present an animation of the solution (34 out of 35 students passed this exam). The students also appreciated and were able to understand the guest talk by Hilding Elmqvist, where he illustrated the power of abstraction and object orientation in Dymola by presenting a complex animation of a model of the dynamics of a car.

6. Conclusions

It is possible to illustrate the principles of “declarative programming” to first year engineering students and to have them define state-space models of (very) simple dynamic systems (by declarative programming we mean the writing of equations and relations without having to supply the algorithms for finding the solutions).

For next year the same basic philosophy will be used, and hopefully the students as well as the teachers have learned from previous year and will improve.

Readers interested in experiences from PBL-use in general has a rich source in [4].

7. References

[1] David Boud, Graham Feletti: The Challenge of Problem Based Learning. Kogan Page Ltd. London 1991. ISBN 0 74940 249 0.

[2] Harold Abelson, Gerald Jay Sussman: Structure and Interpretation of Computer Programs. McGraw-Hill Book Company1989. ISBN 0-07-000-422-6.

[3] Alfred V. Aho, Jeffrey D. Ullman: Foundations of Computer Science. Computer Science Press 1992. ISBN 0-7167-8233-2.

[4] The PBL-LIST on INTERNET.
Send an e-mail To:
roger.hadgraft@eng.monash.edu.au
if you want to subscribe To: pbl-list@eng.monash.edu.au