Some thoughts on international projects in the undergraduate education

Mats Daniels¹, Anders Berglund¹, and Marian Petre²

¹Department of Computer Systems, Uppsala University, Sweden
²Centre for Informatics Education Research, Faculty of Mathematics and Computing, Open University, U.K.

Abstract
Projects have been used in many different forms in computer science education. This has lately also included running projects with participants from different countries. Pastics and Runestone are two international projects. A discussion of educational issues in relation to international projects is conducted with these two projects as examples. Using projects in courses are not without problems and some of the issues regarding coverage of “computer Science” are brought up.

Introduction
Education in computer science is in many ways still searching its form. The exceptional growth in the IT industry and the fast technological development are contributing factors as are the relative young age of the field. There is also a "tension" between theory and craft (skill/practice [Gör92]), as well as between research and teaching.

Projects have, in our opinion, a potential of being an excellent form for covering the whole spectrum from theory to skill including social competence and personal development for computer science students at university level. There are numerous examples of projects in higher education, most local to a department, some run together with a partner from industry [Ber98, Dek97, Jac98], and still fewer in collaboration on an international scale [Cle99, Mac99].

There is however not much done to evaluate how the project form satisfies the requirements for "good computer science education", whatever that is. Our aim is to shed some light on the latter and improve our understanding of our task as (computer science) education providers. These efforts are done within the UpCSERG group (Uppsala Computer Science Education Research Group). UpCSERG is a member of CSERGI (Computer Science Education Research Groups International) with the aim of promoting good practice within computer science education Research [Bra92, Den94, Gal96, Isa89].

This paper use Pastics and Runestone [Dan98b], two international collaborations, as examples in a discussion of pedagogical issues. The focus is on using international projects in the education, and especially looking at the peer-learning aspect. Many of the issues are generally applicable, but special attention is on how this education form can be beneficial for some genuinely computer science needs.

Pastics and Runestone
These projects are used to develop and evaluate how to incorporate international group projects into the undergraduate computer science curriculum. Both projects add new dimensions to student teamwork, requiring students to handle collaboration that is remote, cross-cultural, and linguistically challenging, as well as providing a basis for different forms of peer-learning.
Pastics
Pastics (Peer-ASsisted Teaching In Computer Science) is a collaborative project between the Department of Computer Science at University of Helsinki, Helsinki, Finland, the Institute Universitaire de Technologie, Département Génie Télécommunications et Réseaux, Université de la Méditerranée, Marseille, France, and the Department of Computer Systems at Uppsala University, Uppsala, Sweden.

In the project, Master students in Helsinki, taking a course in Computer Aided Learning Environments, teach Java to third year students in Marseille, taking a course in Network learning. The students involved are thus at different academic level. The project has been run as a pilot study during the spring of 1999.

The objectives are somewhat different at the different sites. Some goals are common, like the experience of international cooperation.

The main objectives in Helsinki were to learn to use Information and communication technology as a tool for learning and teaching, and to design such tools. Also the project work itself was an important objective.

In Marseille, where the students were taking an extra year in commercial issues of telecommunications, the experience of having learnt through the web (since the students were supposed to become technical salesmen of such tools) was important. Also, the experience to study in English was important, as well as, of course, the actual course content: Java.

The Course
The Finnish students designed the course, and also interacted with the French students as teachers or teaching assistants on a daily basis. The course that they designed was thus not of a self-study model, but instead designed for interactive discussion on the course material.

The main part of the course consisted of Web-pages, with explanations of important concepts, links to material on the web, and exercises. The material was scheduled on a daily basis, and was each day updated with solutions to the exercises and other modifications.

During most of the day, two Finnish students were on-line to answer questions and to discuss with the French, mainly through IRC (Inter Relay Chat).

Results
Although the data collected is not yet analyzed, it is clear that findings are encouraging: Both groups of students fulfilled their learning objectives in "Computer Aided Learning Environments" and Java, respectively.

There were some disappointments as well. Perhaps the most apparent was that the interaction between the two different groups was not as strong as hoped for.

Runestone
Runestone started with a prototype version running in winter 1998 with students at Uppsala University, Sweden, and Grand Valley State University, Michigan, USA, and continued with a full-scale version in winter 1999. The students in this project were all at roughly the same academic level.
The project's primary aim is to introduce real international experience into undergraduate computer science education in a way that has value for all participants. The students collaborate closely with their foreign counterparts using appropriate communications and computing technology to solve a given problem. Because the students come from different specializations within CS, they have different knowledge to contribute to the project. Problems are designed to cover the spectrum of backgrounds. Runestone's secondary aim is to identify effective support structures for remote international collaboration, encompassing strategies for communication, management, and technology use. Runestone will evaluate pedagogical and technical solutions for collaboration, will examine the costs, both in time and money, and will investigate how students learn in such a setting and what they learn.

In short, the Runestone project aims to:

• Give students international contacts and experience with teamwork with people from a foreign culture.
• Give students experience of collaboration with a group having a different educational back-ground.
• Encourage learning through peer-learning.
• Give students experience with the use of Information Technology in problem solving.
• Prepare students for the possibility of working in a foreign culture.
• Use the foreign experience to aid students in producing a superior product locally.
• Gain experience with use of new techniques in the running of a course.

Another goal is to create a well-organized setting with courses that, after the initially higher start-up costs, run at normal or lower costs. One example of cutting costs without compromising quality is the use of student peer-learning, which can reduce the demand for staff hours. Another example of cost cutting is that the costs for renewing the course can be distributed across the departments involved. An important side effect is that a close collaboration with staff at another university is beneficial to the staff involved, e.g. by providing insights into other departments and giving ideas for new teaching methods.

In carrying out the Runestone project, we will establish results that address the issue of transferability to other departments and institutions. A distinction will be made between domain-specific and general lessons, particularly with respect to the impact of international collaboration on group interaction and personal development, the extent of peer-learning, and the costs of using this form of education. For example, the project shall examine questions such as how much time is spent on becoming acquainted with new techniques for communication and in what ways (if any) using non-native language impairs learning.

Educational issues concerning projects
The fundamental question for using projects is whether - and in what respects - it is a good education form, meaning that:

• The syllabus is covered at least as well as through 'conventional' methods.
• The actual time the students spend on the course is related to the 'allotted' time.
• The time staff spend on the course is related to the size of the course and is comparable to other ways of delivering the course.
• The cost, apart from staff time, of running a class is not higher than other forms.
• The course contributes to the personal development of the students.
• The form is motivating to students.

Syllabus
The syllabus coverage issue is one of the main discussion topics with regard to project-oriented courses. This is due to the (relative) freedom the students have to complete their project and also a result of a focus on project management issues. It is important that the educational goals are clearly stated and followed up.

In the Pastics project the assignment to the Helsinki students is very open-ended. They can design the course to be given to the French according to their own ideas. One example from the pilot study is that the Helsinki students did not teach the networking aspects and graphic issues of Java as much as was expected both by the staff and the Marseille students. One possible solution is that the two groups discuss the course content more thoroughly before the Finns start to design the course. Another aim that fell short was usage of ICT (Information Communication Technology), a more diversified use of ICT was desired.

The project in Runestone is fairly complex and a specialization on different tasks is necessary in order to complete the project. This is a desirable feature for the course on the Swedish side, since it is mainly aimed to give experience in using already acquired knowledge in a real setting. It is a capstone course on the US side and similar arguments thus hold for the appropriateness of such a set-up. There are on both sides expectation that the students obtain a deeper knowledge, but it is not important in which of the different areas this happens.

Student time
Another potential problem with project-oriented courses is that the student spent far too much time on them. The authors have on several occasions heard comments from colleagues teaching project oriented courses like “it really got their attention, they worked day and night with this”.

The time studies conducted in Runestone and the interviews in Pastics show, to our satisfaction, that the students spent roughly the expect amount of time on the course, although in the Runestone project time was a bit ill-spread with a high load towards the end.

Staff time
Another relevant time measure is the amount the staff spend. It is also interesting on what they spend time. There is an expectation that the staff will spend less time than on a “normal” course.

Both Pastics and Runestone have benefited from reduced development costs, since the actual project was mostly developed at one site, Helsinki and Uppsala respectively. Although there was time needed to set up the pilot study, it is clear that staff time was gained in Marseille, since the actual teaching was made from Helsinki. At Helsinki the extra time for staff was moderate. The time spend with the students in Runestone have not yet been examined, but preliminary feedback from the teachers suggests that time has been saved.
Costs
Equipment?

Personal development
One important aspect of project work is that students get a chance to practice and become aware of their social skills, in short improve their personal skills. Cultural issues are both desirable and problematic in international projects. Experiencing and functioning in a new cultural setting is very rewarding and the students gain confidence in their abilities. This is however not easy and it might turn into a problem that will obscure other goals of the course.

The students in Runestone and Pastics noticed only a few cultural differences between the two groups. Some differences are:

- Educational background (e.g., lack of knowledge of C and use of functional programming)
- Age (in the Runestone case where the Swedes were older: 23-24 vs. 20).
- External obligations; the Americans and the French perceiving that they had more job and family obligations, although some of the Swedish and Finnish students respectively work as consultants, i.e. the groups actually worked under similar conditions.

Nevertheless, the students in both Runestone and Pastics were emphatic that culture was a non-problem; each group described their counterparts as being "just like" or "pretty much like" them. In Runestone, the staff observed the following difference: The American cohort was a collection of individuals, whereas the Swedish cohort worked in concert as a team.

There was in the Runestone pilot relatively little social interaction between the cohorts; the students felt that they didn't know their counterparts very well, and the project didn't help them to get to know each other. Some interactions would probably have been more efficient if the participants had known each other better. Social interaction - jokes and talk about personal topics - increased toward the end, during the hectic efforts to make the project fly. Yet, for each of the students, some part of the process or of their counterparts' actions or interpretations remained mysterious. The full-scale version had several examples of extensive social contact.

The social interaction was rather limited in Pastics. Some of the French students experienced the language as a problem. Even from the Finnish the interest for social interaction was limited, other interests tended to take up their time.

Motivation
Projects are considered to be efficient in raising the motivation for students. Students on international exchange, seem, in our experience, to be more motivated to do well compared with their time at the "home" university.

In both projects we found factors that enhanced the initial motivation of the students:

- The extensive international collaboration.
- The project was an 'experiment'.
- There was a project to do.

In the initial meetings in Runestone, some students stated that the real challenge was to make the group work as a team, and to demonstrate the viability of the experiment; others cited both the teamwork and the challenge of the project itself. There were however also factors that
decreased the motivation. During the project, motivation was neither constant nor evenly distributed; students cited differences in expectations, motivation, sense of urgency, time available, language skills, local cohesion (and hence local group dynamics), and technical skill, within the groups as one of the main problems. At times the awkwardness of physical separation and, mainly in the Runestone case, different time zones impaired student motivation and enthusiasm.

In Pastics we observed that the Finnish students judged that their commitment was stronger than the commitment of the French. To a certain degree this difference was due to cultural differences. For example, the concept of time is slightly different at the two sites, a fact that might have been interpreted as a lack of commitment.

**Features of International Projects**

*Communication is central*

Communication technology and how it is used are important factors for the success of an international project. Different perceptions about motivation between the sides are influenced by the use, or rather lack of use, of communication technology.

ICT is a priori an essential ingredient in computer science education at university level, since it is both the study object and a tool for learning. The key issues facing the future computer science professionals involve globalization of the knowledge base, and increasing specialization and distribution of expertise with resulting need to collaborate in a culturally and linguistically complex environment.

The question in Pastics is, in what ways students, who take part in international collaboration as a part of courses in computer science, experience their learning and collaboration.

The American students in Runestone didn't perceive communication as a problem, while the Swedish Runestone students as well as the Finish Pastics students identified communication as one of the biggest problems. All Runestone students were frustrated by slow or lacking responses to email messages and IRC questions. The students cited multiple missed deadlines as a major problem, although they argued that this might not have happened had the communication been really effective.

Language per se was not a barrier for the students in Runestone. The Swedish students are highly competent English speakers (with 8-9 years of study and English usage required in many university courses), although they are not necessarily fully confident. The students' email and IRC logs are full of jokes - but the students expressed low confidence that their jokes were understood. Everyone was fiercely polite.

In Pastics, language was maybe the single most important problem. Some of the French students claimed that their knowledge in English was too weak for keeping discussions through IRC with the Finnish students. This group of students reacted by withdrawing from the conversations over the net, and mainly working on their own.

*International exposure*

Measures should be taken to ensure that the students actually experience international collaboration. This has not been done explicitly in either of the projects, but so far there seem to have been a good international distribution of responsibilities.
Natural setting for peer-learning

Based on anecdotal evidence from our own experience as teachers and preliminary findings from data collected, we believe that having students explain concepts and solutions to one another is a powerful learning technique. Our conjecture is that there will be plenty of occasions for the students involved with the project to help each other with activities such as explanations, clarification, sharing knowledge or rehearsal of ideas. Occasions for peer-learning can be formal or informal. Formal occasions arise when students at site X present information for the students at site Y. Informal occasions include questions that arise during day-to-day e-mail or simple study sessions.

Both projects will systematically examine peer-learning by considering which settings tend to encourage, or discourage, peer-learning as well as factors that contribute to the effectiveness of peer-learning in these situations. One of our hypotheses is that the rather different educational backgrounds of the two sets of students involved in the project will encourage peer-learning. The differences in backgrounds should motivate the students to articulate their reasoning, rather than assuming that there is mutual tacit understanding between them and their foreign counterparts.

In the pilot run in Runestone peer-learning between the cohorts was limited; it was largely related to craftsman skills, e.g., better technical solutions. This may be accounted for by the lack of familiarity between the students and possibly by the nature of the project, which could be sub-divided in a way that avoided the need to learn about what the others were doing. Some of the Swedes reported peer-learning within the Swedish cohort, but this occurred largely in face-to-face interactions about which no data was collected.

Focus on computer science

Many pedagogical issues about using projects, and especially in an international setting, in our education are about education in general and not particular to computer science. However, we are computer science educators and as such we have special desires about the learning outcome of the project. Defining what students should know and ways to build learning situations where this can be accomplished is part of our effort in the computer science education research area. This area of research is cross-disciplinary and involves for instance finding out what the students have learned and how they study. A fuller coverage of this can be found in [Dan98a], where for instance additional reasons for conducting such research are pointed out, e.g. that it can improve motivation among 'research-oriented' teachers and increase clarity among 'teaching-oriented' teachers.

The effort to marshal the latest technology and to implement the latest teaching methods to serve educational aims must be balanced with the need to seek out and address questions about which concepts, strategies, and techniques are fundamental to a computer science education. The fast pace of technological change poses a double challenge for computer science education: developments affect both the subject and the mechanisms of teaching. Educational methods race to keep pace with the opportunities afforded by technology. We must understand 'what Computing is' in order to teach it - we must marshal appropriate tools and methods to teach it well - and what we teach will influence what computer science becomes. This requires that research looks deeper than merely evaluating implementations, deep enough to examine what changes in teaching practice reveal about underlying issues such as concept acquisition, development of skills and expertise, sources of misconception and superstition, learning processes, the roles of different types of interaction between
teachers, students, and materials, and so on. We need to know not just the effect of introducing new technology or methodology, but also the price.

What distinguishes teaching in computer science?
In computer science, the fast pace of change is not just technological, but also intellectual and methodological. The discipline of computer science, without a firm traditional underpinning or a firm educational tradition, is buffeted by changing definitions of the domain itself. The academic discipline is characterized by many tensions: between science and engineering; between theory and practice; between training and education. Tensions are exacerbated by the current climate; in the face of income-oriented institutional perspectives, the push to satisfy future employers, the competition for students, and so on, the tensions are a matter of continual debate. Hence, the discipline is characterized by an almost unmanageable diversity:

- **Academic perspectives:** Degrees and courses in computer science cover a wide range of goals and values.
- **Representation systems:** Changes in notations and programming paradigms are attended by the need to comprehend and have competence with more than one.
- **Technical context:** Our artefacts must be understood in the current technological context. This diversity and the pace of change mean that, not only must we provide students with a solid foundation, but we must also equip them for continual learning subsequently.

More fundamentally, the nature of what we study - that our tools are also our objects of study and are also a means of teaching - sets computer science apart:

- The objects of study, our artefacts, are abstract and difficult to observe. They are different from physical artefacts (e.g., what does a compiler look like? or an operating system?), although they interact with objects in the physical world.
- Concepts and artefacts encompass many levels of abstraction and complex inter-relations.
- Our artefacts are dynamic; we must reason not just about their properties, but about their behaviour (potentially complex behaviour) in time.

In this context of abstract, difficult to observe, dynamic, interacting objects of study, it is a particular challenge for educators to make theory concrete - without confusing technology with theory.

One challenge in computer science is that the constraints to thinking within the discipline are not physical, but human: our artefacts are constrained primarily by our ability to invent. Hence computer science teaching is about what computer scientists have managed to think about so far, and in what manner: algorithms, paradigms, languages, engines, tools, solutions are all thought products. But they are thought products that interact crucially with the physical world, and the relationship between the reasoning discipline of computer science and its technology is central to its particular character.

Conclusions
Pastics and Runestone are two different examples of projects run with international collaboration. Experiences drawn from these projects are often valid for students within other subject areas as well. Projects are important for computer science students since the form provides an excellent opportunity to practice their trade, to gain confidence in their craftsmanship. Their maturity in the use of computers makes them ideal guinea pigs with a high likelihood of success. There are however many pitfalls, and more efforts should be directed towards how to use projects. These efforts should be conducted in a scientific manner.
and establishing centers for computer science education research and collaboration among them is vital to this task.

References


