Rationale for the Design of a Web-based Programming Course for Adults

Jens Bennedsen & Michael E. Caspersen
Department of Computer Science
University of Aarhus
DK-8200 Aarhus N, Denmark
{jbb, mec}@daimi.au.dk

Abstract:
Web-based distance education is becoming more and more popular in particular for part-time educations primarily because it flexibly adapts to a busy life where family and work has first priority. Students can follow the course without wasting precious time traveling to and from campus several evenings a week, and participation can take place also in situations where it would otherwise be impossible (e.g. a sick child or a business trip). However, these advantages don't come for free: the students lose the social relations following from face-to-face contact with the lecturer and teaching assistant as well as the other students. Another serious drawback is the reduced bandwidth in communication among the different actors. Precautions have to be taken in order to compensate for these and other drawbacks.

We describe a web-based introductory programming course for adults and the rationale for choosing technologies and organizing the course in order to compensate for the drawbacks inherent in this kind of teaching. Our conclusions, although presented in the context of an introductory programming course, holds for most other construction-based courses as well. In the first part we set the stage by describing the relevant context and identifying the special conditions stemming from the course being an introductory programming course. In the second part we present a thorough discussion of the consequences of converting to a web-based course. Based on these consequences, we discuss what is meant by flexibility and how to achieve it. We argue for our choices and define our goals for the transition to a web-based course. Finally we outline how we have planned to evaluate the efforts; the result of this evaluation will be published in a following paper.

1. Introduction

In recent years the demand for people with a degree in information technology has increased dramatically; consequently, the Danish government supports a number of initiatives to increase the opportunities for people with a bachelor degree of any kind to achieve a master degree in information technology. Some of these initiatives are part-time studies at Open University (see Undervisninsgministeriet 2001). In topic, the degrees range from computer science and software engineering (with different flavours) over multimedia/design to more business-oriented degrees on organizational aspects of developing and using information technology.

In general, courses are becoming increasingly web-based [1] (see Lewis 1999), and in particular for part-time studies at Open University there is a growing demand that courses are offered to allow for distance education. The Flexnet project has been established among four Danish universities to support the development of distance education for part-time studies at Open University. The goals for the project are to develop flexible, modular online courses of the high quality (i.e. without changing the contents or decreasing the knowledge of the subject due to the transition to online courses).

Flexibility is an important issue when organizing online courses. In the Flexnet project flexibility is achieved by:

- reducing face to face encounters to a minimum
- developing small, cohesive and clearly defined modules
- supporting collaboration and learning with technology

[1] In this paper we use the term "web-based" indicating that most learning takes place mediated by the web. Others use terms like e-learning or distance learning. We prefer the term web-based since learning is neither electronic nor distant.

Among other things, quality is achieved by:

- choosing technological tools based on a didactical analysis
- awareness of the consequences of achieving maximum flexibility
- systematic evaluation and adjustment of activities

As part of the Flexnet project we are developing an introductory object-oriented programming course; the course has a very mixed audience since it is offered for master students from several different study programs (software engineering, multimedia, and ICT in organizations).

The typical student is male, approximately 35 years old and employed – very different from the average university student. Students at Open University have special needs and motivation factors which we have to take into account when organizing courses. Knowles' theory of andragogy (adult learning) attempts to differentiate the way adults learn from the way children learn (Knowles 1980). Two sources (Cantor 1992) and (Cranton 1992) sums up the key differences:

- adults are autonomous and self-directed
- adults are goal-oriented
- adults are relevance-oriented (problem centered) – they need to know why they are learning something
- adults are practical and problem-solvers
- adults have accumulated life experiences

The introductory programming course at Open University has been taught since 1998, but until now in a more traditional face to face version; the contents of the course is therefore well-defined. In this paper we will concentrate on the transition of the introductory programming course to a web-based course. We believe that the considerations and observations described in this paper are relevant not only for this course, but to most other course with a construction-based nature like this one (e.g. engineering, software development, and programming).

2. An Introductory Programming Course

In this section we identify the special conditions stemming from the course being an introductory programming course for a mixed audience.

Currently the introductory programming course is offered to students from the following three study programs: software engineering, multimedia, and information and communication in organizations. Software engineering cover a variety of topics on issues related to the development of large and complex IT systems (e.g. programming, computer architecture, formal languages, algorithms and data structures, human computer interaction, experimental systems development, and project management). Multimedia cover topics of both technical and aesthetic nature related to modern media types such as video, sound, 3D graphics, text and images (e.g. multimedia production, programming, hypermedia, human computer interaction, design, and aesthetics). Information technology and communication in organizations cover organizational and communicative aspects related to the use of IT (e.g. organizational analysis, human computer interaction, programming, media and communication). Of course, it is a major challenge for the common introductory programming course to address the different expectations from such a varied group of students. Consequently, the major ambition when switching to a web-based course is to put the student at the centre and organize the course material and the whole setting of the course in such a way that diversity is supported as much as possible, i.e. change it from a lecturer-centered to a student-centered course.

As mentioned earlier, programming is construction-based. This means that the students not only need to learn about the topics of programming but also about the process of *creating* well-designed programs (an analogy is if a student of a foreign language after very few weeks is also expected to be able to write fairly complex documents in that language). It is our experience that it is generally difficult for students to learn the programming process; this is particularly the case for students with a liberal arts background.

Teaching introductory programming to non-computer-science students and in particular to multimedia students with a liberal arts background is a big challenge for several reasons. Programming is not a primary interest of the students and many students consider programming to be "nerdy". Most liberal arts students are more inclined to "open-ended topics" in which analysis, discussion and interpretation are core competencies,

and are less inclined to take interest in "closed, absolute topics" like math and programming. Almost all students are lacking in mathematical qualifications, or even worse: many are scared of math and typically have had very bad school experience in that subject. Consequently, most students are de-motivated already at the outset and do not possess the habits, motivation and qualifications that other programming students do. For these and other reasons it is necessary to approach the task of teaching introductory programming to liberal arts students in a new and untraditional way. For a further discussion of this issue see (Andersen 2003). In particular it is important to motivate the need for programming by demonstrating what can be achieved; this suggests a top-down organization of the course where the need for details is motivated before being addressed. For other arguments in favour of this approach see (Bergin 2000).

Programming is construction-based by nature. According to (Bloom 1956), construction is at a relatively high level of competence and requires that the student is confident with the fundamentals of the topic. This indicates that the course should be organized bottom-up with the fundamental topics covered first and the more advanced topics covered towards the end of the course.

Apparently, there is a conflict in choice of organization of the overall organization of the course. We have chosen to solve this conflict by a spiral approach where fundamental topics and "advanced" applications of these are introduced hand-in-hand throughout the course.

Programming is radically different from all other activities in the sense that even the smallest inaccuracy is a fatal error; either the compiler fails to compile, or the program behaves incorrectly when being executed. Programming requires an extreme and unique precision in expression (Brinch Hansen 1996), and this is a requirement that many students, in particular those with a liberal arts background, find very difficult to adjust to. It is our experience that care has to be taken in the first weeks of the course in order to familiarize the students with these novel requirements. When changing to a web-based course, it is therefore important to ensure that the students are not left alone with their insecurity and frustrations. This is the prime reason we have decided on a relatively high density of seminars in the first part of the course as opposed to a more uniform distribution or a concentration towards the end of the course.

Programming being construction-based influences the educational model. Sfard (Sfard 1998) identifies two basic types of educational models, the Acquisition Model and the Participation Model. The idea behind the Acquisition Model is that learning activities focus on learning pre-specified knowledge and development of predetermined concepts. The focus of Participation Model is learning activities where the aim of the activity is that the students become members of a community of practice. Programming in this respect has many predefined programming rules, syntax etc and predetermined concepts, so the Acquisition Model should be the one in focus in the development of the course.

3. Converting to a Web-Based Course

Many roads have been taken when converting a face to face course to a web-based course. One of the roads we tried in the past was to digitalize the traditional course. By this we mean to take the traditional course and put it on the web simply by adding reading instructions for the course material, and maybe setting up a discussion forum, and then anticipate a good learning process for the students. This is what (Qvortrup 2002) call the "under-imaginative" IT supported learning model. Of course, we failed with this "business as usual" approach.

We will now discuss the implications of converting to a web-based course. Basically there are no new possibilities by using web-based learning instead of traditional face to face teaching. In a traditional course it is possible to use discussion forums for the students, use video-based materials, instant messaging to see which of the other students are online etc. But the change to a web-based course forces us to rethink and reflect on the course: What is taught when how is it taught by what material is it supported etc. This means that the ideas, teaching methods, materials, etc. also may influence the traditional course.

When switching from a traditional confrontation-based learning environment to a web-based, the communication bandwidth is drastically reduced. At first, this inevitably seems like a serious limitation, and different kinds of compensations is likely to be provided (e.g. video streaming of lectures). However, it is important very carefully to consider what to provide and what to omit. A video stream of the lecturer wandering around in the lecture room is hardly interesting, but the writing on the blackboard, the contents of the (PowerPoint) slide and the voice of the lecturer is essential to broadcast to the students.

The most serious drawback of web-based learning environments is that the social relations are very difficult to establish on the web (they can be maintained, but to establish is hard). Consequently, regular seminars where the students and teacher(s) meet are very important. In particular it is important to establish a good social atmosphere from the outset; mutual respect and sympathy among students and teachers is necessary

in order to stimulate collaboration and make it function properly throughout the course. Collaboration can take place without this kind of socialization, but it is far from ideal, and it makes it impossible for the sender to adopt communication to the receiving part. (Among many different answers to a question you have a much better chance of choosing the right one if you are familiar with the person posing the question.) Adoption to individual needs is always important, but it is particularly important in this setting due to the large diversity among the students.

Due to a long turn-around time for question/answer sessions, asynchronous communication (e.g. discussion forum and email) is a limitation compared to synchronous communication (e.g. telephone, chat room, video mediated meeting), but it has advantages too. Some students function well in a synchronous setting with quick and immediate communication; other students perform much better in an asynchronous setting which allows for reflection and thoughtfulness – activities that should be reinforced in every university course. Of course, both have their place in a web-based course.

When student and teacher are together, face-to-face, it is often the case that the teacher understands the problems of the student without the student having explicitly posed his questions. In a web-based course this kind of communication-without-words is impossible; the student must be able to formulate his questions in fairly precise terms.

4. Goals for a Web-Based Course in Programming

The main purpose of web-based learning is flexibility. "Any time, anywhere" is one of the buzzwords for web-based learning. In (Jain 2003) a study of motivating factors for part-time students participating in web-based courses were carried out, and flexibility ranks as the topmost motivating factor. There are many different factors in teaching and learning and each one can be more or less flexible.

But what is flexibility? In order to dig a little deeper we need to discuss the notion of flexibility: What kind of flexibility do we want to offer, for whom and why. In (Peters 2001) flexibility is characterized as follows:

> "*Flexibility has assumed four special meanings, which have become instrumental in university reform. First of all there is a focus on increased accessibility. Universities should be flexible enough to attract and to enroll new groups of students. Secondly there is a focus on giving students more choice and control over their learning processes. They should be allowed to learn what they want, when they want and how the want. Thirdly, an outstanding feature is helping the students to take responsibility for their learning. And fourthly, reformers are interested in meeting students' needs by providing more support than has been seen in our conventional universities.*"

This is a very general characterization of flexibility but it does not give any guidance in choosing what kind of flexibility to offer for a course. In (Collis 2001) five different flexibility categories are identified (and within these 19 flexibility parameters are enumerated):

- Flexibility related to time
- Flexibility related to content
- Flexibility related to entry requirements
- Flexibility related to course delivery and logistics
- Flexibility related to instructional approach and resources

We have focused on flexibility related to time, logistics, and instructional approach and resources:

- Time: We have increased flexibility by scheduling as few hours as possible; this increases flexibility both for the student and the teacher
- Logistics: The less teaching bound to particular places the more flexible – both for the student and the teacher
- Instructional approach and resources: We have increased flexibility for the students by supporting a number of different learning styles.

The main reason for choosing these flexibility factors is that the cohort is mainly adult part-time students with a

full workload. With these flexibility factors in mind; we have developed the following five goals for the transition to a web-based course:

1. As efficient learning processes as possible.
2. No compromise on the knowledge of the subject
3. As many asynchronous activities as possible (as opposed to synchronous activities)
4. As open a learning environment as possible
5. A scalable solution

In the following we will discuss each of these goals:

As efficient learning processes as possible
Students have different learning styles. In (Kolb 1984) four different learning styles are defined:

1. Concrete experience: being involved in a new experience
2. Reflective observation: watching others or developing observations about own experience
3. Abstract conceptualization: creating theories to explain observations
4. Active experimentation: using theories to solve problems, make decisions

We have addressed the different learning styles by having different exercises, different way of present solutions etc. The different learning styles are mainly supported by exercises and different kind of materials (i.e. the same topic presented in a book/article or by a video).

No compromise on the knowledge of the subject
One way of succeeding (at least with respect to the number of students passing the exam) would be to decrease the demands on the knowledge of the subject. We do not want this. This is expressed in the fact, that the description of the outcome for the course is the same now as it was before.

As many asynchronous activities as possible
Time and place are one of the two flexibility factors, so it is obvious that this is a goal. On the other hand, some things are better learned by some students in a synchronous setting (where you for example have the possibility to debate in a rapid pace).

As open a learning environment as possible
Students can learn a lot from seeing solutions, discussions, reading notes, etc. of other students. Consequently, we want to create a collaborative learning environment where as much communication as possible is brought out in the open. One of the advantages of web-based mediation of the communication in the course is its digital nature. This makes it possible to save communication and make it accessible for all students.

A scalable solution
It is clear that our first goal, efficient learning processes, can be achieved by increasing the amount of teacher resources; however, this is not an option. On the contrary we have as goal not to increase the amount of teacher resources.

5. Initiatives Based on the Goals

The primary reason to convert to a web-based course is to increase flexibility; in (Collis 1998) a concrete description of activities and tools to be used in a construction-based course (the Acquisition model) is described in order to enhance flexibility. Based on Collis' description for flexibility, we have chosen how to mediate the activities identified by Collis. Our choices for mediation of activities are captured in table 1.

Activity	*Mediation*
General course organization	All information on the WWW
Lectures/contact sessions	*Net mediated-meeting*: An on-line meeting where the teacher and the students can communicate in almost real time, but they can be in different locations. This communication can use text (chat), audio (voice chat) or video. *Video based materials*: The text-book is great for describing concepts, but is much less ideal for describing processes like the process of programming.
Self study and exercises; practical sessions	*BlueJ*: Use of a freeware programming environment designed for teaching purposes.
Multi session projects or activities	*Instant messaging*: It is possible to see which of the other students is on line right now. You can communicate with the other on-line students (chat functionality).
Testing	*Enhancement of the quantity of materials*: Students shall have the possibility to share all types of electronically documents including programs, notes, and solutions to exercises.
General communication	*Discussion board*. It is possible to post messages and other can reply to these messages. The messages and their reply(s) are threaded.

Table 1: Mediation of activities

Course organization and technology support
The course is organized as 15 weeks. Apart from the time frame, a week is defined by a topic and a mandatory exercise that the students submit towards the end of the week. Every week there is an online web-based meeting where specific questions and issues are addressed by the lecturer; the mandatory exercise from the week before is discussed too. Table 2 outlines the weekly schedule of the course. The evaluation form is an oral exam with a computer where the student presents and modifies a minor programming project from the course. This evaluation form is chosen in order to resemble the ordinary working mode from the course because we find it important to test the students in a situation as close as possible to their normal setting when working with the material from the course.

	Activity for the students	*Activity for the teacher*	*Technology*
Friday	Hand in the mandatory exercise.	Present the topic, exercises and mandatory exercise.	WWW, video based material, book, notes
Saturday - Sunday	Work with the topic and exercises.	Write comments to last weeks exercises.	BlueJ, video, instant messaging
Monday	Post questions to the topic and exercises.	The teacher browses through the questions. Some of the questions have an easy answer; this is given directly using the asynchronous channel. Other question can be answered by producing a video. It is expected that the students see the video and they have the possibility to ask questions to it during the synchronous meeting. The third kind of questions are questions that are very concrete and about the topics of the week. By abstracting over the questions, the teacher can go through the topics. This can either result in a video or as the starting point for the synchronous meeting.	Discussion forum
Tuesday	Synchronous meeting	Lecture on the questions, exercises, topics, ...	Streaming video
Wednesday - Thursday	Work with the topic and exercises		Discussion forum, instant messaging

Table 2: Weekly schedule

Comments/observations
We are well aware that this very strict timetable is in opposition to the goal “as many asynchronous activities as possible”. The reason for choosing it is two fold: Firstly, the students are adults, part-time students with a busy life. From other courses we know that they have a tendency to postpone their study work. The implication is

they give up half way through the course. Secondly, the nature of the topic (construction-based) requires a lot of regular practice.

Each week the students are expected to hand in a programming assignment. This assignment gets much attention from the students, so problems with this are often debated in the meetings. Organizing the meeting around the student's problems makes the students feel ownership of the problems being discussed. This ownership increases the motivation of the students and is far better than some peculiar problem the teacher has made up. The role of the teacher is to abstract over the questions and give hints to the theory, thereby giving examples of application of the theory on real problems.

The synchronous meeting is a video-mediated meeting. The students can see (part of) the screen and hear the voice of the teacher. For questions and two-way communication we use a text-chat.

The meetings are streamed and saved. This enhances the time-flexibility, but lowers the learning out-come of the students not participating in the meeting. After the meeting an index of the video is created for easy access to the different sections.

Two persons organize the meeting: a teacher and a secretary. The role of the secretary is to take care of all the administrative duties (inviting students to the meeting, keeping a time/concept log, keeping a log of who is online and when, etc.).

At the start of the course each student signs up for an instant messaging service. This is to give the students a virtual 24 hours a day 7 days a week "classroom" where the students can meet and help each other. It requires some socializing before students will collaborate with other students and ask/answer questions. We have a 2-day face-to-face seminar at the beginning of the course where we addressed the social aspect.

Traditional text books focus on concepts but not on the process of creating a program. Textbooks only present the final solution, but using the video media we can present the development process which is extremely important. As mentioned in section two programming is radically different from all other activities in the sense that even the smallest inaccuracy is a fatal error; this causes many students to lose their self confidence when they start learning programming. Through the unfolding of the development process the students realize that making errors is the rule more than the exception.

6. Evaluation

In section four we defined the following goals for the course:

1. As efficient learning processes as possible.
2. No compromise on the knowledge of the subject
3. As many asynchronous activities as possible (as opposed to synchronous activities)
4. As open a learning environment as possible
5. A scalable solution

Since the course is not over yet, we have no evaluation to report, but a thorough evaluation has been initiated. In this section we will describe the evaluation set-up and the rationale behind it.

In order to give a better evaluation of the first goal, we have given the students a test on their learning style, and will try to answer questions like "does the set-up favour particular learning styles?", "does particular materials favour particular learning styles", etc.

The second goal is rather trivial – the curriculum for the course has not changed.

The first, third and fourth goal concern the students. During the course we have asked the student to log their activities. This will give us quantitative data about the students' use of the systems, materials, etc. and help us evaluate the success of supporting learning with technology and materials. Furthermore we will give a questionnaire at the end of the course, interview selected students and compare the results of the students with the results of the students from the previous face-to-face version of the course. All this will make it possible to evaluate these three goals.

The fifth goal is evaluated by having the teachers fill in a weekly log describing their time consumption. At the end of the course we will compare this with the usual time consumption. We log the different activities so it is possible to which activities consumed what amount of time, and thereby hopefully give some ideas to improvements for the teachers.

7. Conclusion

We have described a web-based introductory programming course and the rationale for its didactical design. We have presented a thorough discussion of the consequences of converting to a web-based course. In particular we have discussed the notion of flexibility, and based on this discussion we have decided upon three categories of flexibility on which we have based our didactical design. >From the flexibility categories we have derived five specific goals for the new course design. Based on these goals we have chosen technologies, materials, schedules, study forms, etc. for the course. Finally we have outlined our plan for the evaluation of the efforts.

8. Acknowledgement

We thank IT-University West for financial support.

9. References

Andersen, P. Bøgh et al. (2003). Teaching Programming to Liberal Arts Students — a Narrative Media Approach. To appear at *ITiCSE 2003*, Thessaloniki, Greece.

Bergin, J. (2000). Fourteen Pedagogical Patterns. *Proceedings of the Fifth European Conference on Pattern Languages of Programs*, July 5-9, 2000, Irsee, Germany.

Bloom, B.S. (Ed.) (1956). *Taxonomy of educational objectives: The classification of educational goals: Handbook I*. Cognitive Domain. New York: Longmans, Green.

Brinch Hansen, P. (1996). *The Search for Simplicity, Essays in Parallel Programming*. IEEE Computer Society Press, pp. 167-168.

Cantor, J. A. (1992). *Delivering Instruction to Adult Learners*. Toronto: Wall & Emerson.

Collis, B (1998). New didactics for university instruction: why and how? *Computers & Education*, 31 pp. 373-393.

Collis, B and Moonen, J (2001). *Flexible Learning in a digital world*. Kogan Page.

Cranton, Patricia (1992). *Working with Adult Learners*. Toronto: Wall & Emerson.

EU (2000). *Memorandum of Lifelong Learning*. Brussels: The European Commission, SEK 1832.

Jain, K (2003). Motivating Factors in E-learning – A Case Study of UNITAR, *Student Affairs Online*. vol. 4 no. 1, Winter 2003.

Knowles, M. S. (1980). *The modern practice of adult education* (revised and updated). Associated Press, Chicago.

Kolb, D. (1984). *Experiential learning*. Englewood Cliffs, NJ: Prentice-Hall.

Lewis L et al (1999). *Distance Education at Postsecondary Education Institutions: 1997-98*. National Centre for Education Statistics, Statistical Analysis Report, December 1999.

Peters, O. (2001). *Learning and teaching in distance education*. Kogan Page.

Qvortrup, L (2002). Det lærende samfund. In *Uddannelse, læring og IT* (in Danish), The Danish Ministry of Education (http://www.uvm.dk/pub/2002/uddannelse).

Sfard, A (1998). On two metaphors for learning and the dangers of choosing just one. *Educational Researcher*, 27 (2), pp. 4-13.

Undervisningsministeriet (2001). *IT arbejdskraft og – uddannelser* (in Danish). The Danish Ministry of Education (http://www.uvm.dk/pub/2001/it).