Learning to program in the context of an industrial simulation: A pedagogical experiment in C++ and its implications for curriculum development

Sylvie Ratté, Jocelyne Caron

École de technologie supérieure / Université du Québec à Montréal

1.   Introduction

Since 1995, our programming courses have been supported by web sites where students can find supplementary resources. Each site contains six main blocks: memo, theory, work, results, links and help. The organizational structure of each block (Figure 1) includes a general home page and specific links to additions that teachers consider relevant for their groups.
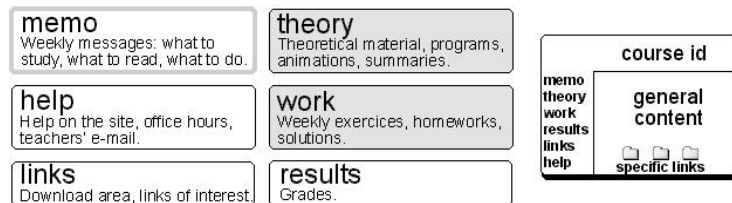

Figure 1. Structure of basic web sites

Contrary to what the structure might suggest, each block does not have the same size and importance. The theory block, with more than 1000 pages presented as rudimentary "hypertextbooks" [1], contains more than 90% of all course resources such as thematic documents, exercises, illustrations, animations, programs in various forms and summaries.

Because of this structure and the interaction system it supports, students require teacher guidance in using the available resources throughout the course. The web supports students with resources when they are outside the university. However, the overall organizational scheme of these sites does not correspond to a specific teaching method.

Last year, we adopted a new teaching method for an advanced programming course in C++. The goal of the method, derived from problem-based approaches, was to support students in programming a large-scale project that lasted the entire semester (similar in spirit with approaches suggested in [2,3]). It was obvious from the start that an independent site for the course would be needed to provide a posting and collaboration space centered around the project.

In this paper, we present our project by placing it in the broader context of web use to support an interactive and authentic learning experience. In this report, "project site" refers to the web site associated with this experiment, while "base sites" is reserved for the general sites introduced earlier (and outlined in Figure 1).

2.   Learning to program in the context of an industrial simulation

The course in this project is an advanced programming course in C++ covering the key concepts of the object-oriented paradigm. We designed this project with two main goals in mind (with similar ideas see [4,5]). First, to be authentic, it had to focus on problems that not only justify but also require the object-oriented paradigm. Second, it had to be presented in a realistic context. To reach these goals, students had to produce a large-scale project that consisted of simulating a factory. The project can be understood at two levels: the first is the problem of developing a discrete simulation of a factory; the second is the creation of the program itself which simulates the industrial context by requiring constant adjustment to new instructions and data.

We will first briefly describe the more specific goals of the project as well as the nature of the problem to be solved and its development. We will then present the criteria that were followed to develop the web site. Finally, we will show how they were put to use during the semester.

2.1   Specific goals

Based on common problems that students face when programming or resolving problems in general, we identified four objectives:

2.1.1 Step-by-step resolution

Our first goal was to slow down the students who want to jump into programming without planning. The context of the project required the development of a step-by-step solution. For this reason we did not supply a written formulation of the problem at any time during the semester. The problem was defined essentially from students' questions and their reaction to requests.

2.1.2 Expertise sharing

Our second goal was to require students to share their expertise and cooperate rather than compete against each other. So that they could benefit from each other's solutions. This was possible because of the method of evaluation and step-by-step approach that we adopted. The posting of the preliminary work on the web site as well as teamwork provided a way to realize this goal.

2.1.3 Relevance of theoretical notions

Our third goal was to introduce theory, exercises and examples when appropriate as students worked through their projects and the need for information arose. Considering that many students have difficulty making analogies, the idea here was to avoid the feeling that the material presented was only remotely related to the problem at hand.

2.1.4 Need to formulate and justify solutions

Our fourth goal was to get students to communicate by formulating and describing problems by themselves. Our sub-goal was to make them ask questions, learn to look for information and participate actively in classes and laboratory sessions. They also had to collaborate with a fictitious client to whom they were accountable.

2.2 The programming project

The students had to simulate the operation of an aluminum smelter. The simulation of factories is an interesting type of problem within a OO programming course because it offers a concrete example of all the covered theoretical elements (queues, stacks, message passing mechanism, etc.). The teacher can also easily adjust this complexity along the way according to feedback from students, as we will see.

Here is a diagram of the smelter the students had to simulate (Figure 2). The smelter consists of five main parts: electrolysis center, warehouse, anodes repair center, smelting center and cleaning center.
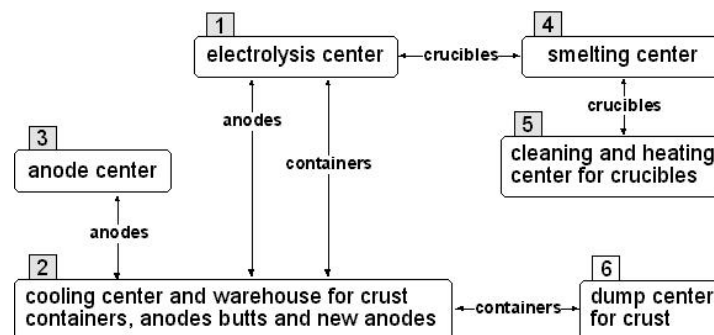

Figure 2. Aluminum Smelter

The main goal of the simulation was to estimate the possibility of increasing production. Consequently, students had to design their programs to help the client test different hypotheses: Does everything have to be doubled if production is doubled? Is the warehouse big enough to stock additional anodes and containers? Can the smelting center support the increase? What options are available?

2.3 Project phases

The student project consisted of four phases, which were roughly the same as those of a professional project: documentation, outline of a solution, prototype and delivery.
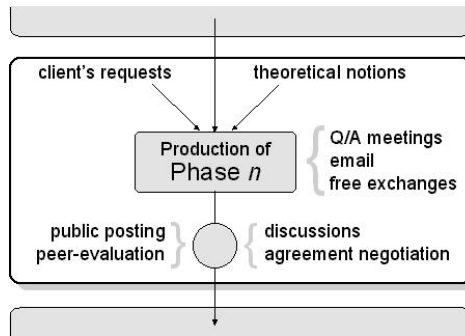
Figure 3. Development of one phase

Three types of information were available in each development phase: the client's input (through oral presentations or written requests), theoretical notions, examples and exercises seen in class and laboratory sessions, and the results obtained by the class from a previous phase (Figure 3). While working on a specific phase, students were also guided by question-answer periods in class, e-mail and discussions. Students could also communicate with the client by e-mail when necessary, being aware that a client can take a couple of days to answer. When justified, the answers were posted on the web site along with any synthesis or documents supplied by students themselves. At the end of each phase the teams negotiated and determined where to go from there.

2.3.1 Documentation

The goal here was to produce a detailed analysis of the factory. First, students had to acquire the terminology in order to be able to speak about the problem clearly. Second, they had to understand the general interactions between the sections of the smelter in order to produce an OO analysis for them.

2.3.2 Solution outline

Using the results obtained in phase one, students had to begin the construction of the program where the interactions between three essential sections of the smelter were simulated (Fig 2, sections 1, 2 and 3). The reduction of the problem to three sections was negotiated at a meeting.

2.3.3 Prototype

The goal here was to create a complete prototype (including section 4) where new requests to estimate the electrical consumption and costs of the factory and some modifications to the initial picture were incorporated. Also, the stocking method in the warehouse changed and section 5 was dropped by the client.

2.3.4 Delivery

In the last phase, the students completed the interface according to the client's instructions and the teams presented their product orally, covering both technical and non-technical aspects.

3. Web site development

The information from all sources, client, teacher, and students, was posted on the project web site. As a result, the site was so connected to their experience that the students quickly adopted it as their own.

3.1 Development criteria

Our own experience in developing web sites demonstrated to us that students participate more actively and perceive a site as being user-friendly when it is developed as the course unfolds. This enthusiasm results from three aspects of the project:

1. The available material is more relevant because:

▪ it is contextualized, so it relates directly to the students' learning experience;

▪ it is presented in small amounts at a time, thus helping students who do not know what to look for or cannot recognize that they found what they need.

2. Students feel they are contributing to the course development by supplying notes, corrections or additions.

3. Finally, students feel that the teacher works as much as they do and that consequently the site is really important since it is being built exclusively for them.
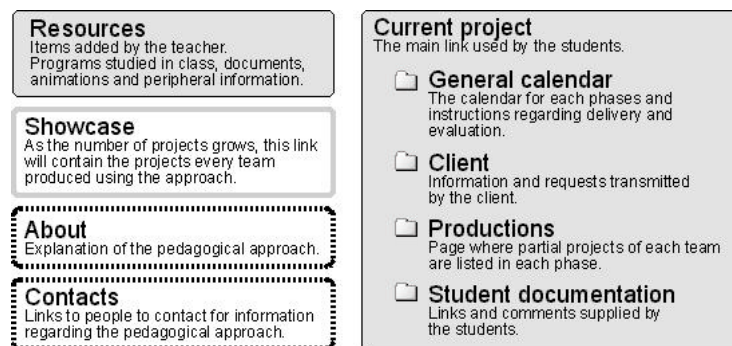

Figure 4. Main blocks of the project site

Therefore, the project site was built over an empty structure that was filled during the semester. In particular, two main blocks were anticipated: a place to find relevant theoretical resources and the grouping of all the information related to the program that students had to create (Figure 4, in gray).

3.2 Progressive nature of the project

In the first class of the semester, the client (the second author), who is not a computer expert, made a short presentation of the factory. This enabled the professor, the first author, to keep her distance and avoid supplying information which might have influenced the students' exploration

paths. The students were also introduced to the project site and the associated base site. They were then asked to use the project site to post the name of their team, the name of its members and leader, as well as a unique e-mail address. Each team could thus be contacted by everyone else who was associated with the project.

3.2.1 Adaptation of theoretical resources

The resources block was increased gradually with examples and documents covered in class. Exercises connected to the project were suggested every week. For example, the second week we simulated a barber answering customers in order to introduce the students to the concept of discrete simulation. In other words, theory was always discussed in light of the problem to be solved.

3.2.2 Classroom interaction

At the beginning of every class, a summary of the questions received by e-mail or otherwise was presented and we engaged in a discussion, "where are we". These short sessions allowed the teacher to orient the planned presentation. When students did not ask questions that could lead to the introduction of the necessary material, the teacher would present one or several theoretical notions (e.g. the implementation of a queue) followed by a discussion about its possible use in the context of the project. For example, near the beginning of the course a student asked about using a queue to manage the posts within the smelting center.

3.2.3 "client" input

We never explicitly announced that the client had uploaded new information on the site. This put the responsibility on the students to keep posted by regularly consulting the client block In this context, the input was used in four different ways:

1. to present the problem non-technically in order to avoid offering solutions or representations (as in phase 1);

2. to answer questions in a way that would illustrate their relevance (as in phase 1);

3. to introduce surprise modifications to counter those that [6] calls the "code warriors" (as in phases 2 and 3);

4. to make exaggerated requests to require students to estimate the implications for their code, learn to estimate time (as in phases 3 and 4).

3.2.4 Publication of preliminary work

After every phase, the programs or documents were published in the production block. The weaker teams were encouraged to borrow the code or elements of analysis of another team. To this end, the teacher met each team individually to check how work was going. At the end of the semester, all final products were posted in the showcase block. The class awarded virtual medals to the best teams.

3.2.5 Evaluation

Peer evaluation was also used in some phases (see [7] for other applications). For example, after phase1, every team had to evaluate the analysis of three other teams. They downloaded a small anonymous package on the web and filled in an evaluation form available on the web site.

After the delivery of the end-product at the end of the course, three different types of evaluations (besides the evaluation of the additions presented in phase 4) were used to test comprehension of the solutions. First, the teams had to answer a series of "what-if" questions. Then they presented their software orally first, as if addressing a group of users and then a group of professionals (an idea suggested also in [8]). Finally, students had an exam where they had to answer three questions directly related to their team program.

4.   Review and discussion

4.1   General results of the pedagogical approach

The project was a great success due to the interaction of a set of factors. First, the open nature of the input favored the sharing of expertise and strengthened the feeling that everyone can contribute, which led to a better learning experience. In a sense, it allowed students to model the behaviors of practitioners [9], but in a controlled environment. Finally, the project was also an enriching experience for the professor. It can thus be seen as another example of the benefit of problem-solving and collaborative learning approaches [10].

On the other hand, this type of project requires constant organization. Aside from class preparation, there are unexpected questions, preparation of relevant exercises, management of student queries, and constant updating of the site. It is impossible to plan more than a week or two in advance. Also, the multidisciplinary aspect of the course requires flexibility and support which the current university system does not encourage. The project could easily cover two or three courses including programming, oral and written communication, and statistics.

4.2   Discussion about the use of the web

4.2.1 The web as a collaboration space

From this experiment we can see that the web is very useful for sharing documents and we intend to further explore this use (see [11,12]). However, it has limitations and cannot be used efficiently in every collaboration situation. For example, how can we transmit via a forum or chat session all the richness of a class discussion? How can we immediately and efficiently pass on the brochure received by a team in an actual smelter? How can we show the rough drawing sketched on the board by a student who is frustrated by not being understood? These technological limitations will be solved, but at what cost and especially for whom exactly and for which purposes.

4.2.2 The web as a coherent space

After the project got underway, the authors had to rethink their use of both web sites. We were so used to using the base site to send directions related to class management (e.g. upcoming exams,

reading lists) that we did not realize at first that the students quickly adopted the project site as their own and therefore were a little bit confused by the double structure. Along the same line, we learned quickly that links to specific documents physically inside the base site should be shown inside the project site to make it more coherent.

This first experiment also underlines the need to present the secondary resources at different levels(e.g. the about and contact blocks). This is also the case for the base sites which have an organizational scheme that presents the class management information (e.g. memo, results, syllabus), on a par with the course material.

4.2.3 The web as a quality space

Although the web makes it possible to post unlimited amounts of information, a web site dedicated to a course should not be overloaded with resources that are not relevant to students' needs. We learned this very early in the project, as resources and links grew exponentially. Relevance, rather than quantity, is what students need.

5.   Conclusion and future direction

The use of the Internet and Information Technologies (IT) in general raises many questions about the appropriate way to structure given knowledge in building an autonomous environment dedicated to course contents. How can technologies help to structure knowledge? How does "putting a course on the web" affect the quality of the learning experience? Our current research attempts to answer these questions. There is no ready-to-use environment that will accomplish miracles. However, we do know that an authentic learning environment should immerse students in the contents and not simply guide them from one block or exercise or video to another. Research in education and human-computer interaction cannot be ignored.

IT are after all only technologies and not learning theories. To bring success to the largest number of students, we now know that a completely dedicated environment favoring the sharing of expertise, questioning and discovery is necessary. The integration of IT in education should be based on pedagogy and equality rather than the technologies themselves, attractive though they might be. The purpose of this project was in part to address these issues.

Web sites discussed in this paper

The project site: www3.sympatico.ca/bailaka/INFO/
Base sites related to C and C++   www.seg.etsmtl.ca/inf125/ and www.seg.etsmtl.ca/inf115/

REFERENCES

1.   Boroni, C. M., Goosey, F. W., Grinder, M. T. & Ross, R. J. in *ITiCSE* 65-69 (Association for Computing Machinery, Charlotte, NC, 2001).

2.   Bernat, A., Teler, P. J., gates, A. & Delgado, N. in *ITiCSE* 17-20 (Association for Computing Machinery, Helsinki, Finland, 2000).

3. Turner, J. A. & Zachary, J. L. in *SIGCSE technical symposium on Computer Science Education* 43-47 (Association for Computing Machinery, New Orleans, LA, 1999).

4. Boyle, T. in *1st Annual Conference of the Learning and Teaching Support Network Centre for Information and Computer Sciences* (University of North London, 2000).

5. Hadjerrouit, S. in *ITiCSE* 171-174 (Association for Computing Machinery, Cracow, Poland, 1999).

6. Weber-Wulff, D. in *ITiCSE* 85-88 (Association for Computing Machinery, Helsinki, Finland, 2000).

7. Gehringer, E. F. in *ITiCSE* 139-143 (Association for Computing Machinery, Charlotte, NC, 2001).

8. Pollock, L. in *ITiCSE* 287-291 (Association for Computing Machinery, Charlotte, NC, 2001).

9. Singer, J. D. & Willett, J. B. in *Symposium on Innovations in teaching: How Can We Improve the Communications Skills of Students? Annual Meeting of the American Stastistical Association* (San Francisco, CA, 1993).

10. Evans Sabin, R. & Sabin, E. P. in *SIGCSE* 304-308 (Association for Computing Machinery, Phoenix, AZ, 1994).

11. Hübscher-Younger, T. & Narayanan, N. H. in *Artificial Intelligence in Education Workshop - External Representations in AIED; Multiple Forms and Multiple Roles* (International Artificial Intelligence in Education Society, San Antonio, TX, 2001).

12. Zeller, A. in *ITiCSE* 89-92 (Association for Computing Machinery, Helsinki, Finland, 2000).