

## Use of the Information and Communication Technologies in a first course of computer programming

Ferran Prados, Francesc Castro, Joan Jesús Pujol

Departament Informàtica i Matemàtica Aplicada. Universitat de Girona  
Escola Politècnica Superior. Edifici P4. 17071 Girona  
{fprados,castro,jpujol}@ima.udg.es

**Abstract** — We present the experience of introducing the Information and Communication Technologies (ICT) in a first-year course of introduction to computer programming which has been held in the University of Girona during this year. One of the main guidelines of the European Higher Education Area (EHEA) is to promote the students involvement in the learning process. In this sense, we have started a series of teaching innovations to motivate the students and promote their self-learning. On one hand, we have introduced the e-learning platform ACME that allows the teacher to monitor the students work, and give the students personalized attention. On the other hand, we have incorporated the possibility for the students of writing their programmes in the same programming language used in the lectures, avoiding any translation to a standard programming language like Java or C. This issue is especially useful for the beginners.

*Keywords* – Computer Programming, Self-Learning, EHEA.

### 1 Introduction

#### 1.1 The European High Education Area

The Bologna Declaration [1], signed in 1999, establishes the basis for the new European High Education Area, that should be completed in 2010. The European countries that signed the Declaration undertook to reform the structures of their higher education systems in a convergent way.

One of the objectives included in such Declaration is to establish a compatible credit system in which the unit is the ECTS credit. Each ECTS corresponds to 25-30 hours of work for a student, including lectures and laboratory sessions, but also including the work “without professor”. Didactically, it involves changing the centre of the teaching-learning process, from the professor to the student.

#### 1.2 The Computer Science Degree in the University of Girona

The University of Girona (UdG) is one of the public universities in Catalunya<sup>1</sup>. The Catalan Government has started a Pilot Project to adapt the Degree Courses to the EHEA. The Computer Science Engineering Degrees in the UdG are involved in this Pilot Project. This fact has forced to introduce a series of didactic initiatives in order to improve these Degree Courses.

---

<sup>1</sup> Catalunya is a country that works as an autonomous region of Spain

### 1.3 Our Contribution

We present in this paper the initiatives we have introduced to the first-year subject “Methodology and Technology of Computer Programming” (from here on MTP), in which the basis of algorithmic and computer programming is introduced to the Computer Science students. We have aimed to improve the achievement of the competences for this subject by using the ICT’s, giving a particular importance to the Self-Learning and to the feedback between students and professors [2,3].

## 2 ICT Tools

### 2.1 POODI

The Professional Programming Languages (like C++ or Java, from here on PPL) are not usually introduced in the lectures of the courses on algorithmic and computer programming, but in the laboratory sessions. Instead, algorithms are written, in the lectures, in a pseudo-language more or less similar to the PPL but a bit less formal.

The philosophy behind the use of a pseudo-language in the lectures lies, on one hand, on the purpose of giving a generic view of computer programming to the students, independent of any specific language, and, on the other hand, on the purpose of releasing the students from the syntactic issues when designing the algorithms. Thus, the students write their algorithms in this pseudo-language (also known as pseudocode), and next they codify these algorithms in a PPL.

Some professors and students from the Department of Computer Science and Applied Mathematics from the University of Girona have designed a pseudocode with a strict syntax. Such pseudocode, known as POODI, allows Object Oriented Programming (OOP), and has been designed in order to make it easier the computer programming process for the beginners. Next we enumerate the main advantages of POODI in front of a standard PPL like C or Java:

- The **syntax is simpler** than in most PPL.
- **Syntax in Catalan**<sup>2</sup> (instead of English).
- **Clearer error messages**: they are written in Catalan, and, in general, they are less cryptic than in most PPL.
- It has “**learning-oriented**” semantics. This means that some “bad” or “untidy” programming customs, allowed in most PPL, produce semantic errors in POODI (in compiling time). This issue contributes to avoid further runtime errors.

A POODI compiler has been developed. This compiler allows to directly codifying in POODI in order to run and test the programs. Note that the students can do this without knowledge of any PPL.

### 2.2 ACME

The ACME project [4,5,6] has been developed, since 1998, in the Department of Computer Science and Applied Mathematics of the University of Girona. ACME happens to be an e-learning platform, whose acronym stands for “*Continuous*

---

<sup>2</sup> Catalan is the natural language for most people in Catalunya

*Assessment and Improvement of Skills*” in Catalan. The leading ideas of this project can be summarized in two main points:

- Communication via Internet, which allows the student to use the system from any computer with a standard Internet connection.
- Use of symbolic manipulation software to create exercises and correct them online. ACME is appropriate for different technologic and scientific subjects (mathematics, programming, chemist, etc.) [5].

The use of ACME aims for increasing the personalized attention to the students, promoting the self-learning, obtaining a fast feedback, and allowing the continuous assessment by the professor of the students’ progress. All these issues have the main goal of improving the teaching-learning process. ACME has been used in different subjects of the UdG during the last years, obtaining very promising results.

We have to remark that the system has been designed with a modular structure which makes it very flexible and adaptable to different subjects. This flexibility also allows adding new didactic tools.

ACME is very appropriate for the subjects of computer programming. The programming module is able to validate code written in any programming language. The MTP professors decided to introduce ACME in their subject in the year 04-05, motivated by the good results obtained with ACME in several introductory programming courses [6].

This year 05-06, the POODI compiler has been incorporated to ACME, allowing the MTP students to have an efficient self-learning tool. This integration has been so easy thanks to the ACME modular distribution. We have to note that there existed a pseudocode compiler in ACME [6], but this “old” pseudocode, appropriate for other computer programming subjects, had some limitations for MTP.

### **3 Our Experience**

#### **3.1 Teaching-Learning Methodology Involving the ICT Tools**

We have introduced the ICT tools described in the previous chapter to the first-year subject MTP. As previously described, this subject is an introduction to computer programming.

In our opinion, practising is the best way to learn computer programming. This statement was considered when we designed the year 2005-2006 for the subject MTP. Let us describe the teaching-learning methodology used during this year, insisting in the ICT tools previously described.

##### *Activities in the classroom*

The MTP course is divided in lecture and laboratory sessions.

**Lectures:** 3 hours per week (along 30 weeks approx.).

We have introduced here the main concepts on algorithmic and computer programming. Many examples have been developed on the blackboard. This year we have left the students to have a more active role. Thus, in part of the lectures time the students, arranged in groups, have carried out exercises with the professor’s

advice. Such exercises have permitted the professor to receive feedback about the evolution of the learning process. Moreover, the professor has corrected, marked and gave back the exercises to the students, who, in this way, have also received an appropriate feedback. POODI has been used in all the work developed in the lectures along the year.

**Laboratory sessions:** 2 hours per week (along 26 weeks approx.) in a computer lab, with 20 students at maximum.

Each week, the professor has introduced one or more exercises, which basically are computer programs to carry out. These exercises have been incorporated to the ACME platform. After some discussion about the exercises, the students have worked on them during the rest of the session.

In the first 8 lab sessions of the year, we have exclusively used POODI as programming language. This has been an important novelty regarding to the previous years. The PPL, in our case Java, has been introduced in the 9<sup>th</sup> session, when the students were more confident with the programming concepts. This means that, during the first lab sessions, the students have not had to mind about the Java syntax problems. Thus these sessions have been totally devoted to the algorithmic issues, the most important in this subject.

#### *Work at home*

The students have had to finish on their own the exercises set in the lab sessions. The ACME platform has had an important role in this issue. Let us describe how it has worked:

- The students send their programs to ACME (the first weeks written in POODI, and the rest written in Java, note that the POODI compiler has been added to the ACME platform).
- ACME gives the students an immediate feedback about the correctness of the solutions sent. The students can modify their code and send it again as many times as needed, until ACME considers it right.
- The professor can supervise all this process, and, if necessary, can write down, via ACME, advises for each of the students (or for all of them) and for each problem to solve.
- ACME allows the professor to monitor the progress of each and all of the students along the year.

A part from the lab exercises, we have provided additional non-compulsory ones, in order to increase the self-learning. It is noticeable that the students, since the first day of the year, have been able to do at home the exercises set in the lectures, using the language POODI and the POODI compiler provided.

#### *Tutorship*

The students have been continuously encouraged to use the tutorship time, visiting the professor's office (individually or in small groups) in order to make it easy their learning process. This year, each student has had to visit their lab professor to comment in situ the solutions send to ACME once revised and marked. Although a first impression of the correctness of the solutions is given by ACME, and although ACME allows focusing the feedback between the professor and the student, we have considered appropriate to promote a direct contact, aiming, on one hand, for getting

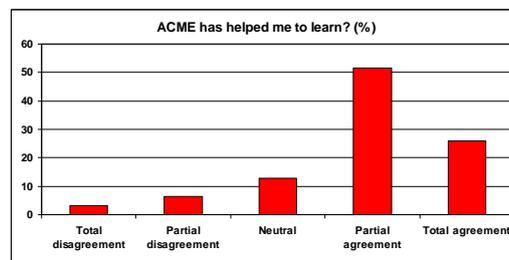
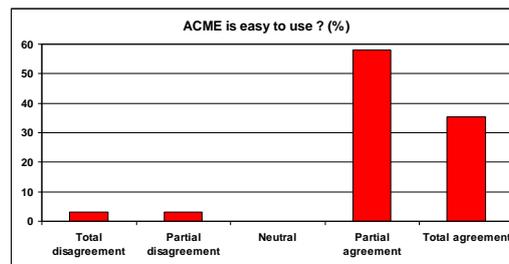
the students used to directly deal with the professor, and, on the other hand, for making it easy the professor's monitoring of their progress.

### 3.2 Feedback Received

ACME has incorporated an opinion poll system to make it easy to get feedback from the students about the ACME suitability in the learning process. The students answer some closed and open questions. In the case of MTP, we have activated these polls at the end of the year.

We want to emphasize and show the results of two of these closed questions: *Is ACME easy to use?* and *Has ACME helped me to learn?* In the next figure we can see the results obtained for both questions. We consider the first one to be very important when a new learning tool is introduced to. We remark the big percentage of students that thought that they had not had difficulties to use ACME.

The second question evaluates the impact of ACME in the daily work of the students. The results demonstrate that the students considered ACME to be a useful learning tool.



Finally, let us remark what the students have said to be the best of using ACME:

- Automatic correction of the solutions sent.
- Flexibility to work remotely.
- The testing method criterion for each problem.
- Environment and schedule tool.

And the worst of it:

- ACME is too strict.
- The deadlines decided by the professors are very tight.

## 4 Conclusions

We have presented the introduction of some ICT tools to the subject MTP. These tools help us to overcome the main limitations of the classical teaching methodology. The ICT tools motivate the students and promote their self-learning, one of the main guidelines of the European Higher Education Area (EHEA).

The e-learning platform ACME has promoted self-learning, monitoring of the professor, and feedback between professor and student.

The pseudo-language POODI has also promoted self-learning. Having a POODI compiler, the students have had a tool to test the correctness of their programs without knowing any standard PPL, but using the same pseudo-language employed in the lectures.

Finally, we consider that the impressions from teachers and students about our experience have been very positive. Thus, we are encouraged to improve the use of these tools in this subject for next year and moreover to introduce new tools, always with the final goal of improving the learning process, according to the EHEA guidelines.

## Acknowledgments

We would like to thank our colleagues from the Department of Computer Science and Applied Mathematics from the University of Girona that contributed to the development of the ACME platform and POODI, and also to the professors who experimented with these tools in the subject “Methodology and Technology of Computer Programming” and in other subjects of the area.

The ACME platform development is funded in part with the project 2005MQD 00160 from the Catalan Government.

## Bibliography

- [1] The Bologna Declaration. <http://ec.europa.eu/education/policies/educ/bologna/bologna.pdf>. 1999
- [2] F. Prados Carrasco; I. Boada Oliveras; J. Poch García; J. Soler Masó; J. Soler Villanueva; El e-learning como complemento a las clases presenciales un caso práctico: el proyecto ACME Congress: Virtual Educa 2004 - Fòrum de les Cultures de Barcelona, Barcelona 2004, Spain.
- [3] A. Venables; L. Haywood; Programming students NEED instant feedback! Congrès: 5th Australasian Computing Education Conference 2003, Adelaide, Australia.
- [4] J. Soler, J. Poch, E. Barrabes, D. Juher and J. Ripoll. *A tool for the continuous assessment and improvement of the student's skills in a mathematics course*. Proceedings of the Symposium TICE (Technology of Information and Communication in Education for Engineering and Industry) pp. 105-110, Lyon, 2002, France.
- [5] F. Prados Carrasco; I. Boada Oliveras; J. Soler Masó; J. Poch García Automatic generation and correction of technical exercises Congrès: International Conference on Engineering and Computer Education: ICECE 2005, Madrid 2005, Spain.
- [6] I. Boada Oliveras; J. Soler Masó; F. Prados Carrasco; J. Poch García A teaching/learning support tool for introductory programming courses Congress: 5th International Conference on Information Technology Based Higher Education and Training: ITHET 2004, Istanbul 2004, Turkey.