

# CoEDApplets

## *Collaborating in the Development of Teaching-oriented Applets*

F. Almeida, V. Blanco

*Dpto. Estadística I. O. y Computación, Universidad de La Laguna, Tenerife, Spain*

J. Regalado, A. Santos

*Galotecnia R. S. S., Tenerife, Spain*

**Keywords:** Java Applets, Simulation, Data structures, Algorithmics techniques.

**Abstract:** In this paper we present the CoEDApplets tool, a web application intended to facilitate the teaching/learning of programming and algorithmics in engineering. Like its predecessor, EDApplets, it is based on Java Applets technology and is animation and display oriented through the use of data structure and algorithm traces. In contrast to the static nature of its predecessor, CoEDApplets is presented as a cooperative portal in which different educational professionals can dynamically incorporate Applets that pursue the same objective. The portal also provides for the internationalization of the Applets, also dynamically. Collaborators will be able to contribute not only with code, but also their translations into various languages, thus helping the collaborative process to transcend beyond borders and into countries where a different language is spoken. CoEDApplets thus introduces elements aimed at supporting diversity within the new teaching framework of the European Higher Education Area, which is expected to create a highly mobile student body.

## 1 INTRODUCTION

The learning of subjects involving algorithms and programming represents perhaps one of the most difficult subjects to learn in the field of engineering. Students have always had problems in assimilating abstract mathematical notions, in particular when these include the dynamics of how algorithms manipulate data. In (Almeida et al., 2004), the EDApplets tool ((EDApplets, 2003)), an interactive tool intended to teach the topics of data structures and algorithms, was introduced. This model aims to bridge the gap created by the dichotomy between different teaching and learning styles.

Montgomery (Montgomery, 1998) asserts that this dichotomy is present throughout each academic year, a conclusion that is supported by an analysis of the dimensions of learning styles proposed by Soloman (Felder and Soloman, 2009) (processing, perception, representation and comprehension) within the context of engineering. He concludes that the vast distance present between the learning styles observed in students and the teaching styles commonly used only serves to exacerbate learning difficulties (see Table 1).

Clearly, one way to individually reconcile students with different teaching/learning styles is through the use of multimedia software for specific subjects ((no Garrido and Lecanda, 1997), (Rodríguez and Caro, 2003)), in which applet-based technology plays a vital role (Wie, 2001), (Sosa and de Sande, 2004). In the specific case of data structures and algorithms, it is common to find various animations scattered online, often under a public use license (Borowski, 1999), or as supplementary multimedia material associated with textbooks ((Rowe, 1997)). This type of approach is backed by over two decades of development. A summary of the state of the art can be found in (Kerren and Stasko, 2001).

Our proposal is rather similar to that used by (Kucera, 2009) and his *Algovision* tool. Our goal is to present the display and animation alongside the code associated with the data structure or algorithm. At the present time we are unaware of any initiatives in the field of data structures or algorithms that allow the user to work cooperatively from the perspective presented here.

The EDApplets proposal starts with the concept of displaying the algorithm in the broadest sense. The

Table 1: Results on learning styles observed by Montgomery.

Dimension	Learning	Teaching
Processing	67% of students learn better by doing	Most classes are typically taught in a passive and reflexive way
Perception	57% of students learn methodically	They are normally taught intuitively
Representation	69% of students learn visually	Classes tend to be oral
Comprehension	28% of students learn globally	Rarely is the global view of a system emphasized

algorithms and their animations are introduced using varying levels of abstraction in an integrated hypermedia environment. In EDApplets, as the students provide the inputs, the display shows the execution of the algorithm step by step, a representation of its most representative code traces and elements (variables, data structures, etc.) and a display of the algorithm's behavior through graphical animations. EDApplets was conceived as a web application based on the technology provided by Java Applets, and is available to the public at [www.pcg.ull.es/edapplets](http://www.pcg.ull.es/edapplets).

In this paper we present the CoEDApplets portal, which is an improvement on the previous version and is shown as a cooperative-type portal in which different education professionals can dynamically incorporate Applets that pursue the same objective: to display data structures and algorithms in an engineering context. Moreover, the dynamic/cooperative nature of the contributions extends to the internationalization process, since features are included to ease the internationalization, also dynamic, of the Applets. Collaborators can contribute not only with codes, but also with their translation into various languages. This will allow the cooperative project to cross borders into countries where a different language is spoken. The portal has been translated by the authors into English. Several students enrolled in the Socrates program have selflessly contributed to the project by translating the portal into German and Danish.

In addition to touching on various teaching aspects that can be aimed at different learning styles (active/reflexive, methodical/intuitive, visual/oral, etc.), the project now includes elements intended to support diversity within the new framework of the European Higher Education Area (EHEA), which is expected to provide for a highly mobile student body. (Almeida et al., 2004) also presents those methodological aspects that should be considered when introducing the tool into the classroom as an instructional supplement. The paper is structured as follows: the 2 section makes a quantitative analysis of the use of the EDApplets tool and lays some of the groundwork for the new CoEDApplets tool; the 3 section describes the interface and general design used in developing

the tool and the cooperative mechanisms offered to every type of user. In the 4 section we introduce the technology used to develop the portal, emphasizing the aspects involving internationalization. Finally, the conclusions are presented in 5.

## 2 EDAPPLET EVALUATION

The EDApplets portal has enjoyed success over the three academic years it has been in use, which saw the development of virtual teaching projects for courses in *Computer Data Processing* for Statistics majors and *Advanced Programming* for Industrial Engineering majors specializing in Electronics, both offered at the University of La Laguna (ULL). A study was conducted on the use of the tool (Figure 1), in which we noted that it is not only used by ULL students, but by those in various universities within Spain and also in Latin America, who access the portal on a regular basis.

Our experience in recent years has revealed some difficulties inherent to the classroom use of EDApplets, mainly as a result of the monolithic nature of the tool which prevents, for example, its expansion through the use of new Applets for instructors not involved in the project. Another particularly rigid aspect in the design involves the internationalization mechanisms. This option is of special interest within the framework of the European Higher Education Area (EHEA). These reasons led us to redesign EDApplets so as to address these two issues. The result is the CoEDApplets presented here.

The CoEDApplets portal has only just been placed into use. We hope to be able to evaluate its usage at the end of this academic year.

## 3 PORTAL DESIGN

Figure 2 shows the main CoEDApplets portal. This portal is being temporarily hosted at <http://www.pcg.ull.es/coedapplets/>. Since it is a



Figure 1: Usage statistics: 2007/2008 Academic year.

teaching-oriented portal, it was decided to moderate the types of contributions received. There are three user types: anonymous, registered and supervisors. Anonymous users can access and execute all of the Applets available in the system and see the source code if desired (see Figure 3). This will be the normal option for students and teachers. Anonymous users will be able to register via the portal. Registered users can also provide new Applets (Upload option) or translations to existing Applets. The uploading of new or translated Applets is moderated by supervisory users. The object of this supervision is no other than to try to ensure coherence in the system, to avoid inconsistencies in the Applet model and to minimize the presence of errors in the portal.

To upload an Applet to the system, an Applet title and description must be provided in either of the two basic languages (Spanish or English), along with the corresponding Applet code (.class, .jar and source code), as well as the necessary information for its internationalization. The user must also provide a category in which to include the Applet. Once an Applet is uploaded, it is queued with other Applets awaiting moderation. An administrator will be tasked with evaluating the various contributions, approving those that are deemed valid and discarding the rest. The moderator will ensure that the Applet is in keeping with the CoEDApplets design model so that it can be adjusted to the teaching methodology behind the CoEDApplets concept. This aspect requires a specific interface design style so as to maintain consistency.

Each applet will simulate the execution of an algorithm. It presents the algorithm's code and the student inputs the data for the simulation run. The student interacts with the Applet through the speed buttons. Every Applet must include the fine panels or work areas (see Figure 4): the data entry panel, the execution speed control panel, the code display panel, the animation panel and the output panel for the trace variables.

Each is described below:

- **Data Entry Panel.** Used to input the values to be simulated by the algorithm. The simulation commences once the 'Start' button is pressed.
- **Speed Control Panel.** Features two buttons, 'Continue' and 'Step by step', which allow alternating the simulation between a continuous and a stepwise execution of the code. In the step by step execution mode, it is necessary to press the 'Step by step' button whenever a new algorithm instruction is to be executed. For the continuous execution model, when so required by the algorithm, a scroll bar is used in the Applet that allows the execution speed to be increased or decreased (see Figure 4).
- **Code Display Panel.** This panel shows the algorithm code to be simulated. All of the codes in CoEDApplets are shown as pseudocode that follows the style of the C programming language. Each line of code is numbered for ease of reference. During the entire simulation process the line



Figure 2: Main page of CoEdapplets portal.

of code being executed is highlighted. At that instant, in the animation panel, the animation action associated with the highlighted line is executed, while the new values assumed by the variables are simultaneously shown in the output panel.

- **Animation Panel.** Shows a graphical display of the algorithm simulation. When so required by the algorithm, the Applet unfolds the panel into as many panels as are necessary to adequately display the simulation, thus increasing the animation capacity. Note in the Applet in Figure 4 how panels have unfolded for the input, output, speed, etc.
- **Output Panel.** Shows the values of the variables that are being traced. The value shown represents the content of these variables before the line highlighted in the code panel is executed.

Figure 4 shows a screen capture during the execution of the Applet For (III). This Applet simulates the assigning of values to an upper triangular matrix. The code used two nested loops with upper limits  $N$  and  $M$ . The values for the loop execution limits are input and the execution shows which piece of data is accessed at every instant and what value it assumes.

This design is intended to provide a channel for the instruction of active and intuitive students through the use of animation and graphical elements, while also adapting to the skills of methodical and reflexive students through the use of reasoning about the code or about its execution trace. The overall integration of techniques and structure allows for a global vision of the objectives when necessary.

In the case of the translations, the cooperative process is similar: a registered user can decide to provide a new translation, in which case a new view would be provided with the translation keys on one side and an empty space to be filled in on the other. Once the process is complete, it is sent and, as was the case with

Applets, queued alongside other translations awaiting moderation. Once a translation is approved, the process of regenerating the .jar file for the Applet in question is started.

## 4 TECHNOLOGIES EMPLOYED

Technically, the project consists of a web application implemented using the Python programming language and the Django rapid development framework (Django, 2009). This framework includes a library set that implements the controller view model and the DRY principle (*Don't Repeat Yourself*).

In addition to using this technology, it was necessary to solve certain problems inherent to the automated handling of the applet translations. Java provides the developer community with a set of libraries that allow internationalization features to be incorporated into those applications that use its libraries (Microsystems, 2009). These libraries require modifying the code to be internationalized, replacing literal chains with calls to a method that is charged with handling its translation. By way of example, below we show a code before and after being internationalized:

```
// Before internationalization
sum = new JButton("sum");
// After
sum = new
JButton(messages.getString("sum"));
```

In this code, the `messages` variable is an instance of the `ResourceBundle` class, which represents a repository of text strings. The proviso lies in the fact that this repository is sensitive to the location of the setting in which the code is executed. Therefore, in a system configured for use in Spanish, the chain that will output the `getString` method

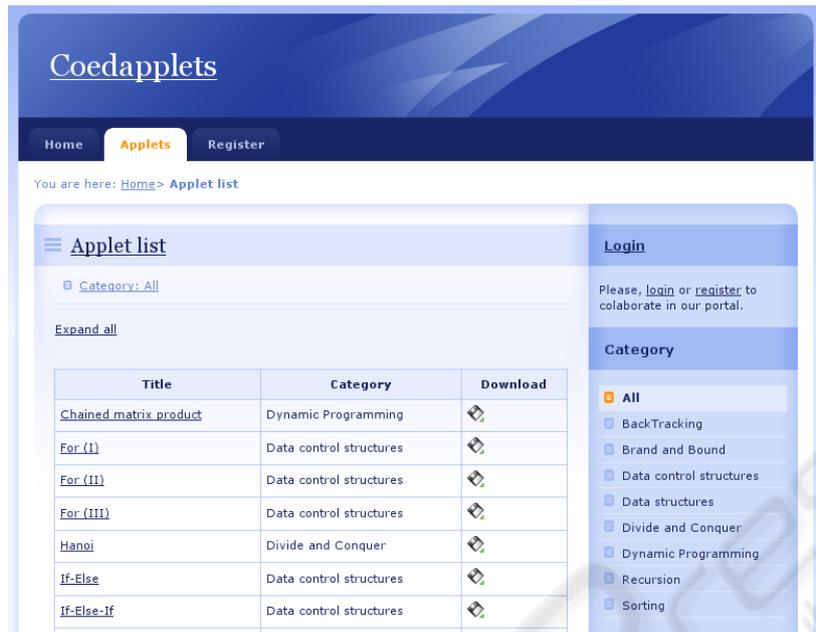


Figure 3: Structure of Applets listing grouped by category.

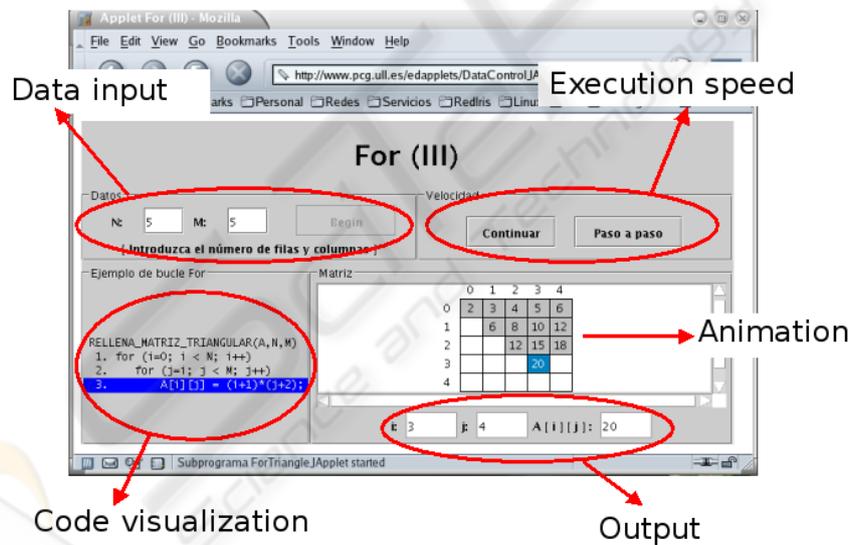


Figure 4: Applet showing a FOR iterative control statement.

will be the translation into said language of the `sum` key. These text repositories correspond to files located in the file system. A master file has to be created (`ResourceBundle.properties`) where the keys and translations are defined for the default language (normally English). Similarly, for each language that is to be supported, a file has to be created `ResourceBundle_XX.properties` in which the chain `XX` is replaced by the ISO code of the language in question. This file will contain the translations as-

sociated with each key defined in the application.

Making use of the internationalization features in Django, both for chains generated in the code and for those stored in the database, enables the interface to be adapted to the user. The mechanism for doing the translations in the case of literal chains present in the code is similar to that mentioned earlier for Java, and involves the use of GNU (GNU, 2009) libraries *gettext* that are based on translation files (`.po` file extension) and on their compiled versions (`.mo` file exten-

sion). In the case of chains present in the database, we resorted to the Django *transdb* application, which expands the database scheme to include the different translations for each of the text fields.

## 5 CONCLUSIONS

In this paper we presented the CoEDApplets tool, a web application for the teaching/learning of programming and algorithmics in Engineering. The most notable facet versus previous versions is that CoEDApplets is presented as a cooperative portal in which different educational professionals can dynamically incorporate Applets that seek the same objective. This cooperation also encompasses the internationalization aspect and provides an added value within the new European Higher Education Area (EHEA). The portal we developed can clearly be conceived as an environment that can be adapted to all types of Applets, not just those intended for the teaching of data structures and algorithms.

## ACKNOWLEDGEMENTS

This work was supported by the Spanish Ministry of Science and Innovation through TIN2008-06570-C04-03 projects. Moreover, it has been developed in the framework of the European network COST-ICT-0805 and the Spanish network CAPAP-H.

## REFERENCES

- Almeida, F., Blanco, V., and Moreno, L. (2004). EDApplets: Una herramienta web para la enseñanza de estructuras de datos y técnicas algorítmicas. In *Actas de las X Jornadas de Enseñanza Universitaria de Informática, Jenui 2004*, pages 409 – 416, Alicante.
- Borowski, B. (1999). Home page. <http://sciris.shu.edu/borowski/sciris.shu.edu/~borowski/>.
- Django (2009). Django framework project. <http://www.djangoproject.com/>.
- EDApplets (2003). Herramientas para la Visualización de Estructuras de Datos y Técnicas Algorítmicas en la Web. <http://www.pcg.ull.es/edapplets/>.
- Felder, R. M. and Soloman, B. A. (2009). Learning Styles and Strategies. <http://www.ncsu.edu/felder-public/ILSdir/styles.htm>.
- GNU (2009). Gettext library. <http://www.gnu.org/software/gettext>.
- Kerren, A. and Stasko, J. T. (2001). Algorithm animation - introduction. In *Software Visualization*, pages 1–15.
- Kucera, L. (2009). How to visualise abstract topics in computer and communication science. In *International Conference on Computer Supported Education*, Lisboa.
- Microsystems, S. (2009). Java Internationalization Technology. <http://java.sun.com/javase/technologies/core/basic/intl/java.sun.com>.
- Montgomery, S. (1998). Addressing Diverse Learning Style Through the Use of Multimedia. <http://www.vpaa.uillinois.edu/tid/resources/montgomery.html>.
- no Garrido, C. C. and Lecanda, R. Q. (1997). Diseños de Entornos de Aprendizaje con Ordenador: Programas Hipermedia para el Autoaprendizaje. Didáctica y Organización Escolar. Universidad del País Vasco. <http://www.cfp.upv.es/>.
- Rodríguez, A. G. and Caro, E. M. (2003). Estilos de Aprendizaje y E-learning. Hacia un Mayor Rendimiento Académico. Departamento de Economía de la Empresa. Universidad Politécnica de Cartagena. <http://www.upct.es/>.
- Rowe, G. (1997). *Intro to Data Structures and Algorithms with Java*. Prentice Hall PTR, Upper Saddle River, NJ, USA.
- Sosa, R. and de Sande, F. (2004). ToTALF: tutorial online para teoría de autómatas y lenguajes formales. In *Proc. de las X Jornadas de Enseñanza Universitaria de la Informática*, pages 219–226, Alicante.
- Wie, C. R. (2001). The Semiconductor Applet Service. <http://jas.eng.buffalo.edu/>.