Mico - University Schedule Planner

Alexandre Freire , Alfredo Goldman ,
Carlos Eduardo Ferreira , Christian Asmussen , Fábio Kon

[1]Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo

{alex,krico}@arca.ime.usp.br {cef,gold,kon}@ime.usp.br
mico.arca.ime.usp.br

Abstract. *This paper describes the development of Mico, the XPUSP University Schedule Planner. It is a system that manages the creation of a schedule for the courses of the Computer Science department and the allocation of professors to teach these courses based on preferences gathered from professors and students. It has been developed using free software and it is itself free software, available under the GNU GPL. We will give a short description of the problem we want to solve and how the system works, then we will discuss the free software frameworks and tools used in the development and our interaction with the free software development community.*

1. Introduction

One of the main administrative tasks in schools and colleges is the construction of the course schedule and allocation of teachers to the courses they will lecture in the period. There are many constraints that play an important role in these allocations. University students show preferences to some elective courses and the period they should be offered. Professors also have their course and time preferences and the department must fulfill its requirements. To find a perfect allocation and schedule is a very difficult task which the department head must face each year.

We did not find any existing software tool that incorporated the features we needed to solve this problem in our department. However, “Arca” (ark in Portuguese): the free software development group at the Department of Computer Science at IME/USP accepted the task to produce this tool. And so “Mico” was born (the tools developed by Arca are named after animals).

Mico is a Web application that manages the creation of such a schedule. It was developed following the eXtreme Programming methodology [1]. The system has been in development since the second semester of 2001. Initially, it was the subject of the first course in eXtreme Programming [1] taught in Brazil. After that, many programmers have contributed. The project is currently maintained by a group of students in the Arca Group [2].

The system works through a website where students and professors input their preferences, professors indicate what time and days of the week they want to lecture in, and students choose which required courses they want to attend and vote on elective courses that they would like to be offered. The application then uses a genetic algorithm [3] to find an optimized schedule and allocate the professors to the courses they should lecture.

Throughout the development of this system many free software tools and frameworks were used following the eXtreme Programming methodology. Section 2 describes the problem requirements we considered. In section 3 we describe the system in some detail, as well as the tools and frameworks used. In section 4 we discuss how the XP methodology applies to the development of free software. This paper ends with some conclusions.

2. The problem

Allocating the professors to the courses they should teach and these courses to given periods is not an easy problem. To solve it, we had to consider many important points.

All student and professor preferences should be taken into account. For instance, the elective courses with the most student votes should be offered, professors can only lecture the courses they are qualified to lecture and in the periods they are available.

Given these and many other restrictions it is possible that a perfect schedule does not even exist. In our application we define some requirements for the solution (e.g. one teacher cannot be allocated to two courses at the same time, the required courses of the semester must be offered). Other constraints are not strong (e.g. preferences of students and teachers) and are used to evaluate the quality of different solutions.

3. The system

The system is accessed via a Web interface available for the students and professors of the department. There are 3 types of users: professors, students and administrators.

Each year the administrators are responsible for starting a poll amongst students and professors. Students can then log into the system and choose which required courses they would like to attend and also vote for their preferred elective courses. The professors can choose which courses they are able to lecture and which courses they prefer. They can also indicate time preferences, tell the system that they will not be available for lectures during a given period or even for a sabbatical semester.

We chose to solve this problem using a known technique for combinatory optimization problems, called genetic algorithms [3], which uses genetic evolution as a metaphor for improving the quality of a solution to a problem. Using this approach we model each possible solution as a gene, and start out with a randomly created population. At each interaction, the algorithm selects the best fit solutions, i.e., the solutions that obey most of the rules we have established, and using different approaches, like mutation or cross-over reproduction, creates a new population. After a given number of interactions, we expect that a near optimal solution will be reached.

After the poll is finished, the administrator executes the solution engine. The algorithm periodically outputs the best solution found so far. When the administrator is satisfied with the given solution he will be able to stop the algorithm and make manual adjustments to the schedule if necessary. He can then publish the results for all users to consult. The system is under development and some parts of this process are not yet implemented.

3.1. Free Software Tools and Frameworks

All throughout the development of this system we have relied solely on a free software infrastructure. Development was carried out on GNU/Linux workstations using common tools, such as bash and Emacs. We chose Java as the main programming language but

also have some code in Perl and C++. The project is currently hosted at SourceForge and we use most of the features available there, such as the CVS server. Tools on SourceForge were used to create a structure to track our project using the XP methodology: store story cards, track progress, and so on.

The system was based on the Model-View-Controller (MVC) pattern [4], we found the necessary tools to speed and re-enforce this pattern available as free software. The Jakarta Group [5] - java sub-division of the Apache Free Software foundation - had a very important role in providing these tool and frameworks.

The code was edited with GNU Emacs, built by Ant (Jakarta's pure java build tool), deployed on Tomcat (Jakarta's servlet container) and persisted on MySql[8]. The versioning of the source is controlled with CVS and hosted on SourceForge. Several other - free software - tools like Open-ssh, GNU make and Dia were used either by the build scripts or on our daily tasks.

Using frameworks such as Turbine[6] (Jakarta's java Web application framework), Velocity[**?**] (Jakarta's java template engine) and Torque[7] (Jakarta's java persistence layer) it was possible to build a database independent, secure Web application with source separated from layout.

To implement the genetic algorithm we have also relied in free software libraries. In the beginning a C++ library was used but due to bugs with the Java to C++ bridge (JNI) this was discontinued. As a replacement we used JGap [9], a java framework.

3.2. Feedback of the free software community

All through development we interacted with the free software community, but most important was the attention we got from other developers. Two North American developers were interested in our project (we believe it is the only free software system available for complex schedule creation) and one of them even submitted some documentation for our project.

In the beginning of the project we mostly asked questions in the Turbine user list and found solutions to common problems. As we grew more competent in these frameworks we also talked to developers, suggesting new features and relating bugs we found. Since we had access to the source code we could even point out where some bugs were located in their code.

In particular, as we started using the JGap library, we maintained a close relationship with the main developer, submitting bugs and feature requests.

4. Producing Free Software using XP Methodology

We have had a great experience using eXtreme Programming techniques during the development, however some problems occurred when dealing with the free software community.

When outside developers showed interest in contributing to our project, we found no plausible way to pair program with them (one of the main "dogmas" of XP). We were located in different continents and time zones, what made it hard for any outside developer to get to know the code and to be able to help. Pair programming is a XP technique where two programmers share the same computer and work together. It is very effective and actually cuts down development time. Also, "the code is the documentation", another XP paradigm, suffered from the fact that outside developers could not come into daily contact with the rest of the team. One of the american developers had trouble understanding some

of the advanced concepts we were using and in the end contributed only documentation that was created to aid him in this understanding, perhaps if the project was simpler and if he could pair with us he could have devoted this time to develop some code as well. Another problem was that a part of this documentation quickly grew outdated.

As support for the eXtreme Programming methodology we have also customized our project page at SourceForge to host online story cards, this way all developers had quick access to all story cards and we could keep an online record of what was being done and track our progress.

Using eXtreme Programming really payed off, we were able to develop a complex system and manage the project easily. It interacted nicely with the free software production model, and even tough we encountered some problems with outside developers we could work around them.

5. Conclusion

We have had a great experience developing free software in the university, and have shown it is possible. It is a good model to adopt since there are so many free software tools, frameworks and libraries available that could be used to accelerate development of many systems. Also, we believe we should give something back to the community and contribute to make it grow.

Having the source available for the projects we had to use was a great asset: we could learn from what others had done and find programming errors easily. Also, the free software community was very helpful and we received some great feedback from interested users.

We felt it was worth using free frameworks even though they were work in progress, we had a good, close relationship with the community and had a clear vision of what was working or not and what to expect. In the Jgap case the developers put together a new release one week after hearing from us, fixing bugs and adding requested features.

It is gratifying to work in free software and find out that the community is interested in your project and would like to use it. Mico has been used to produce the courses allocation for professors in 2003 and 2004. Its use has brought some new necessary features to the system and showed that in house development produced software very useful to the department.

References

[1] [Beck 1999] - *Extreme Programming Explained*.

[2] Arca Project - http://www.arca.ime.usp.br

[3] [Whitley 1993] - *A Genetic Algorithm Tutorial*.

[4] [Model 1988] - *"The Model-View- Controller (MVC) ParadigmUser Interfaces"* OOPSLA'88 Tutorial, ACM.

[5] Jakarta Project - http://jakarta.apache.org/

[6] Turbine - http://jakarta.apache.org/turbine

[7] Velocity - http://jakarta.apache.org/velocity

[8] Torque - http://db.apache.org/torque

[9] MySql - http://www.mysql.com/

[10] Jgap - http://jgap.sourceforge.net