UML .FRI – CASE TOOL DEVELOPMENT

Tomáš BAČA, Milan Juríček, Pavol Odlevák

University of Žilina, Faculty of Management Science and Informatics,
Slovak Republic

e-mail: projekt@umlfri.kst.fri.uniza.sk

Abstract

Modern development of information system is heavily supported with CASE tools (Computer-aided software development). Software analysts and developers use them to communicate among themselves using unified diagrams, usually in UML (Unified Modelling Language), to document their work and to generate parts of code automatically.

Team of master's degree students at our faculty is creating new CASE tool as their master's degree project, aimed at educational process of faculty mainly. The program is named UML .FRI and is released under GNU/GPL license currently as beta version.

***Keywords**: CASE tool, UML, Python, GTK, opensource*

1 INTRODUCTION

Students at Faculty of Management Science and Informatics of the University of Žilina (FMFI) are taught and expected to use UML and few other types of diagrams (E-R, DFD, etc.) during their studies as well. Thus the school has to either pay for the licenses to be allowed to use commercial CASE tools in laboratories and to let students to install them on their private computers or use freeware or opensource products. Both commercial and free-of-charge tools have certain advantages and disadvantages concerning the price of license, speed and guarantee of development and technical support. To benefit from better qualities of both aforementioned options, one could either participate in the development process of an open-source tool or develop one from scratch.

The team of master's degree students at FMFI chose the last option more than two years ago. They started to develop a new CASE tool named UML .FRI. Their main goal is to provide DSM (Domain Specific Modeller) suitable firstly for use in the educational process of the faculty and secondly for public use. The program is released under the GNU/GPL licence currently as beta version. It has been written in Python with use of GTK+ to create GUI. In the current version, several basic features of four common UML diagrams are available.

The most recent development is aimed for improving current functionality and adding new elements to diagrams. Also a new web page of the project is being built. In further future code generation and reverse engineering are planned to be added to the main branch of the project.

2 GENERATIONAL PRINCIPLE OF DEVELOPMENT

UML.FRI is being developed by students at FMFI during their master's degree studies. At the beginning of their studies, students have to choose one project they will be participating in for next three semesters. Students are recommended to use part of three-semester-long work for their master's degree thesis and thus to participate in project for one semester longer.

This project is already in its sixth semester of development and eight students have been taking their part in it so far. During this period of time, it became clear that new participants need significant part of their first semester to learn necessary information to be able to start developing on their own and their main sources are their older colleagues. Also students, working on their master's degree thesis, work more separately than they did before, even if they are still counted as members of the team. During winter semesters at study programme Information Systems, the programme most of the participants are from, students have usually more intensive timetable than during spring ones. All of the mentioned reasons cause students are developing most productively during their second semester of participating in the project.

3 USED TECHNOLOGIES

3.1 Python

Python (CPython implementation) is programming language, the program is being written in. One of its most valuable features is implementation on every widely used platform including Windows, Linux, MacOS X etc, which makes source code, once written, usable on these platforms without change.

Python is multi-paradigm programming language (allows use of object-oriented, structured and other styles of programming). The syntax of Python has been designed with emphasis on readability and productivity. The source code is translated to python bytecode (similar to Java bytecode) and program runs in Python virtual machine.

Therefore high flexibility and development speed is offered at the cost of lower performance under heavy CPU load (which occurs rarely in human-computer interactive programs). If higher performance is needed, module written in C/C++ can be used to solve a problem with a performance bottleneck.

3.2 GKT+, libglade, cairo, pango

GTK+ is cross-platform widget toolkit for creating GUI. The library itself has been written in language C, however bindings for many languages including Python (PyGTK) are available. It was originally designed for X Windowing System, now MS Windows, Mac OS X and other windowing systems are supported as well. GTK+ uses software rendering only.

Glade is RAD (Rapid Application Development) tool for designing GTK GUI. A designed GUI is written into a single XML file, which an application dynamically loads using a library libglade. Thus hard-coding of GUI in Python or other language is replaced by interactive work, easier to be changed.

Cairo is a 2D graphics library with support for multiple output devices. Cairo is designed to produce consistent output on all output media while taking advantage of display hardware acceleration when available. Cairo is implemented as a library written in the C programming language. From the version 2.8, GTK+ has been integrated with Cairo 1.0, giving developers access to its flexible API. The binding for Python pycairo is used.

Cairo is used only to create drawing area of the program to improve performance. The GTK+ canvas had been being used before, however it could not offer the quality the program needed.

Pango is a library for laying out and rendering of text, with an emphasis on internationalization. Pango can be used anywhere that text layout is needed, though most of the work on Pango so far has been done in the context of the GTK+ widget toolkit. Pango forms the core of text and font handling for GTK+. The integration of Pango with Cairo provides a complete solution with high quality text handling and graphics rendering.

3.3 XML (Extensible Markup Language), DOM (Domain Object Model), minidom

XML is general-purpose markup language, text-based, designed to be human-legible. It is open standard recommended by W3C. In the project, XML is used for multiple purpose; configuration files, a metamodel of the program, GUI design used by libglade, output files with models (however they are afterwards packed into ZIP archives).

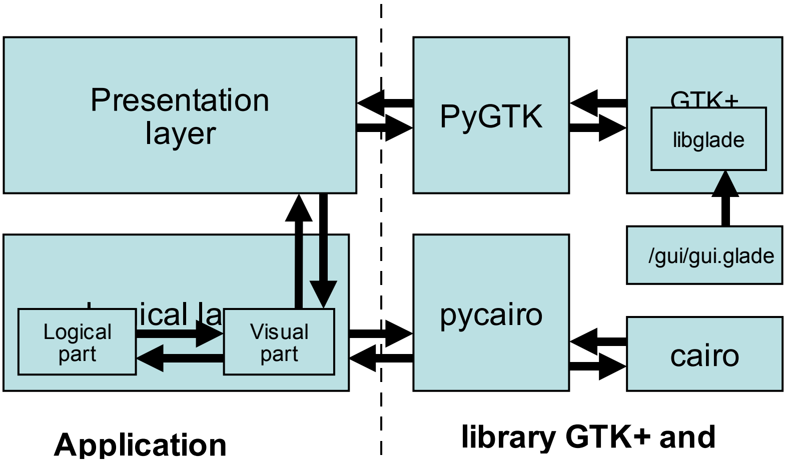
DOM is standard object model for representing XML and related formats. In the project library minidom is used to parse all the XML files mentioned in previous paragraph (except for libglade file).

Since only XML files create metamodel of program new diagrams and elements can be added without change of code of the program. Change can be done also by final user.

4 ARCHITECTURE OF THE PROGRAM

Program is strictly divided into layers with emphasis on object-oriented style of programming.

- Logical layer
  - Logical part – operations with model and metamodel; creation and deletion of elements, handling parameters of elements and diagram, etc.
  - Visual part – geometry of elements, their assigning, clipboard
- Visual layer – GUI of program



Architecture of the program

5 CURRENT FEATURES

The current version of UML.FRI is released as beta version. It has been developed and tested for Linux and Windows platforms.

The program is capable of creating new project, saving it to the file and loading project from the file. The save file format has been inspired by the open document format. ZIP archive containing XML document with model. Save file thus can carry additional information (e.g. icon) accessible by other applications such as browser.

Z-order of elements in diagram can be modified and they can be cut/copied/pasted to/from clipboard. Diagrams are displayed with antialiasing. The whole diagram can be exported to SVG format. Program has Settings dialogue window where appearance of diagram can be modified.

There are four UML diagrams partially implemented, however only class diagram is implemented sufficiently to be usable. Adding extra features has never been high priority, because program is still being tested and concept of metamodel allows new elements very quickly. Users are able and encouraged to add their own features and diagrams as well by editing XML files of metamodel. (The tool for user-interactive editing of metamodel will be created later.)

5.1 Features added during the fifth semester

During the fifth semester there were five students participating in the project.

Jan Janech, the doctoral student, was mainly in position of mentor to the three newcomers. However he created cache of sizes of graphical components, which strongly improved speed of canvas repainting.

Martina Iskerkova, the student in her last year of master's degree studies, has been working on user documentation of the program. The documentation has not been finished yet.

Milan Juríček added Settings menu to the program, which permitted user to change appearance of diagram directly in program. Older way was to manually edit XML configuration file.

Now he is working on the project's web page.

Pavol Odlevak changed original GTK drawing canvas for cairo drawing canvas. Thus he improved speed of repainting and added antialiasing to the drawing canvas. Now he is working on exporting graphical output to files of different formats (e.g. pdf, png ...).

Tomáš Bača has made changes to metamodel adding cardinality to relations in class diagram and improving legibility of captions. Now he is working on ability to move text captions of the connections in diagram independently to the objects they are bound to and further expanding features of diagrams.

6 FUTURE PLANS

The most current development is aimed to complete unfinished tasks; exporting diagrams to various output formats, ability to move captions of connections and the webpage of the project.

The main goal is to use the tool in the educational process of FMFI beginning autumn semester 2008. It implies extending features of currently implemented diagrams and adding new diagrams to metamodel.

There are two branches of project, which were created as master's degree theses a year ago. They partially solve code generation and reverse engineering, however they are not merged to main branch yet. Their merging is one of further future steps.

6.1 Webpage of the project http://umlfri.kst.fri.uniza.sk/

As mentioned in before, the webpage is currently being developed and is already partially active. We hope the webpage will make our CASE tool better known among public and the number of users and testers will increase as well.

The webpage is about to contain articles about UML and diagrams that are supported in UML .FRI, user documentation of program, information about authors, forum for registered users and bugzilla to the project. The webpage will be available in both English and Slovak language.

Users will be able to download the most recent stable version for both platforms and the latest version is already available to download via SVN from address https://umlfri.kst.fri.uniza.sk/svn/ without authentication.

REFERENCES

[1] JANECH, J., SADLOŇ, Ľ.: CASE UML .FRI. In: Objekty 2007, Sborník příspěvků dvanáctého ročníku konference, Ostrava , listopad 2007, pp. 206-210, ISBN 978-80-248-1635-7

[2] http://cairographics.org/

[3] http://en.wikipedia.org/wiki/GTK%2B

[4] http://en.wikipedia.org/wiki/Python_%28programming_language%29

[5] http://python.org/doc

[6] http://www.pango.org/