

Newsletter

FUJABA

Tool Suite

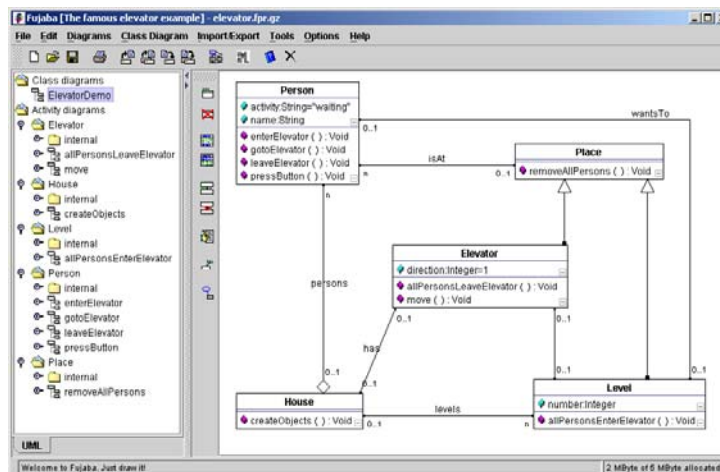
Editorial

...and back again! Welcome to the second issue of our newsletter dedicated to the forthcoming release of FUJABA¹. As we started our newsletter in 2001, we thought of starting a series of newsletters through which we can share with you our latest news and keep you informed about the status of our project. Nearly one year after the release of the first newsletter, we have to admit that we did not reach this goal at all. „Nothing to report on“ you might think. Far from it! In fact, the reorganization of the Fujaba architecture (see our last newsletter) did not leave any time to write a new issue of our newsletter.

Nevertheless, after a year of hard work we are proud to announce that the reorganization and restructuring of the old architecture was a full success. The old monolithic system with over 1,000,000 lines of code has been changed to a modular and highly extendible architecture with just less than 500,000 lines!

This small basis platform supports a plug-in mechanism through which the core functionality can be extended according to the individual needs of the user. Especially for external researchers and

developers who want to use FUJABA as a platform to realize their own ideas the programming becomes easier. With the new architecture they have to learn only some basic concepts on plug-in creation and not - as in the past - the whole rabbit hole FUJABA. In this newsletter we are presenting the plug-in mechanism and some already realized plug-ins. Enjoy this year's newsletter and now, on to the articles...(rw)



1. FUJABA is a project from the Software Engineering Group, University of Paderborn, Germany. See www.fujaba.de for further details.

FUJABA 4.0

We are proud to announce a new version of FUJABA. We will release an alpha version in December 2002. During a whole year of hard work in reengineering and refactoring we added some new features to FUJABA. The most important one is the ability of loading plug-ins. You can read more about plug-ins in FUJABA in the article below.

On our web page (www.fujaba.de) we already announced the new look of FUJABA. We enhanced FUJABA by a modern graphical user interface. We further added in-place-editing for smarter editing of diagrams without opening annoying dialogs. The old dialogs are still there, so users can work in the same way they used to.

As you can imagine, the alpha version is still buggy. We hope that you could help us find and even better remove the bugs by sending bug fixes. How to do this another article in this newsletter will tell you. (lw)

Complete lifecycle support for Jini-Services

If you ever tried to develop, build and deploy Jini-Services you have been stuck with either very limited text-based tools for developing Jini-Services or proprietary frameworks for deploying Jini-Services which are not as robust as you need them. Additionally there was no environment supporting the complete lifecycle of Jini-Services.

This definitely has changed. For the upcoming version of the FUJABA UML-Casetool a plug-in has been developed which supports developing Jini-Services using the sophisticated code-generation facilities of FUJABA. After the development of your Jini-Services you use the newly added support for UML component- and deploymentdiagrams to

connect the different Jini-Services and to plan their deployment. Then this plan is used by a completely fault-tolerant framework for the actual deployment of the services. At last the current deployment of the services in the network can be displayed in deployment diagrams. And as always everything is licensed under the GPL. (mt)

FUJABA Plug-ins

In the five years of FUJABA's life, it has grown to a huge monolithic system of more than 1,000,000 lines of code. It was very hard for new developers to add functionality or even to remove bugs from existing features. Maintenance and project coordination of FUJABA turned into a difficult job for a small group of developers as we are. So we decided to add plug-ins to FUJABA. Existing and new features of FUJABA are now transferred into plug-ins. They are distributed via Internet. FUJABA is able to download and install these plug-ins.

The user is now able to configure FUJABA by extending FUJABA with different plug-ins. He can choose between a small-sized variant of FUJABA with only core-functionality or even add plug-ins as much as he likes.

A sample plug-in with its source code will be distributed along with FUJABA 4.0 alpha. Developers can use this example for building their own plug-in. We hope for a lot of feedback and new plug-ins with great features. Let us know if you are developing a new plug-in so that we can distribute it via our web pages. (lw)

Ever Hunted a BUG in FUJABA?

Hunting is one of mans oldest nature. It was necessary to survive in a world without pubs and pizza-services, and regular competitions took place in order to honor the best.



The age of the computer has changed the picture of such honored gladiators a little bit.

Game animal hunting has taken the back seats and bug hunting has become a popular but also difficult and tiresome task. In the beginning, a fly swatter was sufficient but nowadays we need several specialized tools and a lot of expert knowledge to sweep bugs out of software.

The FUJABA developer community uses another, from our point of view, brilliant idea of dealing with bugs and bug reports: "If you find a bug in FUJABA, do not spam the developers to remove it but report the bug including a deadline YOU will have fixed it."

This principle has two major advantages: First, bugs will be removed quickly, because the person who found the bug is highly motivated to remove it in order to continue his/her own work. Second, all developers gain experience of nearly all parts of FUJABA and become experts.

During the last years, FUJABA got more attention from users, who 'just' wanted to use the different features of the development environment. Therefore, we integrated a bug reporting feature, where every time an exception is raised, the user has to describe the situation the bug occurred and the report is send in addition with the current project file to the FUJABA mailing list. "Have you seen this feature, when using FUJABA?". If not, this is an indication of the nearly bug-freeness. If you have seen the bug report sending dialog, haven't you been frustrated not to be able to fix the bug by yourself? FUJABA includes potentialities for both parties. For the frustrated persons, we offer a membership in FUJABA's developer team.

If you are interested, please contact the management board via email: fujaba-ps@fujaba.de. For those of you, who



wanted to sharpen their skills in bug hunting and also wanted to get a feeling of the spirits of hunting, try the following during your class diagram design. Open the about box, make a triple click on the FUJABA logo and "Do Your Very Best". (jn)

The PLC Plug-in



During our research in an industrial strength case study which deals with generating software for production control systems (the picture shows a shuttle of our track based material flow system), we came across the need for generating code for Programmable Logic Controllers (PLCs). Now, in contrast to Java code generation, object oriented concepts like classes, inheritance or polymorphism, cannot be mapped directly to the PLC target languages. Similar problems arise, if you want to generate C-Code from object oriented specifications which is a common problem in the domain of embedded systems. As target language, we use the so called Structured Text, which is very similar to C.

Unfortunately, this is not the only problem we have to deal with. In addition, we have to consider the restricted resources of the target hardware - our PLCs only have 64kb (!) of memory. (It's like going back to the early eighties and the good old C-64). Moreover, PLCs have very special execution semantics.

A more detailed explanation would go beyond the scope of this newsletter, but more material will be available, soon.

Don't worry! We spare no effort on making the specification of such control software as easy as possible for you. In other words: you can specify your software with FUJABA as you are used to. (un)

May I introduce...

...the initiator of the Fujaba project? It is Professor Albert Zündorf who started the project in 1998 at the University of Paderborn. After only a couple of month, a first release of FUJABA was available. It was offered to the community under the GPL licence. His research interests on reengineer-

ing and design pattern detection influenced significantly the further development of the tool. His work and his innovative ideas made FUJABA to what it is today.



This year, Professor Zündorf changed to the University of Kassel. He is still involved in the development of new concepts and their implementation in FUJABA. Like to get more information? Visit his homepage at <http://pm-pc1.pm.e-technik.uni-kassel.de/fpm> (rw)