

Tool-Box Frameworks - the new Challenge beyond Black-Box and White-Box

Martin Lippert

University of Hamburg
Computer Science Department, SE group
&
APCON Workplace Solutions GmbH
Vogt-Kölln-Straße 30
22527 Hamburg, Germany

lippert@acm.org

1 MOTIVATION

Building applications on top of application frameworks can be a painful job. Even if a framework promises higher reuse and better application design, the engineering often has to be done at source-code level using a simple development environment. We had this experience using our JWAM framework (see [3]), a Java-based application framework for large-scale software development using the Tools & Materials approach (see [1]).

The JWAM framework consists of a number of black-box and white-box (see [2]) sub-frameworks. Our experiences with these models suggest that they are not efficient enough. Often the application developer uses a normal development environment that is not especially designed to support exactly this framework. At first glance, this seems like a fine idea: The development environment should be independent of the used application framework and the application framework should be independent of the used development environment. This promises a high reusability and gives the developer as much freedom of choice as possible. But in practice, the lack of framework-specific support inside the development environment is obvious and painful. From the beginning of the development process to the end the developer has to work on the level of classes, interfaces, methods and other elements of the black- and/or white-box frameworks and the programming language.

2 THE TOOLBOX METAPHOR

To solve this problem we've developed a number of specialized tools to support the application developer in the underlying framework. These tools are specifically built for the JWAM framework and they work on top of the design metaphors of the framework: They don't support the developer in using a single black-box class or subclassing a white-box class of the framework. The

tools help the developer using, for example, the Tool design metaphor, the Material metaphor, or the Domain Value metaphor.

Doing this the user can develop a new Tool on top of predefined (black-box and white-box) classes inside the framework without working on the class level from the beginning, using specialized tools to build the basic foundation for the application. Since the first access to the framework is through the tools provided by the framework we call this a "Tool-Box"-Metaphor.

3 TOOLS FOR THE JWAM FRAMEWORK

The tools for the JWAM framework generate source code that can be subject of ongoing work by the application developer. The generated source code is highly readable and very similar to the normal code written by an application developer using the framework without the tools. The tools are themselves built using the JWAM framework and its design metaphors.

The usage model of the tools is oriented to the tasks the application developer has to do. Examples are:

- building a tool for a specific task.
- adding an event communication inside a tool.
- adding and removing connections between the elements on the user interface and the tool implementation.
- building a domain value.
- building a form as a special material.

There are other tasks that deserve tool support and it is still ongoing research to find the right ones. To find the right tasks we work in close cooperation with our application development projects using the framework as well as with industrial partners using the JWAM framework.

The tools aren't standalone applications. We built small wrappers to integrate them into the JBuilder Foundation Edition 3.5 to enable a smooth application of the tools inside the normal development environment. The JBuilder environment integration is only a demonstration of the principal possibilities. It is ongoing research to find out how close they have to use the exact API of the IDE.

4 CONCLUSION

The demonstration will focus on the developed tools. A small sample application will be developed during the presentation to demonstrate the usage of the tools and the integration into the JBuilder Foundation Edition 3.5

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

OOPSLA 2000 Companion Minneapolis, Minnesota
(c) Copyright ACM 2000 1-58113-307-3/00/10...\$5.00

development environment. We show how the Tool-Box metaphor enhances the possibilities using an application framework.

REFERENCES

1. U. Bürkle, G. Gryczan, H. Züllighoven: Object-Oriented System Development in a Banking Project: Methodology, Experiences, and Conclusions, in *Human-Computer Interactions, Special Issue: Empirical Studies of Object-Oriented Design*, Vol. 10, No. 2 & 3, 1995, Lawrence Erlbaum Associates Publishers Hillsdale, New Jersey, England, 1995, pp. 293-336.
2. E. Gamma, R. Helm, R. Johnson, J. Vlissides: *Design Patterns – Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, Massachusetts, 1994.
3. The JWAM framework: <http://www.jwam.org>