Technology for integration of loosely coupled web-based software components

Dalimír Orfánus and René Pázman

Softec Ltd., Bratislava, Slovakia,
dalimir.orfanus@softec.sk, rene.pazman@softec.sk

Abstract. In this paper we present integration technology which integrates loosely coupled software tools which process heterogeneous information from web sources.
Goals of the integration technology were to unify web presentation and to support cooperation of proposed tools. For this purpose we used development framework Cocoon which is built around the concepts of separation of concerns and component-based web development. We also included the framework Hibernate for object-relational mapping and the library log4j for easily logging. Besides this, we extended Cocoon with specialized generator which helps tools to generate their web presentations, communicate with integration technology and to cooperate with other tools.

1 Introduction

Outputs of our ongoing project of building a job offer portal (development of methods and tools for process heterogeneous information from web sources — see [7]) are tools for defined purposes. They collaborate on different levels. During begging phases of the project each tool had to have its own web presentation. The purpose of the web presentation was to explicitly demonstrate functionality of the tool. Following phases have brought up new tools. Some of the old tools become hidden because they do not directly interact with the user. These hidden tools provide services for other tools.

Some tools are using the web development framework Cocoon for its presentation facilities. It was necessary for the hidden tools to help them with generating web presentation. This help should not influence the integrity of the hidden tools. We had to provide also means for collaboration of all tools.

We had proposed a solution which becomes integration technology. The core of this technology is the web development framework Cocoon. The usage of this framework ensures easy collaboration and integration of the hidden tools with presentation tools. In addition, we enabled to use other frameworks which support integration, loosely coupling, working with database, logging and good practices in general.

Structure of this paper is as follows. In section 2 we present technology overview. In each subsection of section 2 we briefly mention problems and how we coped with them. In section 3 we discuss future features of the integration technology.

2 Technology overview

The integration is based upon following frameworks and libraries:

- `Cocoon` ([1]),
- `Spring` ([5]),
- `Hibernate` ([4]),
- `log4j` ([3]).

The usage of the individual frameworks and libraries is described in the following sections.

2.1 Presentation

Some tools designated for user interaction use the framework `Cocoon` and do not need any change in the way of generating outputs. The problem was how to support service tools with generating their presentation with very small influence on their coherence. All of these service tools will become hidden and it is not convenient to redesign them later, particularly when functionality remains the same. The generation of the web presentation of service tools should not influence their core design and implementation.

For the service tools, it is necessary to wrap them with a presentation layer which:

1. receives stimulus from the environment,
2. transforms these stimuli to the invocation of services,
3. collects responses of the invoked services,
4. resends collected outputs to the environment in a valid form.

Wrappers are based on the same principle as Java Servlet ([7]) technology. Content of the web page is generated directly to the given stream which is sent to the user browser. The integration technology accepts modified XHTML ([8]) format to help wrappers to generate web-pages. The wrappers specify some operations in this format which the technology takes into account. These operations are executed by the technology and the output generated by the wrapper is modified to be a valid XHTML stream.

2.2 Communication with integration technology

Invoked services of the tools need sometimes additional parameters and information for correct processing. To enable tools and their wrappers to obtain all necessary information, we proposed a way of communication with the integration technology. The communication is done through defined interface. The integration technology provides the reference of this interface for the wrappers. This interface provides additional information about http requests, environment, paths on the file system and other used frameworks and it also initializes logging mechanism.

2.3 Collaboration of tools

Each tool can be considered as an independent component with very high degree of coherence. Some of the tools depend on services of other tools. They are using services of the other tools to provide their own services. The design pattern Dependency Injection ([2]) provides an effective solution to support this kind of collaboration. There are several frameworks which support implementation of this pattern. For the collaboration of the tools, we augmented the integration technology with a light-weighted dependency injection container situated in the `Spring` framework ([5]).

2.4 Diagnostics

To perform continuous diagnostics of the whole application, to debug it and to easily find cause of crash, it is necessary to generate diagnostic messages. These messages are generated by the tools as well as by the integration technology. All these messages can be logged to the file, database, e-mail message, etc.

The integration technology uses the common and most spread logging library `log4j`. The technology provides optional initialization of the logging for each tool. This logging is pre-configured to store all diagnostics messages to separate files. Each tool has its own logging file which helps developers to debug their tools. Even thus messages from all tools and the integration technology are being collected and then stored in a common file. We can get overall overview of the whole running application from this common file.

3 Conclusion

We described common problems with integrating loosely coupled web-based components and sketched few basic principles of an integration technology. Such integration technology mediates interaction between user- and service-oriented tools. It is also glue among the tools helping them collaborate mutually.

The next version of the integration technology will focus to provide more specific services and to adapt to new demands of the tools.

Acknowledgements

This work was partially supported by the Slovak State Programme of Research and Development ”Establishing of Information Society” under the contract No. 1025/04.

References

1. Apache Cocoon Project Management Committee: The Apache Cocoon Project. 2005. `http://cocoon.apache.org/`

2. Fowler, M: Inversion of Control Containers and the Dependency Injection pattern. 2005. http://www.martinfowler.com/articles/injection.html
3. Gülcü, C.: Short introduction to log4j. March 2002. http://logging.apache.org/log4j/docs/manual.html
4. JBoss, Inc.: HIBERNATE — Relational Persistence for Idiomatic Java, Hibernate Reference Documentation, 3.1.1. http://www.hibernate.org/hib_docs/v3/reference/en/html/index.html
5. Johnson, R.: Introduction to the Spring Framework. TheServerSide.COM, May 2005. http://www.theserverside.com/tt/articles/article.tss?l=SpringFramework
6. Návrat, P., Bieliková, M., Rozinajová, V.: Methods and Tools for Acquiring and Presenting Information and Knowledge in the Web. In: Rachev, B., Smrikarov, A. (Eds.): CompSysTech 2005, Varna, Bulgaria, June 2005. pp. IIIB.7.1–IIIB.7.6.
7. Sun Microsystems, Inc.: The Java Servlet API White Paper. 2006. http://java.sun.com/products/servlet/whitepaper.html
8. World Wide Web Consortium (W3C): XHTML 1.0 The Extensible HyperText Markup Language (Second Edition), A Reformulation of HTML 4 in XML 1.0. W3C Recommendation, August 2002. http://www.w3.org/TR/xhtml1/