

Using Semantics in Portal Development*

Torsten Dettborn¹, Birgitta König-Ries¹, and Martin Welsch²

¹ Institute of Computer Science, Friedrich-Schiller-Universität Jena, Germany
dettborn|koenig@informatik.uni-jena.de

² IBM Research and Development, Böblingen, Germany
martin.welsch@de.ibm.com

1 Introduction

Portals are more and more becoming environments for complex applications. The new Java portlet standard, JSR 286³, i.e., offers mechanisms for creating portlets as re-usable components which can interact via eventing mechanisms. Thus, they can be assembled into composite applications. In the future, it is envisioned that this assembly shall be done not only by programmers but by business users [1]. This will allow for tailored-made applications. However, finding and correctly combining portlets is a complex task, in particular, when portlets developed by different individuals or providers need to be integrated and when the number of available portlets is high. Non-programmers will need support to perform this task. Formal, machine-understandable descriptions of portlets could help to discover appropriate portlets and to automatically check the designed data flow. We argue that semantic service description languages are a good starting point for such a formalism.

A number of approaches exist, that try to augment classical software reuse systems with semantics to achieve a higher degree of automation and user support [2–4]. To the best of our knowledge, up to now, no approaches specifically tailored to portals have been proposed. Finally, over the last few years, a number of tools like Microsoft Popfly, Yahoo Pipes, and IBM’s QEDWiki to support the creation of mashups have been made available. On the one hand, those tools show that application development by more or less unexperienced users is possible, on the other hand, they do not solve the problems addressed in this paper, still require a certain amount of technical knowledge (at least for more advanced combinations), and are not geared towards a portal environment.

2 Exploiting Semantic Descriptions for Portal Development

Within our group, DSD (DIANE Service descriptions) [5], a light-weight ontology language specifically tailored towards services has been developed and extensively evaluated within the Semantic Web Services Challenge⁴. This language describes services mostly by the set of effects they can achieve and allows

* This work is being supported by IBM through a Faculty Award.

³ <http://jcp.org/en/jsr/detail?id=286>

⁴ <http://sws-challenge.org>

for the configuration of these sets through variables. In the following, we take a look at how this language (or any other semantic service description) could be applied to portlet description. In order to develop their own portal applications, users will need to be able to find portlets that offer a desired functionality and possess the needed communication capabilities, i.e., are able to produce or consume certain events.

The description mechanisms in DSD allow for precise description and efficient comparison. DSD can be used as-is to describe portlet functionality. A slight adaptation is needed to also capture events fired by a portlet. These could be modeled either as additional effects or as additional output variables. Similarly, expected events can be modeled as inputs or possibly preconditions. DSD supports fully automatic discovery and configuration of services. To achieve this, rather sophisticated request descriptions are needed. For portal application development, a semi-automatic approach is sufficient and requires light-weight request descriptions only. Recently, tagging based approaches [6, 7]. have gained some popularity. We propose to combine heavy-weight semantic descriptions of the portlets with lightweight descriptions of user needs. This allows to on the one hand leverage the power of the formal approaches with respect to composition and validation and on the other hand offers a usable interface even to non-expert users.

3 Summary and Conclusion

The semantic description of portlets seems to be a promising basis for end users to create their own portal application. Semantic service languages are a good starting point for portlet descriptions, but they must be extended. Currently, we are working towards extending our description language, DSD, and developing a tool to support business users in portal application development.

References

1. Altinel, M., Brown, P., Cline, S., Kartha, R., Louie, E., Markl, V., Mau, L., Ng, Y.H., Simmen, D.E., Singh, A.: Damia - a data mashup fabric for intranet applications. In: Proc. of 33rd Conf. on Very Large Databases (VLDB). (2007) 1370–1373
2. de Almeida Falbo, R., Guizzardi, G., Natali, A.C.C., Bertollo, G., Ruy, F.F., Mian, P.G.: Towards semantic software engineering environments. In: SEKE. (2002)
3. Sabou, M., Pan, J.: Towards semantically enhanced web service repositories. J. Web Sem. **5**(2) (2007) 142–150
4. Oberle, D., Staab, S., Eberhart, A.: Web systems: Semantic management of distributed web applications. IEEE Distributed Systems Online **7**(5) (2006)
5. Küster, U., König-Ries, B., Klein, M., Stern, M.: DIANE - a matchmaking-centered framework for automated service discovery, composition, binding and invocation on the web. International Journal of Electronic Commerce (IJEC) - Special Issue on Semantic Matchmaking and Retrieval **12**(2) (2007)
6. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R.V., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y.: Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In: WWW. (2003) 178–186
7. Handschuh, S., Staab, S., Volz, R.: On deep annotation. In: WWW. (2003) 431–438