Dynamic Service Reconfiguration and Enactment Using an Open Matching Architecture

Sander van Splunter [a] Frances Brazier [a] Julian Padget [b] Omer Rana [c]

[a] *Faculty of Technology, Policy and Management, TU Delft, The Netherlands*[1]
[b] *Department of Computer Science, University of Bath, UK*
[c] *School of Computer Science, Cardiff University, UK*

1 Introduction

The full paper [3] of this extended abstract presents an architecture for dynamic reconguration of complex services, in which the enactment is automated, and the matching of services is not limited to a pre-determined set of matchers and repositories. The proposed architecture consists of three, previously developed, components: the CoWS template-based reconguration service [4], the Knoogle MatchMaker service [1], and the Triana workow enactment engine [2]. This architecture has the following innovative aspects: 1) automated adaptation of complex services, which is more exible than existing approaches based on replacing failing instances of services within a workow, 2) use of heterogeneous components that may be both local and distributed, and 3) dynamic selection of matchers and repositories.

Web services provide uniform access to software capability with well-defined interface descriptions, and provide some limited options for semantic annotation. Many times, however, a single web service does not provide the precise functionality required for a specific application, therefore web services often need to be combined to form a complex service. If one web service fails, the continuity of the complex service is threatened. Instance-based replacement approaches offer one possible solution, but are limited to the availability of *exact* (function and interface equivalent) replacement services.

Current research allows automated handling of failing services, but is often limited to instance replacement, for example by using late binding. The number of services available is large, but dispersed over different repositories, as different communities build and collect services. Even the design and deployment approaches may vary, for example, between Service Oriented Architecture/Grid and Peer-to-Peer based deployments.

The approach presented in this paper focuses on automated template-based reconfiguration of annotated web services, using (i) a discovery infrastructure with dynamic matcher service selection, repository selection and service policy specification, and (ii) a workflow enactment engine capable of handling heterogeneous components. This combination provides a fully automated reconfiguration service that can be used without human intervention. The resulting architecture enables: (i) automated service reconfiguration based on constraints that have been previously defined by a user; (ii) an *open* matching architecture that allows multiple matching mechanisms to be used, on a set of user-defined repositories—this is particularly useful to enable application-specific matchers to co-exist alongside generic matchers based on term syntax; (iii) generality through integration with a workflow enactment engine, so that the approach can be combined as a component within an existing workflow.

The architecture for this approach is depicted in Figure 1, combining (i) the CoWS template-based web service reconfiguration, (ii) the Knoogle matchmaker, and (iii) the Triana enactment engine. CoWS is an automated template-based approach to reconfiguration of complex web services, in which the scope of the

[1] moved from Intelligent Interactive Distributed Systems, VU University Amsterdam, The Netherlands

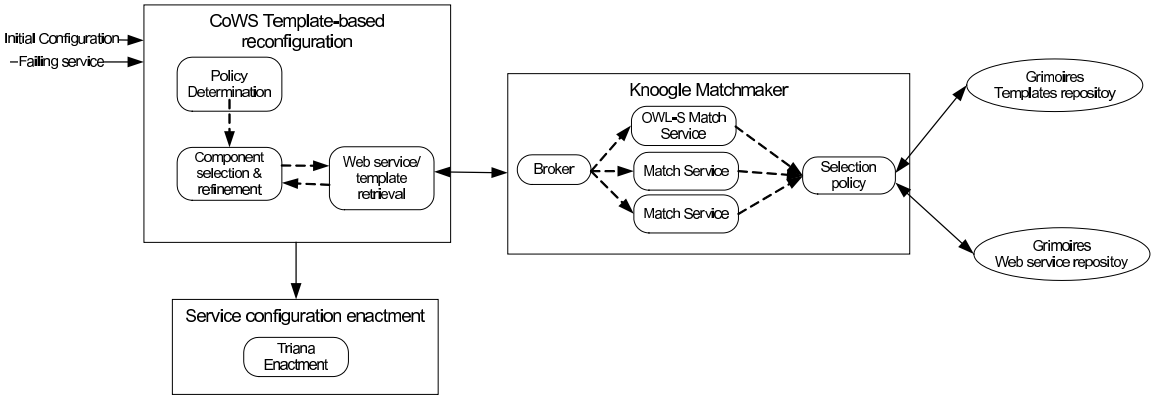

Figure 1: Architecture for dynamic service composition and enactment.

adaptation is adjustable. In addition CoWS enables reasoning on the effect of the replacement of a service in relation to the properties of the complex service.

Knoogle offers a brokerage framework that can be used to deploy partially or fully-configured brokers that query multiple service repositories, employ multiple matching services and apply pre-defined or bespoke selection policies.

Triana is a problem-solving environment designed for both the creation and execution of workflow graphs. Execution of these graphs can be done both locally and via a distributed enactment process (using Triana servers). Enabling enactment of reconfigured services in Triana allows integration of heterogeneous components and both local and distributed management of the execution of the services.

2 Conclusions

Integrating three systems in the manner described makes the approach more complex to use, as it is generally required for a user to specify: (i) service description; (ii) selection policy; (iii) choice of suitable matchers; (iv) list of repositories to search. However, some of these can be pre-configured—such as (ii), (iii) and (iv). Currently, approaches to automated service composition (as outlined in the full paper) are limited, either due to their lack of adaptability to specific application domains, or their overall complexity of use. By allowing different levels of system configuration, and enabling some of these to be pre-specified (not by an end user, but an application/system administrator), we believe it is possible to balance complexity of use with generality of the overall approach.

ACKNOWLEDGEMENTS: The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project (`http://www.icis.decis.nl`), supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. The authors are grateful for the support offered for this research by NLnet (`http://www.nlnet.nl`), and through the Open Middleware Infrastructure Institute (`http://www.omii.ac.uk`) managed program (project Knoogle).

References

[1] N. Chapman, S. A. Ludwig, W. Naylor, J. Padget, and O. F. Rana. Matchmaking support for dynamic workflow composition. In *E-SCIENCE '07: Proceedings of the Third IEEE International Conference on e-Science and Grid Computing*, pages 371–378, Washington, DC, USA, 2007. IEEE Computer Society.

[2] S. Majithia, M. S. Shields, I. J. Taylor, and I. Wang. Triana: A graphical web service composition and execution toolkit. In *Proceedings of the IEEE International Conference on Web Services (ICWS'04)*, pages 514–524. IEEE Computer Society, 2004.

[3] S. van Splunter, F. M. T. Brazier, J. Padget, and O. Rana. Dynamic service reconfiguration and enactment using an open matching architecure. In *Proceedings of the International Conference on Agents and Artificial Intelligence, Porto, Portugal*, January 2009. Accepted for publication.

[4] S. van Splunter, P. H. G. van Langen, and F. M. T. Brazier. Decentralized performance-aware reconfiguration of complex service configurations. In *Proceedings of the Performance for Peer-to-Peer Systems (P4P2P) workshop*. University of Warwick, May 2008.