Automated Web Service Reconfiguration

van Splunter, S.

2010

Summary

This thesis has as its hypothesis that automated adaptation of complex services is feasible using local knowledge. Local knowledge allows changes to be local, reducing structural change to a minimum, and avoiding changes in functionality. To create a system able to autonomously perform reconfiguration based on local knowledge, propagation is needed. Propagations enables reasoning on non-local implications of adaptations, based on local knowledge, e.g., for Quality of Service properties. The following research questions need to be answered:

How can local knowledge be represented and included within a configuration?

What process is needed for automated re-configuration using local knowledge?

Chapter 2 presents related research on automated reconfiguration, and concludes that each of the different approaches has its strengths and weaknesses. Component based replacement is an option that remains closest to the initial configuration, but depends heavily on the availability of an exact match. Single-level reconfiguration supports more advanced adaptations, but this requires each configuration to be specifically annotated. Multi-level configuration allows more variation in adapting the part afflicted by the reconfiguration, but requires more compositional annotated structures. Configuration as reconfiguration is often hindered by the lack of explicit requirements for the configuration that needs to be reconfigured. Ideally a hybrid approach is constructed, which can vary the scope of the adaptation, and allows the most appropriate approach for the selected scope. The approach described in this thesis is a hybrid approach.

The representation and inclusion of local knowledge within a configuration is addressed in Chapter 3. In this chapter the following structures are defined for this purpose: templates in Section 3.2, slots in Section 3.2, and template-based configurations in Section 3.3. A template defines and annotates the requirements associated with a complex web service in more

detail than an individual service, yet in less detail than a complete complex service. A template represents a single level of a configuration. The slots in a template specify the requirements for each of the components (i.e., a web service or a template) in a template. Propagation, see Section 3.4, resolves dependencies between different levels in a template-based configuration to enable reasoning on requirements on, for example, Quality of Service using local compositionality.

The process needed for automated re-configuration using local knowledge is addressed in Chapter 4. A generic reconfiguration process is defined, with as subprocesses Focus Determination in Section 4.2.1, Requirement Determination in Section 4.2.2, Template-based Configuration in Section 4.2.3, and Integration in Section 4.2.4. A prototype is implemented as described in Section 4.3, with further details in Appendices D to H.

The feasibility of template-based reconfiguration is shown by its application in two domains: classification services in Chapter 5, and mathematical services in Chapter 6. Single-level reconfiguration is illustrated in Chapter 5, which illustrates that the approach presented is at least as strong as the current reconfiguration approaches. Multi-level reconfiguration, illustrated in Chapter 6, shows the added value of a gradual increase of the scope of adaptation, and of local reasoning on Quality of Service attributes. A single prototype supports two use cases (i.e., scenarios). These two use cases show that (1) the structures defined to represent and include local knowledge are generic enough be used with existing services from two different domains, (2) the process defined for reconfiguration is generic enough to resolve reconfiguration problems in two different domains, and that it can handle both single-level reconfiguration challenges, as described in Section 5.3, as multi-level reconfiguration challenges, as described in Section 6.3.