International GTE Conference
MANUFACTURING 2008

6-7 November, 2008
Budapest, Hungary

Using intelligent agents for service adaptation

András Micsik, Paul Karaenke

MTA SZTAKI Budapest, e-mail: micsik @ sztaki.hu
University of Hohenheim, Information Systems II, e-mail: paul.karaenke @ uni-hohenheim.de

INTRODUCTION

Services are becoming widely used in the world of engineering and manufacturing as well. Both traditional and computational resources are often encapsulated as Web services, providing the basis for building service-oriented architectures (SOA) based on standards. Services are used on the basis of contractual agreements. A service level agreement (SLA) provides the operational definition of a service as part of a contract between a service provider and a service consumer and includes explicit format statements about the contracted service level. Thus, a SLA contains obligations and guarantees regarding the service in a business relationship. A machine-understandable representation of SLAs enables customers and service providers to monitor the fulfillment of SLAs, and make adaptations in case of SLA violations. The present paper describes a methodology to manage SLA adaptations combining existing techniques of software agents and the Semantic Web.

AGENT FRAMEWORK FOR ADAPTATION

In this section we present the agent-based framework designed and implemented in the European BREIN project for managing SLAs and resources. The goal of the framework is to support performing complex tasks on a variety of resources. Tasks are composed of atomic tasks, and each atomic task can be executed using a certain type of resources. Resources are encapsulated by software agents, managing the task executions on these resources. Furthermore, service providers have coordinator agents, coordinating the execution of complex tasks on several resources. Each atomic task has an associated SLA which was created during the SLA negotiation process which is out of the scope of this paper.

ISBN-978-963-9058-25-5

SLAs contain information about the guarantees assured in an agreement. The service level guarantees are defined as expected service parameters in terms of given metrics. Currently, SLAs are mostly defined using XML (e.g. WS-Agreement [3]). It is unfeasible to collect and standardize all possible service parameters, service metrics, and economic values, as these are assumed to be domain dependant. Hence, the dynamic interpretation of SLAs plays an important role. It can be achieved by using decoupled intelligence in SLA management, such as semantic reasoning techniques and intelligent agents. In our framework SLA definitions are mapped into a knowledge base, so that standard reasoning mechanisms can be used to interpret the SLA definitions properly, relate them to other service parameters and metrics, and finally make decisions based on a sound logical basis.

The agents in the framework are implemented on the basis of the belief-desire-intention (BDI) architecture approach [4]. The BDI model consists of the following concepts: *beliefs* capture informational attitudes realized as a data structure containing current *facts* about the world. *Desires* capture the motivational attitudes realized as *goals* that represent the concrete motivation. *Intentions* capture the deliberative attitudes realized by reasoning to select appropriate *actions* to achieve given goals or to react to particular situations. A BDI agent is equipped with sensors to assist it on its environmental awareness, and effectors to impact the environment by actions. A reasoning mechanism between the sensors' input and the effectors' output deduces the necessary actions for achieving the agent's goals (Figure 1).

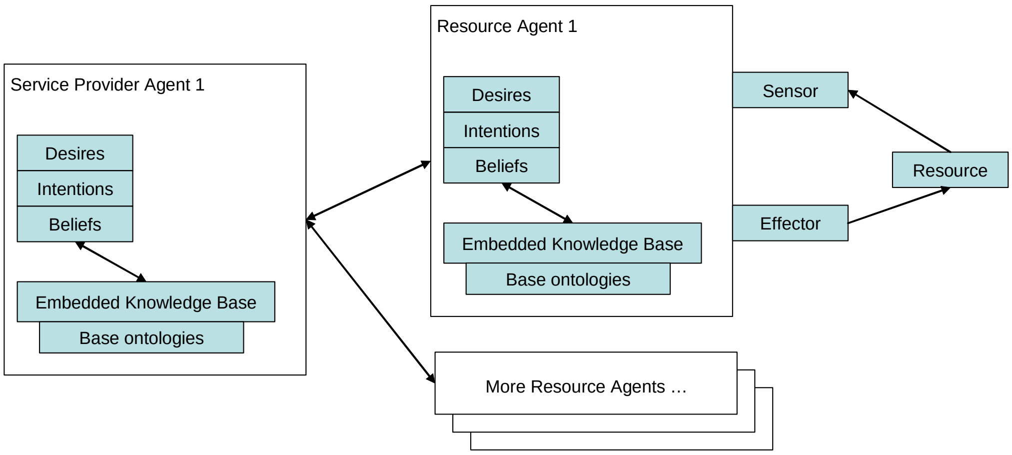

Figure 1: Agents and components in the service provider domain

The agent acquires new beliefs in response to changes in the environment and through the actions that it performs as it carries out its intentions. In the case of the BREIN framework, an agent representing a resource has beliefs about the current state, capabilities, and SLAs of the resource. The agent's desires represent the business goals of the resource provider, while intentions provide a collection of possible actions to select and execute tasks. In our experiments Jade was used as an agent platform and Jadex as core BDI implementation on top of Jade.

Figure 2 shows the interactions of the agent's internal components for semantic BDI reasoning: based on an internal (pro-active behavior, step 1) or external (reactive behavior, step 2) event, the agent first stores new facts into its beliefbase (step 3). The agent utilizes semantic reasoning to assess the event, deriving new knowledge (step 5-10) and especially appropriate intentions to achieve the agent's goals (step 11). These intentions lead to actions (step 12) which potentially include interactions with external components via the agent's effector (step 13).

ISBN-978-963-9058-25-5

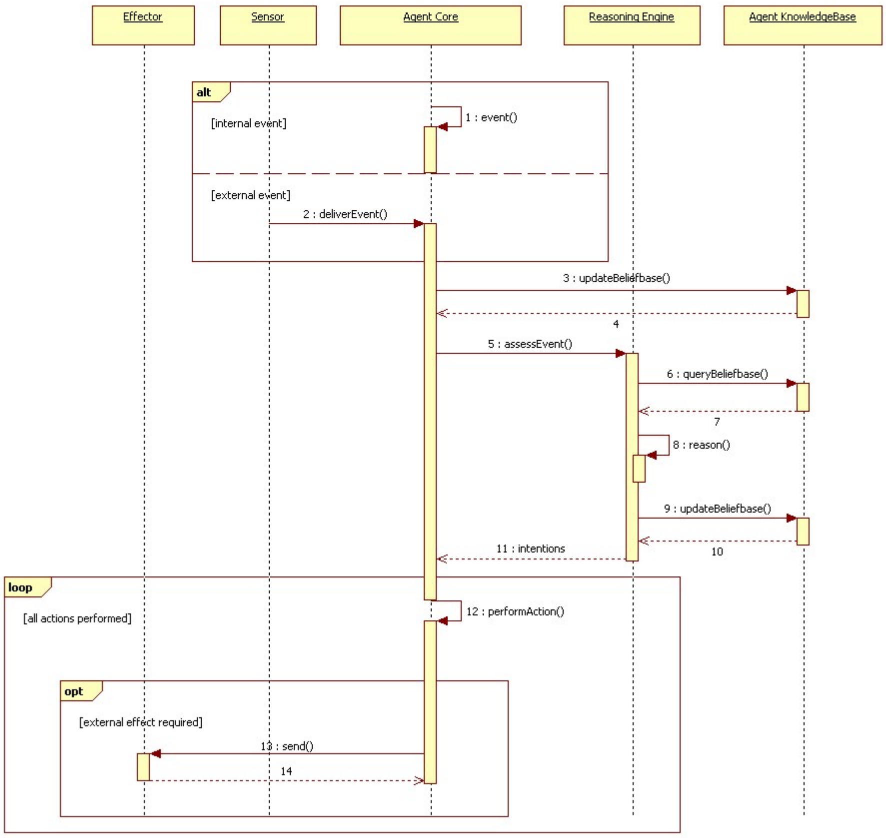

Figure 2: Semantic BDI Reasoning Sequence Diagram

Tools provided by the Semantic Web are embedded into BDI agents with two aims: first, to enhance the reasoning capabilities of the agents, and secondly to enhance the interoperability with other parts of the “semantic world”. Conceptual definitions of service parameters, metrics, and economic values as well as resource characteristics are given using OWL DL [5]. When agents receive data via their sensors including agent communication, this data is inserted into the OWL DL model describing the agent’s beliefs. These beliefs are automatically enhanced via the DL reasoning process. For example, in case of a resource break-down, the current problem can be mapped onto the existing semantic classification of problem types, which may determine the necessary actions to be taken or can be used to estimate the time needed for repair. Furthermore, the history of previous resource usage and previous resource errors are also available in the knowledge base, and can be used to infer new beliefs regarding potential service execution parameters. Jena is used as an embedded knowledge base and reasoning engine in our experiment. Whenever the standard reasoning with Description Logic is not sufficient, custom rules provided by the built-in Jena rule engine can be used to enhance and customize reasoning within agents.

The use of OWL ontologies for knowledge representation enables to re-use existing knowledge available in the Semantic Web. An agent’s knowledge consists of parts specific to the application domain (e.g. engineering), parts specific to SLA (e.g. quality of service (QoS) metrics) and parts specific to the business goals of the service provider. The BREIN project also aims at providing re-usable ontologies to help concrete applications of the framework in filling their knowledge bases. The Grid Resource Ontology facilitates the description of computational resources using OWL. The core ontology of QoS can be used to describe agreed service parameters in SLAs, either using the provided ontology directly or through required extensions. Finally, the BREIN project will also provide a core Business Ontology, which will cover business related concepts (such as product, price, pricing scheme, business policy etc.) and related technical terms (such as negotiation,

ISBN-978-963-9058-25-5

requirements, matching etc.). The Business Ontology will allow clients to compare different service providers who use different terms to describe their SLAs. This enables a "guideline" for existing business scenarios to "virtualize" their business by mapping their concepts to the commonly understood Business Ontology.

APPLICATION SCENARIO

As a tool for validating the idea and the developed framework, we present an application scenario in this section. The scenario is in the field of virtual engineering, and enables teams of engineers to collaborate remotely in the design process using simulation and visualization of engineering processes. Techniques such as Design Optimization or Design of Experiments (DOE) require large series of carefully planned simulations, which are executed on special simulators (software or hardware). As these simulations take long time and the necessary hardware and software are expensive, the optimal use of resources is essential for the service provider, and the fulfilling of SLAs is of high importance for both the customer as well as the service provider. Therefore, we utilize agent-based adaptation to provide goal-driven information system support to reach the above mentioned business goals.

In the validation scenario resources such as computers running a given simulation software are represented by Resource Agents. Task Agents represent complex engineering tasks given by the customer which have to be executed. The SP Agent enact higher level business goals of the service providers. They are responsible for negotiating SLAs for task executions with customers and internal scheduling negotiations for task executions by Resource Agents.

We assume an established SLA for task execution as an example. The Resource Agent executing the task anticipates that the task will not be finished in the time guaranteed as part of the SLA. It notifies the SP Agent accordingly, which decides to move the task to another Resource Agent. Therefore, the SP Agent announces the task description and the required SLA parameters among its Resource Agents. Each Resource Agents provides an offer to the SP Agent to execute the task. This offer is created based on the internal knowledge base of the Resource Agents. Thus, the Resource Agent can calculate with expected free slots and estimated performance of the resources. The offers are evaluated by the SP Agent, using its internal knowledge about SLAs and related penalties for not fulfilling the SLA guarantees. For example, reverse multi-attribute combinatorial auctions can be used as the protocol for allocating tasks to resources, taking the required service level into account. The SP Agent finally selects one offer, and moves the job to another Resource Agent.

CONCLUSIONS

A framework was presented using agents and semantic reasoning to facilitate SLA fulfillment in complex task execution environments such as engineering or manufacturing by providing the required adaptivity. The presented approach offers advantages when centrally pre-planned execution is not applicable, or there are often deviations from previously created plans, and thus adaptation is needed on-the-fly. The proposed solution integrates BDI agents with a reasoning engine, enabling goal-driven behavior based on Description Logic reasoning, and the re-use of existing OWL ontologies. The presented solution also offers an agent-based approach for managing SLAs on a semantic basis, which is more flexible and distributed than traditional deterministic and centralized approaches.

ACKNOWLEDGEMENTS

This work has been supported by the BREIN project (http://www.gridsforbusiness.eu) and has been partly funded by the European Commission's IST activity of the 6th Framework Programme under

ISBN-978-963-9058-25-5

contract number 034556. This paper expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this paper.

REFERENCES

[1] Dietrich, A. J.; Kirn, S.; Sugumaran, V.: *A Service-oriented Architecture for Mass Customization - A Shoe Industry Case Study*. In: *IEEE Transactions on Engineering Management*, 54(1), 2007, S. 190-204.

[2] Karaenke, P.; Kirn, S.: *Service Level Agreements: An Evaluation from a Business Application Perspective*. In: Proceedings of the eChallenges e-2007 Conference, 24-26 October, The Hague, Netherlands, 2007

[3] Open Grid Forum (OGF): *Web Services Agreement Specification (WS-Agreement)*, OGF Proposed Recommendation GFD.107, 2007.

[4] Wooldridge, M.: *Intelligent Agents*. In: Weiss, G. (ed.): Multiagent Systems: *A Modern Approach to Distributed Artificial Intelligence*, 3rd edition, MIT Press, Cambridge, 2001, pp. 27-77.

[5] W3C: *Web Ontology Language (OWL)*, 2004, http://www.w3.org/2004/OWL/ (accessed 31.10.2008).

ISBN-978-963-9058-25-5

ISBN-978-963-9058-25-5