Agent-supported Service Management and Monitoring for Flexible Inter-enterprise Cooperation

András Micsik[1], Paul Karaenke[2]
[1]MTA SZTAKI, Budapest Lágymányosi u. 11. H-1111, Hungary, micsik@sztaki.hu
[2]University of Hohenheim, Information Systems 2,
Schwerzstr. 35, 70599 Stuttgart, Germany, karaenke@uni-hohenheim.de

Abstract

Service-oriented architecture (SOA) can be used to facilitate technical interoperability over organizational boundaries. This can be extended towards the support of cross-organizational cooperation in heterogeneous distributed environments. This paper presents concepts and an architecture aiming at enhanced service use through the help of multiagent technology. The architecture provides intelligent management and monitoring of services as well as agent-supported cooperation in service provision.

Keywords: multiagent systems, service-oriented architecture, cooperation, monitoring

1 Introduction

The main objective of the service-oriented architecture (SOA) paradigm is to facilitate technical interoperability over organizational boundaries, representing real-world business relationships on the technical information system level. The challenge is to provide means for cross-organizational cooperation in heterogeneous distributed environments on the technical access level as well as integration and interoperability with existing service-oriented systems. SOA is the currently preferred paradigm for the delivery of services in open, dynamic, uncertain environments without centralized control. Typical SOA solutions provide remote and secure access to services offered by an enterprise, but still the usability of such SOA environment requires more, for example: (1) monitoring of service usage, (2) intelligent management of services, and (3) cooperation in service provision.

In this paper, we present a generic framework which is able to provide additional benefits for service-oriented architectures using multiagent technology. In addition, we utilize formal semantics to tackle inter-organizational terminological heterogeneity. The framework combines several techniques and paradigms to achieve such an integrated functionality.

2 Service Monitoring and Management

Monitoring services provides the basic information about service availability and alerts in case of service failures. Beyond that, monitoring can provide valuable data for the scheduling and performance planning of services. While simple services may be available in abundance, services in supply chains or high-performance computing may be limited or may require several resources. In such cases the monitored performance can be used for management of service provisioning.

Management of services may also reach further than stopping and starting services. Each service request implies a certain amount of required resources for its completion. The assignment of resources to service requests is an activity which is orthogonal to the service execution. For simple services, the resource requirements may be static and known a priori. In more complex scenarios, resource requirements may be dependent of the concrete input to the service request and may vary over the service's runtime. Then, monitoring of the resource utilization is required to anticipate the actual short-term resource requirements for the completion of all service requests. If the anticipated resource requirements exceed the service provider's own available resources, it might be preferable to cooperate with other service providers to be able to answer all service requests in time.

There are two typical types of cooperation in service provision, when different service providers are working together for the provision of a service. The first type of cooperation is resource sharing or outsourcing as outlined above, where different service providers have to agree on the conditions of consuming other's resources for service provision. The second type of cooperation is service composition, where the required functionality is achieved by several services acting together. Both types of cooperation require a coordination of the activities of the involved parties.

The framework developed to support the above mentioned tasks of service monitoring and management uses multiagent technology as the basic enabling technology, but it also introduces some novel features. In the following sections, the novelties of our architecture are highlighted and explained in more detail.

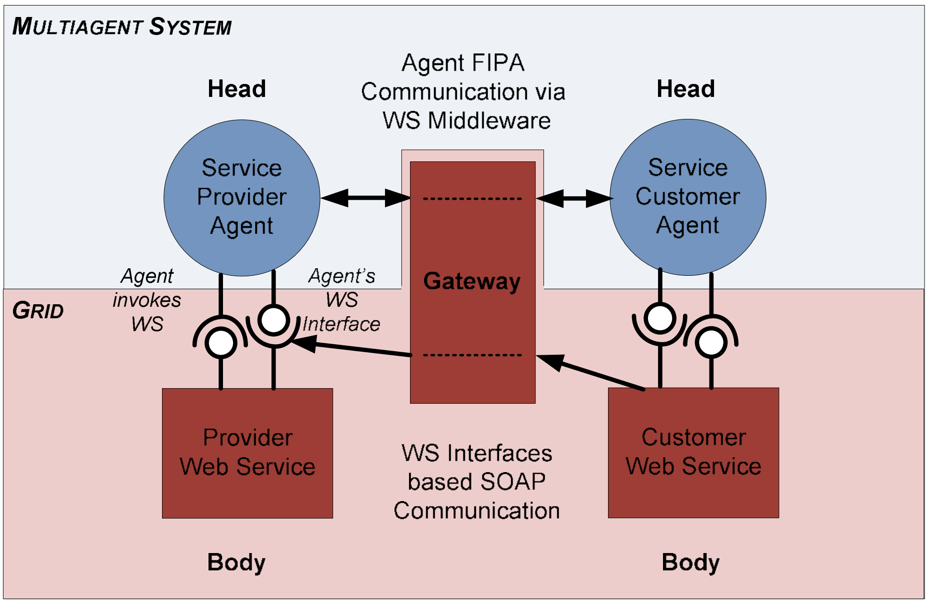

Figure 1: Head Body Architecture in BREIN

3 Head-body Paradigm

The head body paradigm is used as the leading metaphor for the architecture: it implies a conceptual separation of a software agent into two parts – head and body. The agent's head is used for interactions with other agents being member of the agent society. This includes reasoning about interactions such as participating in cooperative processes for problem solving. The body is encapsulating any other (domain) functionality of an agent. The head body paradigm is used as follows (Figure 1): web service (WS) resources that are represented by agents are part of the body. The agent's core capabilities are implemented in the head; i.e., interactions and especially coordination with other agents in the agent society.

On the other hand, the agent body has a set of web service operations as effectors, which are able to control some services providing core business. The agent has full control over the WS and communicates with it via agent-to-WS and WS-to-agent mechanisms presented in the next section. A web service which is represented by an agent can transparently be invoked by other web services, respectively clients. The agent can evaluate the invocation requests and can reason if an invocation of the encapsulated WS is in accordance to its own goals. If the invocation request is opposed to the goals, the agent can intercept the invocation and the encapsulated WS is not invoked. In addition, the agent is able to gather monitoring information regarding service execution.

4 Communication between Agents and Web Services

Typical WS-to-WS communication is extended towards agent-to-WS, WS-to-agent and SOAP-based agent-to-agent communication. These mechanisms enable agents to poll and manage web services and enable web services to inform and notify agents. In addition, the SOAP-based agent-to-agent communication enables agents to communicate securely across organizational domains. In the center of Figure 1 the Gateway component is shown. This component (or several instances thereof) protects the organizational boundaries on both outgoing and incoming communication. In case of incoming communication requests, the authority and rights of the requesters are checked. For outgoing communication, the authorization to send information to a third party is verified. These mechanisms ensure the security of information and service access on a finer scale than traditional IP-based firewalls, and this kind of solution is expected in real business scenarios. Therefore, all communication between parties is made uniform in the form of SOAP and routed through Gateway components in our approach.

Uniform message handling makes system administration tasks easier and provides additional possibilities for messaging. The problem is that agents typically use Agent Communication Language (ACL), for which SOAP is not yet supported. This creates a gap between agent and web service communication. In order to bridge this gap, we implemented a specific solution called Message Transport Protocol (MTP) for SOAP [Micsik et al, 2009]. For multiagent systems, FIPA is the organization that produces standards. FIPA provides specifications in several subject groups, among which the relevant groups for our topic are Agent Communication and Agent Message Transport. The SOAP MTP [SOAP MTP add-on, 2008] has been implemented as an add-on to the Jade agent platform, as MTP functionality is well separated in Jade. The SOAP MTP provides a SOAP-based communication which is transparent for agents, so ACL messages are converted to SOAP messages and backwards in the background. This enables agents to communicate through the Gateway components as web services do. The implementation of the SOAP MTP in Jade configures a WS server and a WS client at startup, having separate buses, so they can be configured independently. This is advantageous when messages between the client and the server have to go through a gateway which does not leave SOAP messages intact.

5 Evaluation

The SOAP MTP has been evaluated in a use case scenario of the BREIN IST project [BREIN, 2009]. BREIN attempts to bring recent Grid research results closer to business applications and also to enhance e-business environments with agent and semantic technologies. The use case contains several service providers competing for orders from the customer. Each service provider has a service-based architecture for daily business and an agent community for monitoring and adaptation purposes. Agents record the monitored data about availability and behavior of web services. When a failure of the service is detected by an agent, the agent can initiate an adaptation process to re-schedule the running tasks on currently available resources. Communication among all parties is secured by the Gateway component, which transfers SOAP messages between involved parties using WS-Security and WS-Addressing. WS-Addressing is used to identify the underlying business agreement, so the BREIN Gateway can accept or refuse to forward service requests.

As a result of the previous enabling technologies, utilization of multiagent coordination mechanisms becomes applicable in existing SOA environments. Certain agents of the environment (as heads) are able to use negotiation techniques to agree on service provisioning or on resource consumption. In the BREIN airport ground handling scenario agents represent coordinators, service providers and resources, and reverse combinatorial multi-attribute auctions are used to find the most appropriate services to be used for executing ground handling tasks.

6 Related Work

The coupling of agents and WS resources in a similar approach is investigated in [Negri et al, 2005], though the authors remain on a very high level of abstraction and do not consider agent-to-agent communication based on WS-technology. In [Bunruangses et al, 2004], a centralized agent manager is proposed for controlling agents representing grid resources. The approach focuses on service discovery and sketches the idea of agent-to-agent communication via SOAP without giving technical details. Due to the complexity of grid systems, it is impossible to define a central, system-wide performance matrix and management policy [Ferguson et al 1996]. Tianfield et al [Tianfield et al, 2005] propose a decentralized approach without giving implementation details. For example, agents are assumed to provide a WS interface without giving details about the realization. Cao et al. [Cao et al, 2002] propose ARMS, an agent-based grid resource management system. The approach is limited to high performance computing (HPC) resources and all resources are considered to be able to execute each job; i.e., explicit semantics for resource descriptions are not considered. In their work, [Rochford et al, 2006] present an agent-based monitoring system built for monitoring almost 200 services of Grid-Ireland. The role of agents is solely information collection in a mostly centralized way, with minimal cooperation among agents. Load balancing of resources using agents has several results, for example [Galstyan et al, 2004] uses reinforcement learning, where agents continuously model the efficiency of resources. [Cao et al, 2005] utilizes application performance prediction algorithms inside agents for load balancing. These approaches do not consider the dynamic negotiation aspect of scheduling, when the outcome depends not only on the past experience but also on current conditions defined dynamically by other parties (e.g., outsourcing scenario).

7 Conclusion

By integrating the worlds of web services and multiagent systems, this architecture offers several benefits for both agents and SOA. For agents the benefits include better accessibility via SOAP, management, and security. For SOA the benefits are in the application of intelligent techniques implemented in multiagent systems and usable for monitoring, management and cooperation of web services. Thus, the architecture enables to facilitate the advantages of both, multiagent and WS, technologies in a single environment.

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 contract number 034556.

References

[BREIN, 2009] BREIN project, http://www.eu-brein.com/

[Bunruangses et al, 2004] Bunruangses, M.; Poompattanapong, W.; Banyatnoparat, P.; Piyatamrong, B.: QoS Multi-Agent Applied for Grid Service Management. In: Proceedings of the 2004 international symposium on Information and communication technologies, Las Vegas, Nevada. June 16th-18.th, 2004. Computer Science Press, Dublin 2004, 74-79.

[Cao et al, 2002] Cao, J.; Jarvis, S. A.; Saini, S.; Kerbyson, D. J.; Nudd, G. R.: ARMS: An agent-based resource management system for grid computing. In: Scientific Programming, Special Issue on Grid Computing 10 (2), 135-148.

[Cao et al, 2005] Cao, J.; Spooner, D. P.; Jarvis, S. A.; Nudd, G. R.: Grid load balancing using intelligent agents. Future Generation Computer Systems, Volume 21, Issue 1, 1 January 2005, Pages 135-149, ISSN 0167-739X, DOI: 10.1016/j.future.2004.09.032.

[Ferguson et al 1996] Ferguson, D. F., Nikolaou, C., Sairamesh, J., and Yemini, Y.: Economic models for allocating resources in computer systems. In Market-Based Control: A Paradigm For Distributed Resource Allocation, S. H. Clearwater, Ed. World Scientific Publishing Co., River Edge, NJ, 1996, 156-183.

[Galstyan et al, 2004] Galstyan, A.; Czajkowski, K.; Lerman, K.: Resource Allocation in the Grid Using Reinforcement Learning. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, July 19-23, 2004, New York, Volume 3. ACM Press, New York 2004, 1314-1315.

[Karänke et al, 2009] Karänke, P.; Micsik, A.; Kirn S.: Adaptive SLA Management along Value Chains for Service Individualization, International Symposium on Services Science, Leipzig, March 2009.

[Micsik et al, 2009] Micsik, A.; Pallinger, P.; Klein, A.: Soap based message transport for the jade multiagent platform. The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009) Industry track, Budapest, Hungary, May 10-15, 2009.

[Negri et al, 2005] Negri, A.; Poggi, A.; Tomaiuolo, M.: Intelligent Task Composition and Allocation through Agents. In: Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05), 2005, pp. 255-260.

[Rochford et al, 2006] Rochford, K.; Coghlan, B.; Walsh, J.: An Agent-Based Approach to Grid Service Monitoring. Proceedings of The Fifth International Symposium on Parallel and Distributed Computing (ISPDC), 2006, 345-351.

[SOAP MTP add-on, 2008] SOAP Message Transport Protocol Add-on for Jade Multi-Agent Platform, http://brein.dsd.sztaki.hu

[Tianfield et al, 2005] Tianfield, H.: Towards agent based grid resource management. In: Proceedings of the 2005 IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2005), Cardiff, Wales, May 9-12, 2005, Volume 1. IEEE Computer Society Press, Los Alamitos, 2005, 590-597.