MAMS Service Framework

(Extended Abstract)

Thomas Konnerth
Thomas.Konnerth@dai-labor.de

Silvan Kaiser
Silvan.Kaiser@dai-labor.de

Alexander Thiele
Alexander.Thiele@dai-labor.de

Jan Keiser
Jan.Keiser@dai-labor.de

DAI Labor
Technische Universität Berlin
Ernst-Reuter-Platz 7, 10587 Berlin

ABSTRACT

In this paper we describe the service platform for the MAMS framework which is an agent based platform for the creation, deployment and execution of service compositions created by non-IT-experts. Our platform provides a flexible service execution environment, which utilises agent technology to improve scalability, platform & service management and stability.

1. INTRODUCTION

Nowadays, even though technologies for global provisioning of services are well established, it is still difficult for small and medium enterprises to act as service providers. The reasons for this come from the need for technical know-how to create services and the necessary infrastructure to provide the created services.

The project MAMS (Multi Access - Modular Services) [5] addresses these issues by allowing non-technical persons to quickly and easily create, deploy and manage services, according to the users needs. Platform providers as technical experts can concentrate on the provisioning of tools and basic services for service creation, and on the infrastructure for reliable hosting of the services.

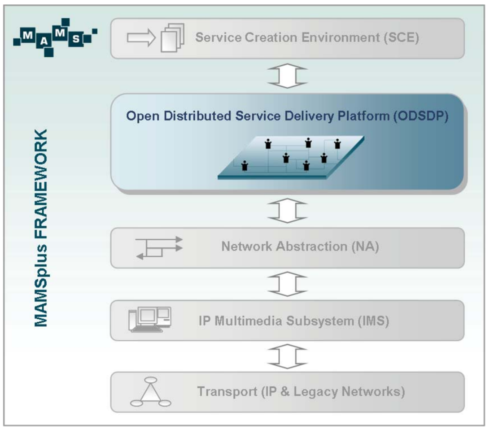
With the incorporation of agent technology, we have realized an open distributed service delivery platform (ODSDP) as part of the overall MAMS framework, which enables adaptability to dynamic environments.

2. MAMS PROJECT

The development of new services for telecommunication applications and other IT systems is one of the most important vehicles of innovation for telecommunication and other service providers. Competition and ever growing expectations of users require a faster service development and the means to deliver those services promptly. The MAMS project employs graphical service composition and agent-based approaches as answers to these problems.

First, a visual service creation environment for non-IT-expert users has been developed that allows the easy composition and deployment of new services. Thus, the non-experts become developers and can apply their creativity and domain knowledge to literally construct new services and applications.

Second, the development tools and the service execution environment are integrated into a unified framework - the MAMS Service Framework. This allows a smooth deployment of the developed service compositions and enables further acceleration of the development process. The service execution environment as well as the development tools offer integration of up-to-date technologies and infrastructures. Figure 1 shows that the MAMS framework offers a complete vertical integration of all layers, from the service creation over execution down to the network layer.



MAMS
Service Creation Environment (SCE)
Open Distributed Service Delivery Platform (ODSDP)
MAMSplus FRAMEWORK
Network Abstraction (NA)
IP Multimedia Subsystem (IMS)
Transport (IP & Legacy Networks)

Figure 1: MAMS Framework

2.1 Agent Technology in MAMS

The enabling technology for integration of the different components of the MAMS Framework is a multi agent system (MAS). This MAS works as a service execution environment, and is called Open Distributed Service Delivery Platform. Using a multi agent system for service execution environment has several advantages.

First, when used as an organisational and functionally closed entity, agents provide an intuitive concept for mapping functionalities such as basic services or service compositions to logical entities (i.e. agents). Based on this mapping, it is now possible to provide each service stakeholder with a set of introspective and — if applicable — management features that can be tailored to the respective service. Further-

Cite as: MAMS Service Framework, (Extended Abstract), Thomas Konnerth, Silvan Kaiser, Alexander Thiele, Jan Keiser, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1351–1352

more, through this organisational service-to-agent mapping, the system becomes highly scalable by the agents potential to be distributed.

Another important aspect of agents is the ability for adaptive behaviour. It has already been shown that agent technology and especially multi agent systems are well suitable for creating systems that incorporate adaptive behaviour [3, 2, 4]. We think that adaptive agents can also provide reasonable features in the context of service execution and thus enrich the MAMS service execution environment.

2.2 Platform architecture

In the MAMS framework we distinguish between two different notions of services. The first is the *basic service* which is an atomic building block for the modelling of user generated service compositions. These building blocks are typically equivalent to web service operations, i.e. they provide a single functionality which is easy to understand and can be applied in various contexts.

The second notion is that of a *service composition* that is composed from basic (or other) services by the non-IT-expert. Once such a service composition is complete, it can be deployed on the ODSDP and is then available to the service users.

The service creation process starts with graphical Service Description Language (SDL) which is a data flow oriented modelling language that can be used by the non-IT-expert for service compositions. From this, we generate a BPEL based intermediate model, which is used for storage and compatibility. Finally, we translate this to a Declarative Formal Language (DFL) which is interpreted by the agents and includes ontology based declarative service descriptions which allow the application of reasoning and service matching to the composition.

While the intermediate model can be executed with a normal BPEL-engine, it also contains OWL-S [1] descriptions for all used basic services. Thus, in the DFL we can replace the WSDL-call with an abstract call to the OWL-S description. The DFL script is then deployed into an agent on the ODSDP that is able to execute this script. However, as that agent now has a semantic description of the basic services that will be executed, it is able to apply reasoning and service matching to the service composition in order to optimize the service execution.

For example, the composition may only contain a call to an abstract *messaging-service* that states that a message is to be delivered to a certain user. The agent is then able to decide at runtime, which mechanism can be used to deliver the message, e.g. SMS or eMail, depending of the available addresses and preferences of the user.

The runtime environment of the agents is the ODSDP. It consists of an arbitrary number of agent nodes which are interconnected. This allows an organisational as well as a physical distribution of the agents that provide the services.

The platform uses an adaptive load balancing method applying two basic load balancing strategies. A simple load distribution manager starts new agents on the least loaded node of a platform. Additionally, Scheduler Agents monitor load on their local Agent Nodes. They act on very high local system load by migrating local agents to other Agent Nodes that are significantly less loaded. This runtime load balancing allows to react to changing workloads during the lifetime of an agent based application.

A MAMS agent is based on a component architecture, which contains two mandatory components: a memory for the storage of agent-internal knowledge and an execution cycle to enable pro-active behaviour. Optional components allow the addition of further generic functionalities such as service matching or execution of service compositions. The interaction between all components inside an agent takes place via the memory according to the blackboard metaphor.

Security mechanisms such as Single Sign-On(SSO) based authentication and authorization ensure that the management interface and the services can only be used by the groups of users which are specified by the responsible service provider or platform provider.

3. RESULTS AND CONCLUSION

During the MAMS project, we implemented two prototypical scenarios from the eHealth and eBusiness domains, in order to verify our approach. Both scenarios were used to evaluate the complete processes defined in MAMS for the creation of services by non-IT-experts, including service composition, deployment and service execution. Ongoing Testing shows that the features of a service delivery platform like the ODSDP can be efficiently based onto a service execution environment with the help of agent-oriented software design. By assigning each new service composition to a single execution agent, runtime deployment and control of service execution could be achieved. We have successfully applied adaptive service execution strategies by including semantic service matching and load balancing mechanisms. We conclude that multi agent technology is of great benefit in the design and implementation of a service execution environment as shown in this paper.

4. REFERENCES

[1] A. Barstow, J. Hendler, M. Skall, J. Pollock, D. Martin, V. Marcatte, D. L. McGuinness, H. Yoshida, and D. D. Roure. OWL-S: Semantic Markup for Web Services, 2004. http://www.w3.org/Submission/OWL-S/.

[2] L. Bozzo, V. Mascardi, D. Ancona, and P. Busetta. COOWS: Adaptive BDI agents meet service-oriented computing – extended abstract. In M. P. Gleizes, G. A. Kaminka, A. Nowé, S. Ossowski, K. Tuyls, and K. Verbeeck, editors, *Proceedings of the 3^{rd} European Workshop on Multi-Agent Systems (EUMAS'05)*, pages 473–484. Koninklijke Vlaamse Academie van Belie voor Wetenschappen en Kunsten, 2005.

[3] I. Dickinson and M. Wooldridge. Agents are not (just) web services: Considering BDI agents and web services. In *Proceedings of the 2005 Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE'2005), Utrecht, The Netherlands*, July 2005.

[4] T. Konnerth, B. Hirsch, and S. Albayrak. JADL — an agent description language for smart agents. In M. Baldoni and U. Endriss, editors, *Declarative Agent Languages and Technologies IV*, volume 4327 of *LNCS*, pages 141–155. Springer Berlin / Heidelberg, 2006.

[5] MAMS. MAMS multi access - modular services. http://www.mams-platform.de/.