SOAP based Message Transport for the Jade Multiagent Platform

András Micsik
MTA SZTAKI
Lágymányosi utca 11.
1111 Budapest, Hungary
micsik@sztaki.hu

Péter Pallinger
MTA SZTAKI
Lágymányosi utca 11.
1111 Budapest, Hungary
pallinger@sztaki.hu

Achim Klein
University of Hohenheim
Information Systems 2
Schwerzstr. 35
70593 Stuttgart, Germany
achim.klein@uni-hohenheim.de

ABSTRACT
The combination of multiagent technology and Web Services has been a challenge since several years, as the flexibility of autonomous software agents is to be enriched with standards based, industrial strength, open and secure Web Service technology. The contribution of this paper is a SOAP-based message transport implementation for the Jade multiagent platform. This message transport is transparent for agents and allows for enacting Web Service specifications on inter-organizational multiagent communication, to facilitate the advantages of both technologies in a single environment. The message transport is evaluated in an industrial setting.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence - *Multiagent systems*

H.3.5 [**Information Storage and Retrieval**]: Online Information Services - *Web-based services*

General Terms
Management, Security, Standardization

Keywords
Multiagent systems, Web Services, Message Transport Protocol, SOAP, WS-Security, WS-Addressing, Agent Communication Language

1. INTRODUCTION
Web Services (WS) provide means for realizing reliable, secure, scalable, and interoperable distributed and service oriented software systems using the web on the basis of well defined and established standards. For this reason, they have been widely used in the IT industry in recent years. Multiagent systems provide another means for realizing open and flexible distributed software systems, focusing on distributed problem solving. Multiagent systems, however, remain almost untouched by the evolving paradigm of web services and service-oriented architecture.

Cite as: SOAP based Message Transport for the Jade Multiagent Platform, András Micsik, Péter Pallinger, Achim Klein, *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. 101 – 104

The objects of investigation in this paper are technologies for integrating both paradigms. The combination of Web Services and Multiagent platforms offers several enhancements regarding interoperability and availability. The numerous, widely accepted recommendations for making communication with web services more secure, reliable and configurable should become applicable to multiagent systems as well. For multiagent systems, FIPA [1] 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 Agent Communication group deals with communication on a higher level than message transport, it defines the message structure and high-level interactions between agents, such as contract net or brokering protocols. The Agent Message Transport group defines ACL (Agent Communication Language) and envelope representations and the transport protocols that can be used to transfer agent messages between hosts. Currently IIOP and HTTP protocols are supported for agent message transport (WAP support is experimental). The problem concerning multiagent and WS integrations is that the use of IIOP and WAP is declining, while the HTTP based MTP (Message Transport Protocol) is incompatible with SOAP, which is the most commonly used base WS transport protocol. To enable the agent world to catch the momentum of emerging WS-* standards and recommendations as well as new transport techniques, we propose a SOAP based message transport for the Jade Multiagent Platform.

The paper has the following structure: we first present related results and solutions, then collect the requirements for the MTP implementation, and describe the actual implementation. Subsequently a use-case scenario is described which has been used for evaluating our approach. Finally, conclusions are provided.

2. RELATED WORK
In recent years much attention has been paid to web services interoperability within the agent community (cf. [15]). FIPA has set up the Agents and Web Services Interoperability Working Group (AWSI WG) [4] to investigate this issue and propose solutions. In 2007 the AWSI WG reported about its evaluation of available technologies and their priorities set [5]. Their first priority is the translation of FIPA ACL to SOAP and vice versa, while interaction protocols supporting agent interaction with Web services (and vice versa) are their second main interest. As the first task is inherently difficult, it created an obstacle for their second goal. While the AWSI WG admits that “standardization in

use of various transport mechanisms would be of great immediate benefit in supporting the interaction of agent systems and Web services" [5], at the time of writing there is still no publicly available solution for using SOAP in agent communication.

In the following some of the solutions proposed to the AWSI WG are presented and evaluated regarding our main goal of enabling WS technology based agent message transport. The approach of Soto falls closest to ours [4], but in this case the contents of FIPA envelopes are mapped to WS-Addressing headers. Furthermore, Soto also converts the message payload to XML using a specific Jade codec. These conversions provide additional potential for misinterpretation, under-specification and other problems disabling the recipient to interpret the message correctly. Furthermore, WS-Addressing is used to convey FIPA headers, thus it is unavailable for use by the transport layer itself. We opt for the much simpler solution presented in this paper, because it provides more flexibility for the transport.

The Web Service Integration Gateway (WSIG) [7] is an official Jade plug-in. Functionally, WSIG is a centralized bidirectional invocation facility, by which Jade agents can execute Web Services and Web Services can execute Jade agent services. WSIG is based on some translation facilities at it its core. Firstly, FIPA ACL [8] messages are translated to SOAP messages and vice versa. Secondly, FIPA ACL service descriptions are translated to Web Service Description Language (WSDL) [9] and vice versa. This is supposed to enable Jade agents to execute a WS as if it were an agent.

WSIG has a different focus in that it wants to connect web services with agents, and not agent platforms via SOAP. Practical examinations by the authors revealed that WSIG is quite difficult to set up and the configuration of ACL message translation to SOAP often requires manual tuning. Furthermore, each agent has to register individually for receiving web service calls. This solution is not very suitable regarding our goal, even using workarounds.

WS2Jade [10][11] is an integration approach focusing on closely coupling Jade agents and Web Services by representing a WS by a gateway agent. Thus, WS2Jade focuses on agents executing WS. Functionally, WS2Jade provides for deploying and controlling Web Services as agent services at run time. A new gateway agent is created automatically for a WS by WS2Jade and the WSDL2Jade tool [12] is used to create a service-specific Jade ontology from the WSDL description of the WS. The ontology describes operations and respective parameters. This solution cannot support a bi-directional agent-to-agent communication via SOAP either, because agents cannot receive messages from web services.

Further approaches, which focus on allowing agent services to be executed by WS (e.g., the Web Service Agent Integration (WSAI) project [13]), rather focus on translating SOAP to ACL instead of using SOAP within agent-agent communication.

Also, approaches for which no implementation is available to our best knowledge, such as the AgentWeb Gateway, proposed by Shafiq et al.[14], are not considered.

3. MESSAGE TRANSPORT PROTOCOL

3.1 Requirements

The requirements for our agent Message Transport Protocol (MTP) are summarized briefly in this subsection. Agent messages inside a platform are delivered internally, so no transport protocol is required. The MTP is only used for sending messages to agents residing in a remote agent platform. Thus, messages to remote agents have to be encapsulated into SOAP messages, and transferred to the remote agent platform, where the same message structure should be rebuilt as on the originator side. Furthermore, we require that message transfer involving SOAP encapsulation should be transparent for the agents beside configuration of addresses. This enables using the SOAP-based MTP very easily in existing multiagent applications.

There have been attempts to transform parts of the agent message to a format more compatible with SOAP, and we presented some of these attempts in the Related Work subsection. Generally, this is the main hindrance of providing an agent-web service interoperability solution which is ready to be taken up by the industry. The FIPA ACL is a complex language for transferring content using semantic descriptions, thus it works on a higher level than SOAP, which transfers plain typed structural data. Any solution to translate between these formats needs to be backward compatible with both standards.

It is a question whether web services should be able to understand agent messages. If not, then it is enough to encapsulate ACL into SOAP messages, which is clearly easier to do.

In our case, there is no direct need for any transformation between FIPA ACL and SOAP. The agent-to-agent communication can be solved even if the transport layers cannot understand the internals of the message. The benefits of transforming the agent message are questionable compared to the difficulties of harmonizing two well-established worlds. However, there is one clear benefit of not transforming the message: the transport layer can use the SOAP headers and body for its internal purposes, which helps to enhance the interoperability between agent platforms.

The requirements for addressing also need clarification. In order to reach agents in Jade, we need the agent's name and a transport specific address. The former is independent of the transport, and unique within an agent platform, thus it can be used to find the recipient inside the platform. The transport specific address is needed to deliver the message to the recipient's platform. This may change when using the SOAP MTP. Therefore, our requirement for message transport is that proper transport specific addresses are to be used in agent messages, which enables delivery and reply for agent messages. The requirement for discovery is that agents register themselves with the proper (SOAP) specific transport address at the Directory Facilitator.

3.2 Implementation

The SOAP MTP has been implemented as an add-on to the Jade agent platform, as MTP functionality is well separated in Jade. The SOAP transport is supported by the Apache CXF open source services framework. Apache CXF helps to build services in a simple way using front-end programming APIs. Services built using CXF can understand a wide variety of message and transport protocols. Internally CXF uses a bus metaphor (the same as Spring framework uses), which can be configured using XML configuration files. Therefore, many adaptations of the transport can be done by changing configuration files only. Deeper adaptations of sending and receiving messages are also possible using so-called interceptors. Interceptors can be dynamically injected into the chain of standard interceptors which assemble outgoing messages or disassemble incoming messages.

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 very useful when messages between the client and the server have to go through a gateway which does not leave SOAP messages intact.

When a new agent message to be delivered is received by the MTP, both the envelope and the payload are prepared to be passed as SOAP request parameters. The address of the remote web service is extracted from the agent message, the WS client is configured with the address and the parameters, and the remote WS is executed via the WS client. The server responds with a single value representing the status of message transfer similarly to HTTP. Possible response values include “200 OK” or an integer between 500 and 599 with an error message. On the receiving side the server passes the message envelope and payload to the message dispatcher, and then the processing of the message is followed by the standard dispatching procedure of Jade.

The process described above can work only if the proper endpoint addresses are known for remote agent platforms. By default each agent platform assembles its own endpoint address automatically using system properties and run-time parameters (such as port number). There is an option to provide the full endpoint address as a run-time parameter, which we call virtual endpoint address. This can be useful when the address to be distributed is different from the address used locally. The endpoint address is a URL, and it is automatically injected into the sender address of any outgoing message. Therefore, an agent platform can always reply to a received message. Agents can also register with their SOAP address at Directory Facilitators so that they can be discovered dynamically.

3.3 Evaluation Scenario

The SOAP MTP has been evaluated in a use case scenario of the BREIN IST project [2]. 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. These two environments have to work together and complement each other. Messages traveling between different service providers or customers are encrypted using WS-Security specifications. Messages between these administrative domains are sent and accepted by the BREIN Gateways of each domain (Figure 1). The BREIN Gateway requires messages to be signed, encrypted using WS-Security and addressed using WS-Addressing. WS-Addressing is used for two purposes: first, it conveys business context information about the current message. By identifying the underlying business agreement, the BREIN Gateway can accept or refuse to forward service requests. Secondly, WS-Addressing is used to create a virtual address layer on top of real addresses. Virtual addresses used in SOAP messages can be translated dynamically to appropriate real services. This enables service fault management, on-the-fly reconfiguration and other advanced solutions to enhance reliability of the service environment. For example, a web service can be moved to a different host very easily, only its virtual address has to be remapped to the new real endpoint address inside the BREIN Gateway.

It is a natural requirement that agent messages should also use the BREIN Gateway for messaging purposes. Uniform transportation of agent and WS messages simplify system administration and enables common mechanisms to be introduced in routing and delivery. This is achieved with the SOAP MTP add-on presented in the paper. Each agent platform uses the SOAP MTP configured with a virtual endpoint address (which is mapped to the agent platform address in the Gateway), and a separate client and server bus configuration for CXF. The client bus is configured to use three interceptors, which in turn inject WS-Addressing headers into messages, enable the use of attachments in messages and finally encrypt and sign outgoing messages. Since the BREIN Gateway of the recipient side decrypts and modifies incoming messages, the SOAP MTP server side needs slightly different configuration, with different interceptors.

The following steps are executed when sending a message to a remote agent platform: the consumer agent addresses the message

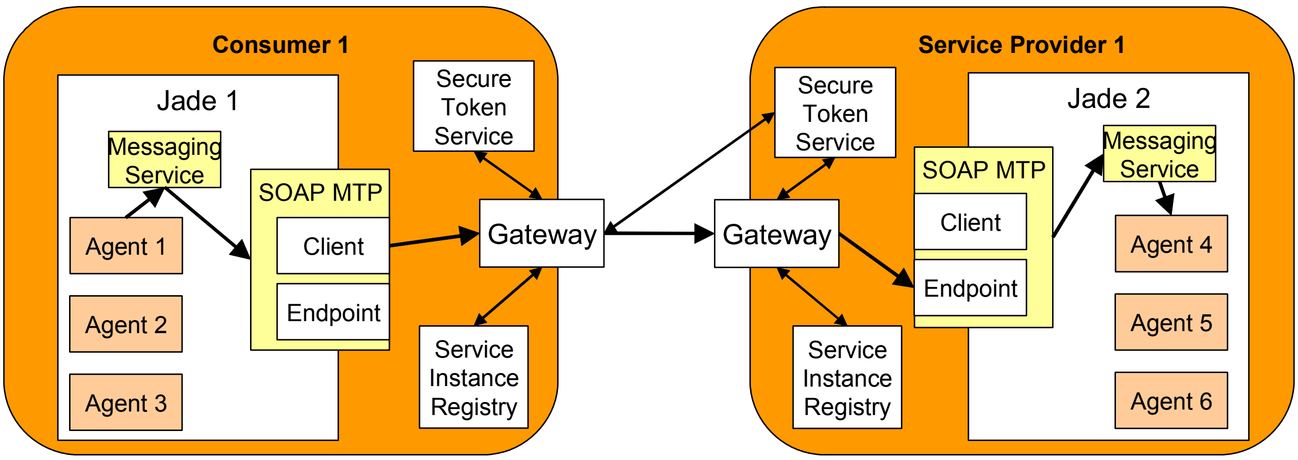

Figure 1. Use case of SOAP MTP in the BREIN project.

using the virtual endpoint address of the remote agent on the Service Provider side (Figure 1). The Messaging Service detects that this address belongs to the SOAP MTP, and forwards the message to the SOAP MTP add-on for delivery. The SOAP MTP client prepares the SOAP message, and Apache CXF delivers it to the virtual address of the remote agent, which is actually caught by the local BREIN Gateway. The local BREIN Gateway identifies the recipient Service Provider using the Service Instance Registry, and arranges for a security token with the Security Token Services of both sides. Then the message is sent to the Gateway at the Service Provider side. The SP Gateway checks the access rights for the service, decrypts the message, then finds the real endpoint service using the Service Instance Registry, and calls the endpoint of the Jade platform. The SOAP MTP of Jade Platform 2 reconstructs the original agent message and passes it to the Jade Messaging Service, which finally delivers it to the recipient agent.

The setup described above has been successfully tested within the scenario. The scenario demonstrates that using the current solution based on WS technology, agent messages can be transferred in a secured way, agent messages can be routed through gateways, and agent addressing can be virtualized (the agent platform can be dynamically relocated to a different address).

4. CONCLUSION

This paper proposes a SOAP-based Message Transport Protocol add-on which enables the Jade multiagent platforms to use SOAP for the purpose of inter-platform messaging. In the evaluation scenario we used the add-on to connect agent platforms via WS gateways using SOAP, WS-Addressing and WS-Security.

By integrating the worlds of web services and multiagent systems, this add-on offers several benefits for agents regarding accessibility, management, and security. First, the accessibility of agent platforms can be enhanced as SOAP based transport can be more tolerant with firewalls and other security restrictions. Secondly, heterogeneous web service and agent environments may use a homogeneous message transport layer which reduces the complexity of system administration. It also enables secure (inter-organizational) transfer of agent messages between agent platforms, thus facilitating the advantages of both, multiagent and WS, technologies in a single environment.

Our future plans include extending the presented add-on towards supporting communication between agents and Web Services. The add-on is available for the Jade users' community at the following URL: http://brein.dsd.sztaki.hu/JadeSoapMTP

5. ACKNOWLEDGMENTS

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.

6. REFERENCES

[1] FIPA - The Foundation for Intelligent Physical Agents. [Website] [cited 2008-12-08] Available from: http://www.fipa.org/.

[2] Jade - Java Agent DEvelopment Framework. [Website] [cited 2008-12-08] Available from: http://jade.tilab.com/.

[3] BREIN project homepage. [Website] [cited 2008-12-08] Available from: http://www.gridsforbusiness.eu/.

[4] FIPA Agents and Web Services Interoperability Working Group (AWSI WG). [Website] [cited 2008-12-08] Available from: http://www.fipa.org/subgroups/AWSI-WG.html

[5] Greenwood, D., Lyell, M., Mallya, A., and Suguri, H. 2007. The IEEE FIPA approach to integrating software agents and web services. In Proceedings of the 6th international Joint Conference on Autonomous Agents and Multiagent Systems (Honolulu, Hawaii, May 14 - 18, 2007). AAMAS '07. ACM, New York, NY, 1-7. DOI= http://doi.acm.org/10.1145/1329125.1329458

[6] Soto, E. L. 2007. Agent Communication Using Web Services, a New FIPA Message Transport Service for Jade. P. Petta et al. (Eds.): MATES 2007, Springer LNAI 4687, 73–84.

[7] Web Services Integration Gateway (WSIG). [Website] [cited 2008-12-07] Available from: http://jade.tilab.com/community-addons.php.

[8] FIPA Agent Communication Language Specifications. [Website] [cited 2008 2008-01-10] Available from: http://www.fipa.org/repository/aclspecs.html.

[9] Web Services Description Language (WSDL). 2001. [Website] [cited 2008-12-07] Available from: http://www.w3.org/TR/wsdl.

[10] Nguyen, X.T. Demonstration of WS2Jade. in AAMAS. 2005. Utrecht, Netherlands: ACM.

[11] Nguyen, X. T. 2005. Demonstration of WS2JADE. In Proceedings of the Fourth international Joint Conference on Autonomous Agents and Multiagent Systems (The Netherlands, July 25 - 29, 2005). AAMAS '05. ACM, New York, NY, 135-136. DOI= http://doi.acm.org/10.1145/1082473.1082820.

[12] WSDL2Jade. [Website] [cited 2008-12-07] Available from: http://sas.ilab.sztaki.hu:8080/wsdl2agent/.

[13] Web Services Agent Integration Project. [Website] [cited 2008-12-07] Available from: http://wsai.sourceforge.net/.

[14] Shafiq, M. O., Ali, A., Ahmad, H. F., and Suguri, H. 2005. AgentWeb Gateway - a middleware for dynamic integration of Multi Agent System and Web Services Framework. In Proceedings of the 14th IEEE international Workshops on Enabling Technologies: infrastructure For Collaborative Enterprise (June 13 - 15, 2005). WETICE. IEEE Computer Society, Washington, DC, 267-270. DOI= http://dx.doi.org/10.1109/WETICE.2005.16

[15] Foster, I., Jennings, N. R., and Kesselman, C. 2004. Brain Meets Brawn: Why Grid and Agents Need Each Other. In Proceedings of the Third international Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1 (New York, New York, July 19 - 23, 2004). International Conference on Autonomous Agents. IEEE Computer Society, Washington, DC, 8-15. DOI= http://dx.doi.org/10.1109/AAMAS.2004.78