

# PROVIDING MESSAGING INTEROPERABILITY IN FIPA COMMUNICATION ARCHITECTURE

Heikki Helin<sup>1</sup> and Stefano Campadello<sup>2</sup>

<sup>1</sup> *Sonera Corporation*

*P.O.Box 970, FIN-00051 Sonera, Finland*

Heikki.j.Helin@sonera.com

<sup>2</sup> *Nokia Research Center*

*P.O.Box 407, FIN-00045 Nokia Group, Finland*

Stefano.Campadello@nokia.com

**Abstract** We describe an on-going technical work done by FIPA standardization organization in the field of agent communication between heterogeneous FIPA agent platforms. The goal of this work is enabling flexible agent communication while providing sufficient interoperability. The flexibility is achieved by introducing several options for different layers of communication. Interoperability is assured by messaging gateways translating between incompatible options.

**Keywords:** Software Agent Technology, Agent Communication, Gateways, FIPA

## 1. INTRODUCTION

In a distributed system communications is an essential component. This is also true in the software agent systems where multiple agents are involved. In order to exchange the knowledge, agents should be able to communicate with each other. In the lower layers, the agent communication does not necessarily differ from the communication in traditional distributed systems. In fact, same transport protocols and messaging techniques as in modern distributed systems should be used. From the lower-layers point of view, agents are just sending data. What makes the software agent communication different from the communication in the traditional distributed systems is the usage of the agent communication languages (ACLs). Typically, the ACLs are based on a speech act theory: messages are actions—communicative acts—as they are intended to perform some action by virtue of being sent. In this paper we assume that agents do communicate with each other, using some ACL. Further, we assume that, at some level, the communication fails due to certain incompatibilities.

FIPA<sup>1</sup> is a non-profit standardization organization promoting development and specification of generic agent technologies. FIPA has specified several communication options in order to enable flexible communication in environments with different characteristics. For example, wireless environments are taken into account in FIPA communication model. Since wireless environments typically have significantly different characteristics than wireline environments, most of the communication layers have an option tailored for wireless environments.

Having various options obviously decreases the direct interoperability, as the message originator cannot assume that the destination understands the protocols and encoding the sender uses. In order to achieve reasonable interoperability between domains using different communication means, interoperability gateways can be used. These gateways translate between message transport protocols and encoding of message components where direct end-to-end interoperability is impossible, impractical or undesirable.

Mediators and gateways have been used in various architectures. In WAP architecture [13], the WAP-gateway translates between various layers in the WAP communication stack and Internet protocols. In the CORBA architecture, similar gateways are called “half-bridges” [12]. Additionally, application level proxies or gateways are used in many architectures where wireless and wireline environments are combined (see for example [10, 11]).

The rest of this paper is structured as follows. In Section 2 we give an overview of messaging in the FIPA communication architecture. Section 3 presents the concept of FIPA messaging interoperability gateway. Finally, Section 4 concludes the paper.

## 2. MESSAGING IN FIPA ARCHITECTURE

At the heart of the FIPA’s model for the agent systems is agent communication, where agents can pass semantically meaningful messages to one another. Figure 1 depicts layered model of the FIPA agent communication. The transport protocol layer and Message Transport Protocol (MTP) layer provide together basic messaging between agents or agent platforms. These layers are not independent. In FIPA, there are three options for MTP: IIOP [7], HTTP [6], and WAP [8]. Each of the MTPs implicitly or explicitly also defines the transport protocol. The message envelope layer provides the communication stack with MTP independent message delivery information (e.g., how message should be routed, etc.). For the message envelope, three encoding options are specified: XML [4], bit-efficient [5], and one which concrete representation is defined in terms of an IDL interface [7]. The ACL layer defines both the semantics and

---

<sup>1</sup>Foundation for Intelligent Physical Agents; <http://www.fipa.org>

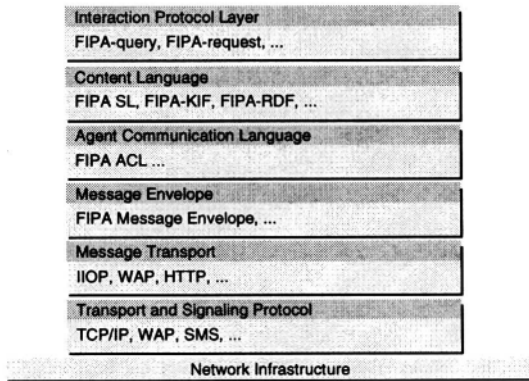


Figure 1. Communication layers in FIPA architecture

the syntax for ACL messages. FIPA has defined three encoding options for its ACL [1, 2, 3]. FIPA-ACL defines only the outer language used in communication, but not the actual content of the message. For example, on the ACL level, the sender defines the type of the message, for example “request”, but does say nothing about the action, which should be performed by the receiver. For this purpose, FIPA has specified several content languages (see [9] for details). Lastly, agent communication typically falls into common patterns (conversation layer). In the FIPA specifications, these are called interaction protocols. An interaction protocol defines a common pattern of conversations used to perform a task.

For successful direct end-to-end interoperability both the sender and the receiver should agree on the MTP and the encoding of various message components. Given that there are three options for MTP-, message envelope-, and ACL-layers, there are in total 27 combinations that can be used. The choices for the content language increase the number of combinations even more. In practice, however, the situation is better. For example, if HTTP is used as MTP, the message envelope and the ACL are typically encoded using XML.

### 3. MESSAGING INTEROPERABILITY GATEWAYS

In some cases direct end-to-end interoperability is impossible, impractical or undesirable. Obviously it is impossible when communicating platforms/agents do not support any common message transport protocol or encoding of FIPA-message component. Direct end-to-end interoperability might be impractical, for example, when communicating over a slow wireless link and the peer at the fixed network does not support message transport protocol suitable for wireless links.

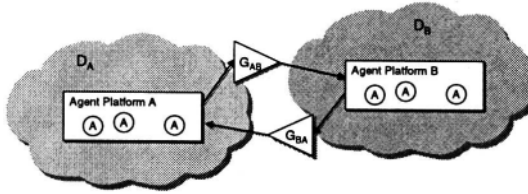


Figure 2. Gateways between incompatible domains  $D_A$  and  $D_B$

Messaging interoperability gateways are needed in order to provide sufficient interoperability in different communication layers using different protocols or encodings. In this paper, we are interested in messaging gateways, but there might other type of gateways as well (e.g., gateways that translate between heterogeneous directory services).

Messaging interoperability gateways are logically situated between agent platforms belonging to different domains. Figure 2 depicts two agent platforms  $A$  and  $B$  belonging to domains  $D_A$  and  $D_B$  respectively. They employ the gateway  $G_{AB}$  in communication from  $A$  to  $B$  and the  $G_{BA}$  in communication from  $B$  to  $A$ .

Logical reference model of a gateway comprises four levels (Figure 3). A gateway on a given level is defined as a function that translates from a source protocol or encoding to a target protocol or encoding, respectively. At the lowest level of the reference model, translation between message transport protocols is performed. On the second level, the translation between concrete representation of the message envelope is performed. Similarly, third and fourth levels are for the ACL and the content language translations, respectively. Additionally, the gateway might perform translations between representations of the application data. This, however, is not a concern of FIPA. A gateway implementing the translation function from  $a$  to  $b$  is not required to implement the inverse function, although this might be the typical case.

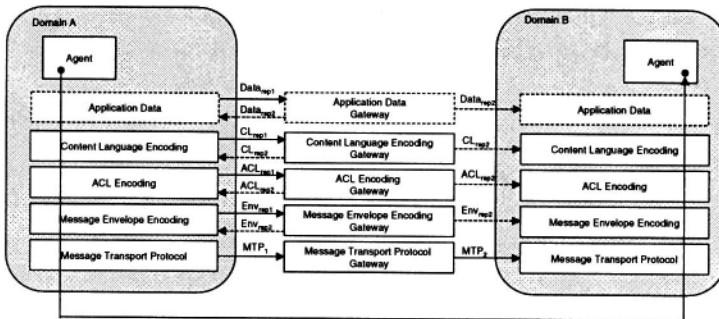


Figure 3. Reference model for messaging interoperability gateway

The communicating agents or agent platforms can request the messaging gateway service in two ways. Firstly, an agent that recognizes that it cannot directly communicate with its peer (which can be either the destination agent platform or some third party agent platform in between) asks the gateway to perform the necessary translations. The agent can both send the message to the gateway and implicitly ask the gateway to forward the message, or the agent can ask the gateway to perform the necessary translations and return the translated message back. Obviously, the latter method cannot be used in the case of the MTP translation. Secondly, an agent that knows that it can handle only specific encoding can request the gateway to perform necessary translations for each incoming message. For example, an agent that is situated in a mobile device, can request some gateway at the fixed network to translate all the incoming messages to a format which is suitable for the wireless link.

#### 4. CONCLUSIONS

Flexible messaging is a desired feature in any communication architecture. We presented briefly the FIPA communication architecture and introduced the concept of a messaging interoperability gateway. FIPA has standardized flexible communication architecture, but in many cases direct end-to-end interoperability is impossible, impractical, or undesirable. Therefore, messaging interoperability gateways are needed.

A messaging interoperability gateway is able to translate between FIPA message transport protocols as well as between different concrete encoding of various FIPA message parts. At the time of writing, the standardization work in this area has just begun. The experimental standard is expected to be released in the beginning of the year 2002.

The security is an important issue that has to be addresses in the context of messaging interoperability gateways. This, however, is something that we have not yet considered thoroughly.

#### Acknowledgments

The authors express their thanks to the rest of the members of the FIPA Gateways Technical Committee, especially to Michael Berger, Kari Koivuniemi, Heimo Laamanen, Mikko Laukkanen, Jamie Lawrence, Milla Mäkeläinen, Satoshi Nishiyama, John Shepherdson, and Santtu Toivonen.

#### References

- [1] Foundation for Intelligent Physical Agents. *FIPA ACL Message Representation in Bit-Efficient Specification*. Geneva, Switzerland, October 2000. Specification number XC00069.

- [2] Foundation for Intelligent Physical Agents. *FIPA ACL Message Representation in String Specification*. Geneva, Switzerland, November 2000. Specification number XC00070.
- [3] Foundation for Intelligent Physical Agents. *FIPA ACL Message Representation in XML Specification*. Geneva, Switzerland, October 2000. Specification number XC00071.
- [4] Foundation for Intelligent Physical Agents. *FIPA Agent Message Transport Envelope Representation in XML Specification*. Geneva, Switzerland, November 2000. Specification number XC00085.
- [5] Foundation for Intelligent Physical Agents. *FIPA Agent Message Transport Envelope Representation in Bit Efficient Specification*. Geneva, Switzerland, November 2000. Specification number PC00088.
- [6] Foundation for Intelligent Physical Agents. *FIPA Agent Message Transport Protocol for HTTP Specification*. Geneva, Switzerland, October 2000. Specification number XC00084.
- [7] Foundation for Intelligent Physical Agents. *FIPA Agent Message Transport Protocol for HOP Specification*. Geneva, Switzerland, November 2000. Specification number XC00075.
- [8] Foundation for Intelligent Physical Agents. *FIPA Agent Message Transport Protocol for WAP Specification*. Geneva, Switzerland, October 2000. Specification number XC00076.
- [9] Foundation for Intelligent Physical Agents. *FIPA Content Languages Specification*. Geneva, Switzerland, October 2000. Specification number XC00007.
- [10] S. Hadjiefthymiades and L. Merakos. A survey of web architectures for wireless communication environments. *Journal of Universal Computer Science*, 5(7):390–417, 1999.
- [11] J. Jing, A. S. Helal, and A. Elmagarmid. Client-server computing in mobile environments. *ACM Computing Surveys*, 31(2): 117–157, 1999.
- [12] Object Management Group. *The Common Object Request Broker: Architecture and Specification*, 1999. formal/99-10-07. Version 2.3.1.
- [13] WAP Forum. *Wireless Application Environment Overview*, Version 1.2, November 1999.