TU Clausthal

Clausthal University of Technology

Communication platform for open heterogeneous MASs

Peter Novák

IfI Technical Report Series **IfI-08-13**

Department of Informatics
Clausthal University of Technology

Impressum

Publisher: Institut für Informatik, Technische Universität Clausthal
Julius-Albert Str. 4, 38678 Clausthal-Zellerfeld, Germany
Editor of the series: Jürgen Dix
Technical editor: Wojciech Jamroga
Contact: wjamroga@in.tu-clausthal.de
URL: http://www.in.tu-clausthal.de/forschung/technical-reports/
ISSN: 1860-8477

The IfI Review Board

Communication platform for open heterogeneous MASs

Peter Novák

Department of Informatics
Clausthal University of Technology
Julius-Albert-Str. 4, D-38678 Clausthal-Zellerfeld, Germany
peter.novak@tu-clausthal.de

Abstract

In this position paper we argue for a need of a lightweight communication platform aimed at supporting development of open heterogeneous multi-agent systems, such as multi-robot systems, or applications of ambient intelligence. We also argue, why the state-of-the-art FIPA compliant agent platforms are not directly suitable for this task and suggest a set of features of a suitable communication platform.

1 Introduction

Visions like ubiquitous computing, or ambient intelligence require large open multi-agent systems (MAS) comprising of heterogeneous cooperative agents. In such scenarios a number of agents embedded in consumer electronic devices, small robots, desktop computers, etc., interact to seamlessly support human everyday activities, or even fulfill complex tasks without human intervention. One of the main aspects of such systems is inter-agent coordination, i.e. communication. To enable communication in such MASs, there is a need for a middleware platform imposing as few requirements on the architecture and implementation of the MAS as possible.

To date, research interest in the area of multi-agent systems mostly focused on development of agent platforms adhering to standards, such as FIPA [5] or OMG MASIF [6]. Even though developments of these standards over time resulted in specification of abstract architectures taking into account a large range of MAS types (e.g. FIPA Abstract Architecture [3]), their main focus is mainly on enabling MASs, which are homogeneous w.r.t. the implementation programming language used.

We argue, that *the state-of-the-art MAS agent platforms and their associated toolkits are rather unsuitable for development of heterogeneous open multi-agent*

systems. We also identify a set of features a suitable communication platform for supporting such MASs should include.

2 State of the art

The state-of-the-art MAS platforms[1] either follow a widely accepted standards for interoperability of agents (FIPA or OMG MASIF), or use a proprietary, rather non interoperable approach. *JADE*[2] platform [1] is an epitome of a FIPA compliant agent platform, the currently prevailing MAS interoperability standard. While the group of the proprietary platforms is rather large and their main focal points vary, the noteworthy ones with a broad range of application domains include *OAA*[3] [2], *Cougaar*[4] [7] and *RETSINA*[5] [10].

While providing a high degree of interoperability w.r.t. inter-agent communication as well as platform distribution, the FIPA complying platforms are not suitable for development of open heterogeneous multi-agent systems. A fully FIPA compliant platform implementing the mandatory AMS service requires agents to physically reside on the platform and it overtakes management of the full agent lifecycle of agents running in it ([4], par. 4.2.2). Therefore, regardless of their specific features, all the state-of-the-art FIPA compliant agent platforms commit to a single agent programming language, which is almost exclusively *Java*. The standard does not specify any external API for connection of agents not physically residing on the platform.

The benefits of using FIPA-compliant platforms is however the strong support for inter-agent communication. Agents in these platforms communicate using FIPA Agent Communication Language (FIPA ACL), which became a de-facto standard for agent communication languages. Thus, use of FIPA ACL enables access to a plethora of specialized domain specific agent services and gateways implemented for the FIPA interoperable platforms, such as *JADE*.

On the other hand, except for *Cougaar*, most of the proprietary platforms like e.g. *OAA*, or *RETSINA* (specifically *RETSINA* Communicator [9]) are not so tightly bound to a specific agent programming language. The downside is however that they do not provide a sufficient interoperability support similar to that of the platforms following a standard. So even though it is relatively easy for agents running outside the platform, such as an on-board robot controller, to connect and register with it, they can only communicate with agents and services residing on the same platform. Often such plat-

[1] Although a plethora of MAS platforms was developed, most of them are not maintained any more. We focus only on freely available, still maintained platforms.

[2] `http://jade.tilab.com/`

[3] `http://www.ai.sri.com/oaa/`

[4] `http://cougaar.org/`

[5] `http://www.cs.cmu.edu/~softagents/retsina.html`

forms natively support KQML/KIF agent communication languages. Therefore multi-agent systems developed with such platforms are rather closed w.r.t. the outside world.

3 Communication platform

Availability of energy-efficient and affordable small form factor computers opens possibilities for application of the multi-agent metaphor to application domains like ubiquitous computing, ambient intelligence or multi-robot systems. To support development of a wide range of embodied open heterogeneous MASs, such as networks of agents controlling household appliances and devices for smart homes, or teams of robots, a specialized middleware is necessary. In particular, it is important on one side to *ensure a wide interoperability of multi-agent systems* using such a middleware solution, while at the same time *provide decoupling of agents from the platform, as well as from a specific agent programming language*. In the following, we propose and argue for a set of features such an agent communication platform should support.

agents should be platform independent entities taking care of their own execution and lifecycle;

interoperability issues the platform has to address and APIs it should provide are twofold:

- **inter-agent:** agents should be able to communicate in a standard communication language, such as FIPA ACL supporting a variety of message transport protocols applicable according to a particular application domain;
- **inter-platform:** agents associated with the platform should not only be able to communicate with agents registered with the same platform, but with agents and services on other standard (FIPA) compliant platforms as well;

platform services of the platform should stem from a widely accepted standard (FIPA), so that the platform provides the necessary services to the agents, but at the same time does not constrain their independence. The essential services should therefore include

- **directory service** with which agents can register their coordinates, properties and capabilities and which should provide a look-up facility;
- **message transport service** enabling the inter-agent communication on the same, or on other standard complying different platforms,

or gateways. It should also provide a translation between various message transport protocols;

technical implementation has to support agent vs. platform decoupling, i.e. should result in a *lightweight and modular middleware solution* supporting

portability and scalability: the middleware should enable deployment on various hardware and software platforms, with a focus on a wide range of computers from small form factor computers, to server solutions.

lightweight APIs: the interfaces the platform provides should be *agent programming language agnostic*, i.e. the platform designers should impose as few restrictions on associated agent implementation technology as possible (e.g. provide a plain TCP socket interface);

robustness: in a heterogeneous open multi-agent systems a platform cannot rely on correctness of behaviours agents associated with it perform, hence it has to be able to deal even with malicious behaviours in a robust manner.

4 Related work & Conclusion

The proposed communication infrastructure for *open heterogeneous multi-agent systems* stems from the FIPA Abstract Architecture [3], however not imposing strong constraints on agents it manages. The most prominent issues it relaxes are the strong coupling with a specific agent programming language and the management of agents execution and lifecycle by the platform. In the case of FIPA, the nowadays prevailing standard, these constraints are a result of the FIPA Abstract Architecture reification, in the form of the Agent Management Specification [4], in particular Agent Management Services (Subsection 4.2 therein) and Agent Platform (Section 5 therein) specifications.

Except for the use of a standard ACL, perhaps the closest relative of the proposed agent communication platform is the *CoABS Grid*[6] [8] infrastructure, which however is not in freely available under open source licensing terms.

References

[1] Fabio Bellifemine, Federico Bergenti, Giovanni Caire, and Agostino Poggi. *JADE - A Java agent development framework*, volume 15 of *Multi-*

[6] `http://coabs.globalinfotek.com/`

agent Systems, Artificial Societies, and Simulated Organizations, chapter 5, pages 125–147. Kluwer Academic Publishers, 2005.

[2] Adam Cheyer and David L. Martin. The Open Agent Architecture. *Autonomous Agents and Multi-Agent Systems*, 4(1/2):143–148, 2001.

[3] Foundation for Intelligent Physical Agents. FIPA Abstract Architecture Specification. #SC00001.

[4] Foundation for Intelligent Physical Agents. FIPA Agent Management Specification. #SC00023.

[5] Foundation for Intelligent Physical Agents. FIPA Specifications. available from `http://www.fipa.org/`.

[6] Object Management Group. Mobile Agent Facility Specification, version 1.0, January 2000. available from `http://www.omg.org/technology/documents/formal/mobile_agent_facility.htm`.

[7] Aaron Helsinger, Michael Thome, and Todd Wright. Cougaar: a scalable, distributed multi-agent architecture. In *SMC (2)*, pages 1910–1917. IEEE, 2004.

[8] Martha L. Kahn and Cynthia Della Torre Cicalese. The CoABS Grid. In Walt Truszkowski, Christopher Rouff, and Michael G. Hinchey, editors, *WRAC*, volume 2564 of *Lecture Notes in Computer Science*, pages 125–134. Springer, 2002.

[9] Onn Shehory and Katia P. Sycara. The retsina communicator. In *Agents*, pages 199–200, 2000.

[10] Katia P. Sycara, Massimo Paolucci, Martin Van Velsen, and Joseph A. Giampapa. The RETSINA MAS infrastructure. *Autonomous Agents and Multi-Agent Systems*, 7(1-2):29–48, 2003.