Chapter 1
A gateway protocol based on FIPA-ACL for the new agent platform PANGEA

Alejandro Sánchez, Gabriel Villarrubia, Carolina Zato, Sara Rodríguez, and Pablo Chamoso

Abstract Communication is one of the cornerstones of the intelligent agents paradigm. There are different forms of communication between agents, just as there are many platforms for creating them. However, one of the problems we encountered when using the agent paradigm is the actual communication between platforms. That is, to have a gateway of communication between different types of agents regardless of the platform has been used to create them. To this end, a new way of communication between PANGEA and other multiagent platforms is presented in this paper. In this communication process the FIPA-ACL standards are used.

Key words: PANGEA; Multi-agent systems; Gateway; Communication agents;; ACL messages; FIPA-ACL

1.1 Introduction

Currently, multi-agent systems (MAS) have become an efficient mechanism for the development of tools and distributed applications with strong needs of scalability, interaction and of course, gifted with intelligence at different levels. The main problem is that each tool is built on a different agent platform and in most cases it is difficult to communicate with each other platforms despite the FIPA [7] standards. This drawback forces to build middlewares or frameworks to enable this interaction between systems running on different platforms. This paper presents the development of a gateway to exter-

Department of Science Computing and Automation,
University of Salamanca, Plaza de la Merced, S/N,
37008, Salamanca, Spain.
e-mail: asanchezyu@usal.es;gvg@usal.es;carol_zato@usal.es;srg@usal.es;chamoso@usal.es

nal communication with the platform called PANGEA [28] using FIPA and IRC standards. This gateway allows external agents to get some result from PANGEA. Next section introduces the problem and explains why there is a need of agents communication between different platforms. The following section describes the main features and capabilities of the system. Finally we will present the results and conclusions, including the future work.

1.2 Background

When discussing MAS, the idea of a single agent is expanded to include an infrastructure for interaction and communication. Ideally, MAS include the following characteristics [13]: (i) they are typically open with a non-centralized design; (ii) they contain agents that are autonomous, heterogeneous and distributed each with its own personality (cooperative, selfish, honest, etc.). They provide an infrastructure specifically for communication and interaction protocols. Multi-agent systems allow the participation of agents within different architectures and even different languages [27] [5]. The development of open MAS is still a recent field of the multi-agent systems paradigm and its development will allow applying the agent technology in new and more complex application domains. Open MAS should allow the participation of heterogeneous agents with different architectures and even different languages [27]. However, currently, most of the multi-agent platforms (JADE [2],JACK [11], Jadex [13], Jason [3], S-MOISE+ [12], J-MOISE [14], Janus [9], MadKit [10], Cartago [24], NetLogo [26], SeSAm [19], Magique [25], MALEVA [4], Malaca [1]), are not able to communicate with agents of other platforms different than their own. Nowadays, the agent communication standard is characterized by ACL standard [6] (Agent Communication Language) of the Foundation for Intelligent Physical Agents (FIPA). This language marks a type of message, which contains the necessary fields for a good communication between agents [8]. The message structure has the following fields:

- Participants in communication: sender, receiver, reply-to.
- Performative.
- Content.
- Description of content: Language, encoding, ontology.
- Control of conversation: Protocol, conversation-id, reply-with, in-reply-to, reply-by.

All these fields have been defined by FIPA for a complete agent communication and any parameters will not be missed for the coherence of the message. Almost all existing multi-agent systems are using the standard FIPA-ACL for communication between agents, such as JADE [2],Jadex [23], JACK [11], Janus [9], MadKit [10], and NetLogo [26]. There are previous studies that try to create middleware for heterogeneous communication such as [21]. Other

studies focus on a gateway platform with web services [22](not a middleware). In our case, the implementation is to join several platforms with PANGEA [28]. PANGEA allows us to create virtual organizations of agents, and therefore, agents, regardless of platform and language. Since FIPA is the standard, the work presented in this paper, it also allows interaction between players who use this means of communication used in many other agents platforms.

1.3 Pangea Overview

PANGEA is a service oriented platform that allows the implemented open MAS to take maximum advantage of the distribution of resources. To this end, all services are implemented as Web Services [29]. Due to its service orientation, different tools modeled with agents that consume Web services can be integrated and operated from the platform, regardless of their physical location or implementation. This makes it possible for the platform to include both a service provider agent and a consumer agent, thus emulating client-server architecture. The provider agent (a general agent that provides a service) knows how to contact the web service, while the remaining agents know how to contact with the provider agent due to their communication with the ServiceAgent, which contains information about services. Once the client agents request has been received, the provider agent extracts the required parameters and establishes contact. Once received, the results are sent to the client agent. Using Web Services also allows the platform to introduce the SOA (Service-oriented Architecture) into MAS systems. SOA is an architectural style for building applications that use services available in a network such as the web. It promotes loose coupling between software components so that they can be reused. Applications in SOA are built based on services.

Using PANGEA, the platform will automatically launch the following agents:

- OrganizationManager: this agent is responsible for the actual management of organizations and suborganizations. It is responsible for verifying the entry and exit of agents, and for assigning roles. To carry out these tasks, it works with the OrganizationAgent, which is a specialized version of this agent.
- InformationAgent: this agent is responsible for accessing the database containing all pertinent system information.
- ServiceAgent: this agent is responsible for recording and controlling the operation of services offered by the agents.
- NormAgent: this agent ensures compliance with all the refined norms in the organization.
- CommunicationAgent: this agent is responsible for controlling communication among agents, and for recording the interaction between agents and or-ganizations.

- Sniffer: manages the message history and filters information by controlling communication initiated by queries.

The platform agents are implemented with Java, while the rest of the agents may be implemented in other programming languages. Communication between agents has been based on the IRC protocol. The IRC protocol was used to implement communication. Internet Relay Chat (IRC) is a real time internet protocol for sim-ultaneous text messaging or conferencing. This protocol is regulated by 5 standards: RFC1459 [20], RFC2810 [15], RFC2811 [16], RFC2812 [17] and RFC2813 [18]. It is designed primarily for group conversations in discussion forums and channel calls, but also allows private messaging for one on one communications, and data transfers, including file exchanges [20]. The protocol in the OSI model I located on the application layer and uses TCP or alternatively TLS [16]. An IRC server can connect with other IRC servers to expand the user network. Users access the IRC networks by connecting a client to a server. There have been many imple-mentations of clients, including mIRC or XChat. The original protocol is based on flat text (although it was subsequently expanded), and uses TCP port 6667 as its primary port, or other nearby ports (for example TCP ports 6660-6669, 7000) [17]. The standard structure for an IRC server network is a tree configuration. The messages are routed only through those nodes that are strictly necessary; however, the network status is sent to all servers. When a message must be sent to multiple recipients, it is sent similarly to a multidiffusion; that is, each message is sent to a network link only once [15]. This is a strong point in its favor compared to the no-multicast protocols such as SimpleMail Transfer Protocol (SMTP) or the Extensible Messaging and Presence Protocol (XMPP). One of the most important features that characterizes the platform is the use of the IRC protocol for communication among agents. This allows for the use of a protocol that is easy to implement, flexible and robust. The open standard protocol enables its continuous evolution. There are also IRC clients for all operating systems, including mobile devices

1.3.1 Gateway Pangea-ACL Messages

The gateway arises from the need to communicate external agents with PANGEA multiagent system. The external agent must send an FIPA-ACL object to the IP ad-dress of the PANGEA MAS, through the 6668 port. The object must have all the necessary fields for a good communication. This is the case of the ontology, content, sender, etc. The responsible for making inside-outside communication is the ACLAgent, which is deployed in PANGEA. This special agent is the responsible of converting FIPA-ACL messages in PANGEA messages. In addition to this, it also makes the demanded operations and it must return the results by an ACL object. At first, it takes cares of the operations based on the request, inform, subscription, and contract-net

protocols. The subsections below shows these operations based on the protocols. We use JAVA introspection to differentiate the protocols mentioned before. The ACLAgent manages the input object via Java introspection as follows:

Request protocol:

This kind of operation is used in order to get a result of a known web service which it is offered in PANGEA platform. In Figure 1.1 it is possible to observe the workflow that is followed in this protocol. First of all, the external agent sends the object to the ACLAgent, The ACLAgent sends a message to OrganizationManager to get a service result. The OrganizationManager talks with the InformationAgent if there is a service with a specific name. If there is not a service with the same description, the OrganizationManager ask to execute the service and get the result. Finally ACLAgent sends the response to the external agent.

if *object.protocol equals REQUEST* **then**
  send request to ServiceAgent;
  receive Response;
  send Response to external agent;
end

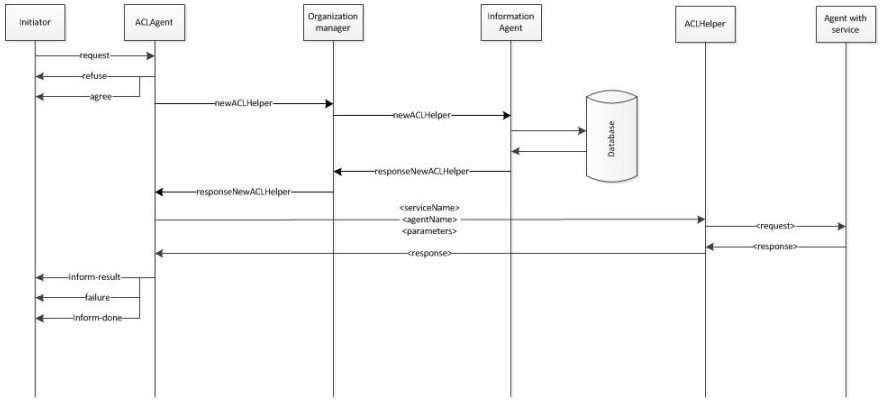

Fig. 1.1 Request workflow

Inform protocol:

The inform protocol is used to inform PANGEA that a new service is offered. So, the external agent sends a message with the content right format and it will receive an answer (refuse, failure, inform-done). In Figure 1.2, the inform workflow is shown. The external agent wants to register a new service to Pangea MAS. It sends an object to the ACLAgent with the service description. The ACLAgent sends it to the OrganizationManager and it registers the service. Finally, ACLAgent sends a response to the external agent.

if *object.protocol equals INFORM* **then**
  send new Service to OrganizationManager;
  send Response to external agent;
end

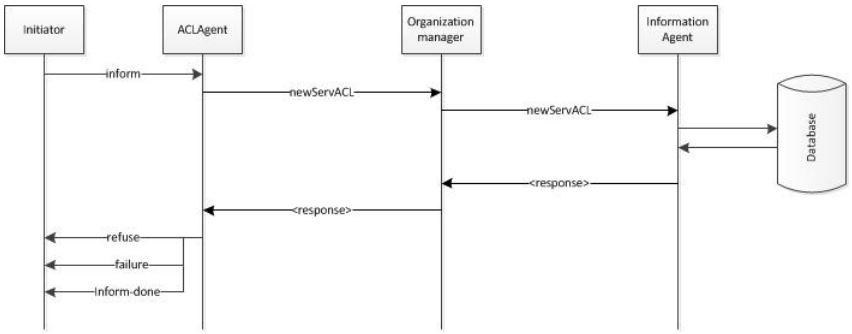

Fig. 1.2 Inform workflow

Subscription protocol:

External agent makes the request of a subscription. The ACLAgent will return it every news or modifications made in that subscription. In Figure 1.3 we can see the subscribe workflow.

if *object.protocol equals SUBSCRIPTION* **then**
  create subscriptionHelperAgent;
  if *message received from subscription is TRUE* **then**
    send Message to external agent;
  end
end

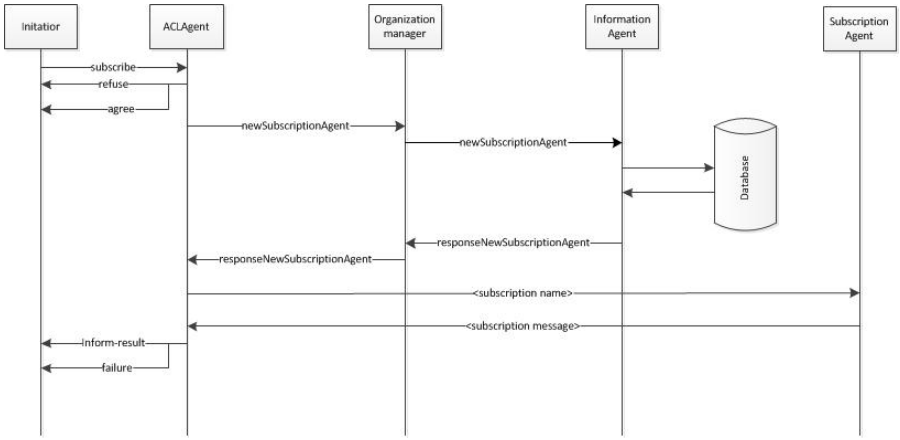

Fig. 1.3 Subscribe workflow

Contract-net protocol:

In this protocol, the external agent can search, find and execute a specific service. For this purpose, sending multiple messages is needed. Firstly, the name of the service which will be executed, must be known, so, a CFP message (Call for proposals) is sent with a little description of the service needed in the content field. Several answers will be sent to the external agent. The agent will choose one of them, and the chosen one will be executed. In Figure 1.4, we can see the contract net protocol workflow.

if *object.protocol equals CONTRACT-NET* **then**
  send request to ServiceAgent;
  receive service names;
  send response to external agent;
  if *response from outside* **then**
    send request to ServiceAgent;
    receive Response;
    send Response to external agent;
  end
end

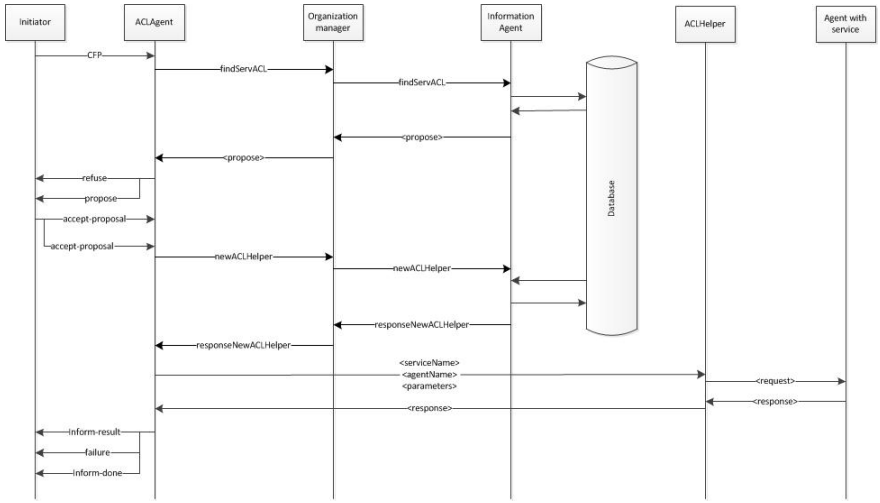

Fig. 1.4 Contract-net workflow

In JADE, sending a message is instant, the programmer gives the order, and the message is sent ipso facto. The ACLAgent does not it in this way. It has a messaging queue, so that when a programmer gives the order to send a message, the message is automatically inserted in the scheduled queue and manages the input object via Java introspection. In Figure 1.5, a comparative picture between JADE and the changes made in the gateway with PANGEA is shown.

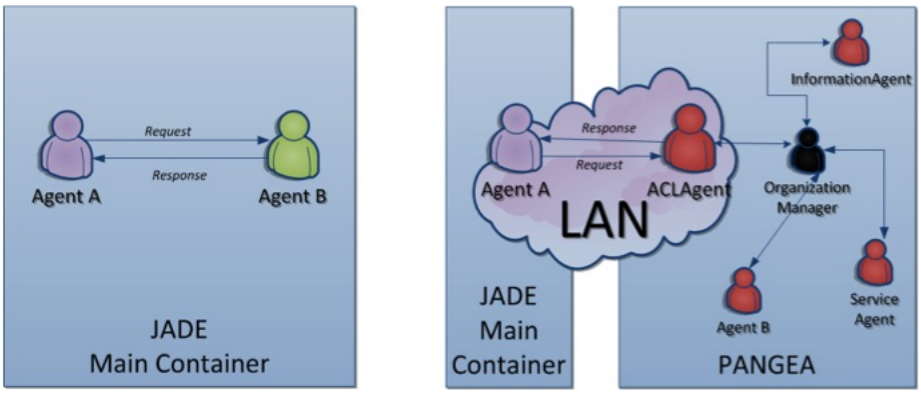

Fig. 1.5 (a) Sending a message in JADE. (b) Sending a message incorporating PANGEA

1.4 Case Study

The case of study consists on the communication between the PANGEA platform and some agents developed in the JADE platform. Three computers are involved in this test; they are connected to the same LAN. In one of these computers, PANGEA has been deployed with all the control agents, including the ACLAgent, which is the responsible for the external communication. In a second machine, we have other PANGEA agents which offer services to PANGEA multi-agent system. One of these agents offers solutions for basic arithmetical operations (addition, multiplication, division, and substraction). The other agent offers news from a RSS channel. These are the agents that are asked for the services. The external agents are deployed in the last computer. These agents are developed in JADE platform. The only thing we have to do in these agents, for being able to send messages to PANGEA platform, is to send a jade.lang.acl.ACLMessage object by 6668 port. Tests have consisted in the execution of the four different options we can make. We can divide the test in two groups:

- Simple workflow tests: These the tests of the protocols INFORM and REQUEST are. In these tests, the workflow consists in question-answer. There are one only input, and one only output. Both tests have been satis-factory, because, correct outputs have been obtained.
- Complex workflow tests: These the tests of the protocols SUBSCRIBE and CONTRACT-NET are. In the first case, it is composed by one input and several outputs (one per event done in the subscription). In the other hand, the other protocol consists on a communication between both parts, in which there are several input and output messages. Both parts must have to be listening for possible answers of the other agent. The obtained results of these tests have been satisfying, especially the part of the gateway, which has work perfectly. On the other hand, if the external agent does not control the flow of the messages, some of these will be lost and the communication will be incomplete.

1.5 Results and Conclusions

With the development of this gateway, the communication between different architectures has been achieved. The gateway allows several multi-agent systems work jointly for a common target, and share all its knowledge. After the made tests during its development, obtained data has been significantly positive, because communication between both parts is very fluid and fast, so you hardly notice they are working on different machines. This is in that way because of the low weight of the packet sent through the LAN and the PANGEAs message management. Thanks to the use of the gateway, new

communication ways are opened. External agents will benefit from the services of the PANGEA platform, and vice versa. PANGEA agents will ask for some tasks to the outside of the platform. The implementation of the rest of the FIPA-ACL communication protocols have been proposed as future work. The rests of the protocols will make a complete com-munication, and all kind of queries would be made to PANGEA platform. It is also though to port and develop this idea in mobile environments.

Acknowledgements

This research has been supported by the project OVAMAH (TIN2009-13839-C03-03) funded by the Spanish Ministry of Science and Innovation.

References

1. Amor, M. and Fuentes L. (2009). Malaca: A Component and Aspect-Oriented Agent Architecture. Information and Software Technology pp. 10521065
2. Bellifemine,F. , Poggi,A. and Rimassa G. JADE A FIPA-compliant agent framework. Proceedings of the Practical Applications of Intelligent Agents 1999.
3. Bordini, R.H.,Hbner J.F., and Wooldridge M. Programming multi-agent systems in AgentSpeak using Jason. 2008
4. Briot, J-P, Meurisse T., and Peschanski F. Architectural Design of Compo-nent-Based Agents: A Behavior-Based Approach Programming Multi-Agent Sys-tems. pp. 7190. 2007
5. Corchado, E. Pellicer, M. A., Borrajo, M. L. A MLHL Based Method to an Agent-Based Architecture.International Journal of Computer Mathematics. Taylor and Francis, Volume 86, Issue 10 and 11, pgs. 1760-1768. 2008 ISI JCR Impact Factor: 0.423
6. FIPA ACL Message Structure Specification. http://www.fipa.org/specs/fipa00061/index.html
7. FIPA, The Foundation for Intelligent Physical Agents. http://www.fipa.org/
8. Foundation for Intelligent Physical Agents - FIPA ACL Message Structure Specification http://www.fipa.org/specs/fipa00061/SC00061G.pdf 2002
9. Gaud, N., Galland, Hilaire V., and Koukam A.. An Organisational Platform for Holonic and Multiagent Systems .Programming Multi-Agent Systems. 2009
10. Gutknecht, O. and Ferber J. The MadKit Agent Platform Architecture Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Sys-tems.2001
11. Howden,N., Rnnquist R., Hodgson A., and Lucas A. JACK Intelligent Agents - Summary of an Agent Infrastructure.5th International conference on auton-omous agents . 2001
12. Hbner, J.F., Sichman J., and Boissier O. Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent lev-els. International Journal of Agent-Oriented Software Engineering pp. 370395. 2007
13. Huhns, M., Stephens, L.: Multiagent Systems and Societies of Agents. In:Weiss, G.(Ed.), Multi-agent Systems: a Modern Approach to Distributed Artificial Intelligence, MIT, 1999.
14. Hbner, J.F., Sichman J., and Boissier O. Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent lev-els International Journal of Agent-Oriented Software Engineering 1.3, pp. 370395. 2007

15. Kalt, C., "Internet Relay Chat: Architecture", RFC 2811, April 2000.
16. Kalt, C., "Internet Relay Chat: Channel Management", RFC 2811, April 2000.
17. Kalt, C., "Internet Relay Chat: Client Protocol", RFC 2812, April 2000.
18. Kalt, C., "Internet Relay Chat: Server Protocol", RFC 2813, April 2000.
19. Klgl, F., Herrler R., and Oechslein C. From Simulated to Real Environ-ments: How to Use SeSAm for Software Development Multiagent System Technol-ogies. pp. 10971098. 2003.
20. Oikarinen, J. and Reed D. , "Internet Relay Chat Protocol", RFC 1459, May 1993.
21. Omair Shafiq,M . , Ali, A. ; Farooq Ahmad, H. ; Suguri, H. AgentWeb Gateway - a middleware for dynamic integration of multi agent system and Web ser-vices framework - Enabling Technologies: Infrastructure for Collaborative Enter-prise, 2005. 14th IEEE International Workshops on pp 267 268 13-15 June 2005
22. Omair Shafiq,M, Ding Y.; Fensel D. - Bridging Multi Agent Systems and Web Services: towards interoperability between Software Agents and Semantic Web Services Enterprise Distributed Object Computing Conference, 2006. EDOC '06. 10th IEEE International pp 85 96 Oct. 2006
23. Pokahr,A. , Braubach L., and Lamersdorf W. Jadex: A BDI Reasoning Engine. Multi-Agent Programming: Languages, Platforms and Applications. pp. 149174. 2005
24. Ricci, A., Viroli M., and Omicini A. CArtAgO: A Framework for Prototyp-ing Artifact-Based Environments in MAS. Environments for Multi-Agent Systems III. pp. 6786. 2007
25. Routier, J.C., Mathieu P., and Secq Y. Dynamic Skills Learning: A Support to Agent Evolution. Proceedings of the Artificial Intelligence and the Simulation of Behaviour, Symposium on Adaptive Agents and Multi-agent systems (AISB01). 2001
26. Wilensky, U. NetLogo itself. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL. 1999
27. Zambonelli F., Jennings, N.R., and Wooldridge M. . Developing Multiagent Systems: The Gaia Methodology. ACM Transactions on Software Engineering and Methodology, 12:317370, 2003.
28. Zato, C., Villarubia, G., Snchez, A., Bajo, J., Rodrguez, S. And Corchado, J.M. PANGEA Platform for Automatic coNstruction of orGanizations of intElligent Agents. Distributed Computing and Artificial Intelligence. Advances in Intelligent and Soft Computing Volume 151, 2012, pp 229-239
29. Zato, C.; Sanchez, A.; Villarrubia, G.; Rodriguez, S.; Corchado, J.M.; Bajo, J.; "Platform for building large-scale agent-based systems" , 2012 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), pp.69-73, 17-18 May 2012