An approach to conforming a MAS into a FIPA-compliant system

Christos Georgousopoulos
Department of Computer Science
Cardiff University, P.O.Box 916
Cardiff CF24 3XF, UK
Tel: +44 (0) 7799417703
geolos@cs.cf.ac.uk

Omer F. Rana
Department of Computer Science
Cardiff University, P.O.Box 916
Cardiff CF24 3XF, UK
Tel: +44 (0) 29 2087 5542
o.f.rana@cs.cf.ac.uk

ABSTRACT
The conversion of a MAS into a FIPA-compliant system (i.e. one that adheres to FIPA standards), is important to support interoperability across different MAS. Supporting such a conversion will also allow system developers to make more effective use of their existing systems. Such a conversion imposes amendments on the system architecture to conform to the new (FIPA) standards, which require extensive code re-writes and testing procedures. We propose a different approach to achieving FIPA compliance by providing specialized gateways, which act as wrappers between the legacy (non-FIPA) systems, and a system developed to FIPA specifications. We demonstrate this approach based on a MAS utilising an active digital library composed of multi-spectral images of the Earth, as part of the Synthetic Aperture Radar Atlas (SARA).

Categories and Subject Descriptors
C.4 [Computer Systems Organization]: Performance of Systems; D.1 [Software]: Programming Techniques; I.2 [Computing Methodologies]: Artificial Intelligence

General Terms
Performance, Standardization

Keywords
Methodologies and tools, agent languages and environments, agent-based software engineering, standards for agent and MAS.

1. INTRODUCTION
The highly interactive nature of multi-agent systems points to the need for consensus on agent interfaces in order to support interoperability between different agent systems. The completion and adoption of such a standard is a prerequisite to the commercialization and successful exploitation of intelligent agent systems.

A system that conforms to FIPA specifications (standards) is a FIPA-compliant system, and can interoperate with any other heterogeneous system which is FIPA-compliant also.

In this paper we briefly discuss and compare the most important efforts that define interoperability between agents on different types of platforms, with emphasis on FIPA efforts. We propose an approach to conforming a MAS into a FIPA-compliant one, using FIPA-compliant gateways and we outline its advantages. This approach is also suitable for legacy agent systems that need to be interoperable based on FIPA standards. Finally, we demonstrate how interoperability can be applied to a MAS, with particular to the SARA (Synthetic Aperture Radar Atlas) architecture[21].

2. DIFFERENT APPROACHES OF STANDARDIZATION
Currently, there are three important agent standardization efforts that define interoperability between agents on different types of platforms[18][19]. KQML community, OMG’s MASIF and FIPA.

KQML[15] was one of the first initiatives to specify how to support social interaction characteristic of agents using a protocol based on speech acts. However, it is not a true de facto standard in the sense that there is no consensus on a single specification or set of specifications that has been ratified by common agreement within an organization or forum. As a result, variations of KQML exist such as KQML classic, KQML '93 and KQML-Lite, leading to different agent systems that speak different dialects and that are not able to interoperate fully.

MASIF[16] differs from both KQML and FIPA in that it regards the defining characteristic for an agent as its mobility from one location to another. MASIF does not support the standardization of communication between agents on different agent platforms. Furthermore, MASIF restricts the interoperability of agents to agents developed on CORBA platforms whereas the focus of FIPA is to directly support the interoperability of agents deployed on agent frameworks which can support heterogeneous transport. OMG is exploring how to support other characteristics of software agent than mobile agents. It issued a “Request For Information” (RFI) on agents in 1999[17], although the outcome of this is not clear. FIPA has supplied its specifications as input to this request. This is still work in progress at this time.

The standardization work of FIPA is in the direction to allow an easy interoperability between agent systems, because FIPA beyond the agent communication language specifies also the key agents necessary for the management of an agent system, the...
ontology necessary for the interaction between systems, and it defines also the transport level of the protocols (unlike KQML).

In contrast to MASIF, both KQML and FIPA emphasise agency and social interaction between multiple agents as the defining properties for software agents. They both define interaction in terms of an Agent Communication Language (ACL) whereas MASIF defines interaction in terms of Remote Procedure Calls (RPC) or Remote Method Invocation (RMI).

FIPA specifications are not arbitrarily set, they have a life-cycle and in order for a specification to become standard two years of experimental tests must pass. Since FIPA was established the membership of companies and organizations has been increasing. In 1996 FIPA consisted of 25 companies (5 of which were Universities), while in the early 2001 this increased to 63. The increase of the members incorporated into FIPA, the presence of companies such as IBM, NASA, Intel, Phillips etc., and the utilisation in large-scale projects (such as Facts[8], FIPA-NET[9], Cameleon[3], AgentCities[1]) based on FIPA’s specifications; are factors that are likely to contribute to making FIPA specifications a universally accepted standard.

3. OVERVIEW OF FIPA SPECIFICATIONS

Since January 2000, FIPA has adopted a new procedure for classifying, organizing and releasing specifications to ensure coherence, completeness and consistency of its work as well as its relevance to industrial and commercial interests[5]. It is important to note that FIPA specifications do not attempt to describe how developers should implement their agent-based systems, nor do they attempt to specify the internal architecture of agents. Instead, they provide the interfaces through which agents can communicate. Each specification is given a subject association that describes the general area in which it belongs in the FIPA specification structure, depicted in figure 1. FIPA specifications are divided into five categories: Applications, Abstract architecture, Agent Communication, Agent Management and Agent Message Transport, which are briefly described below. Each area of specifications has one or more specification documents assigned to it, which can be downloaded from FIPA’s web-site[12].

![Figure 1. Specification breakdown](image)

Abstract Architecture: The purpose of the FIPA Abstract Architecture is to foster interoperability and reusability. To achieve this, it is necessary to identify the elements of the architecture that must be codified. By describing the relationships between these elements, it becomes clearer how agent systems can be created so that they are interoperable.

Agent Message Transport: The FIPA Agent Message Transport Specification deal with the delivery and representation of messages across different network transport protocols, including wireline and wireless environments. At the message transport level, a message consists of a message envelop and a message body. The envelope contains specific transport requirements and information that is used by the Message Transport Service (MTS) on each agent platform to route and handle messages. The message body is the real payload and is usually expressed in FIPA ACL but is opaque to the MTS since it may be compressed or encoded. The agent message transport reference model depicted on figure 2.

![Figure 2. Agent Message Transport Reference Model](image)

![Figure 3. Agent Management Reference Model](image)

Agent Management: The FIPA Agent Management specification provides the framework within which FIPA agents exist and operate. It establishes the logical reference model for the creation, registration, location, communication, migration and retirement of agents. The entities contained in the agent management reference model depicted in figure 3, are logical capability sets i.e. services and do not imply any physical configuration. Additionally, the implementation details of agent platforms and agents are the design choices of the individual agent system developers. The reference model describes the primitives and ontologies necessary to support the following services in an agent platform:

- White pages, such as agent location, naming and control access services, which are provided by the Agent Management System (AMS). Agent names are represented
by a flexible and extensible structure called an agent identifier, which can support social names, transport addresses, name resolution services, amongst other things.  
- Yellow pages, such as service location and registration services, which are provided by the Directory Facilitator (DF).  
- Agent message transport services as described in Agent Message Transport.

**Agent Communication:** Communication between agents in FIPA is based on a model of semantically grounded communication i.e. communication that is pre-defined, semantically rich and well understood by agents. FIPA specifies a number of communicative acts (performatives), such as request, inform and refuse in a well-defined manner that is independent from the overall content of the message. The message that is supplied with a communicative act is itself wrapped in a well-specified envelope, called an agent communication language (ACL). ACL provides mechanisms for adding context to the message content, the sender and receiver, the ontology and interaction protocol of the message. The actual content of a message is expressed in a content language, such as XML. Finally, the set of FIPA interaction protocols describe entire conversations between agents for the purpose of achieving some interaction or effect, such as auctioning, issuing a call for proposal, negotiating brokering services and the registration and deregistration of subscriptions.

**Agent Applications:** FIPA has developed specifications of four agent-based applications that contain service and ontology descriptions and case scenarios:  
- Personal Travel Assistance  
- Audio-Visual Entertainment and Broadcasting  
- Network Management and Provisioning  
- Personal Assistant

Additionally, the Agent Software Integration specifications contains a guide for integrating legacy software i.e. software that does not communicate using FIPA ACL.

**4. AN APPROACH TO CONFORMING A MAS INTO A FIPA COMPLIANT ONE**

The conversion of a MAS into a FIPA-compliant system (i.e. a system that adheres to FIPA standards), implies that system developers must rebuild their systems based on FIPA specifications. Such a conversion imposes amendments on the system architecture to conform to the new standards, which results in extensive code implementation and testing procedures. Based on the guidelines provided by FIPA association, for an agent platform implementation to be considered FIPA-compliant, it must at least implement the “Agent Management” and “Agent Communication Language” specifications, which should conform to the latest experimental and/or standard status specifications.

The usual approach to conforming a MAS into a FIPA-compliant one, is to modify the whole system based on FIPA specifications. A different approach that has not yet been adopted by any developer is to amend just a part of the system’s architecture. The top picture of figure 4, represents a typical multi-agent system (MAS 1) that has been conformed to FIPA specifications in order to be able to interoperate i.e. receive/send data from/to other FIPA-compliant multi-agent systems (EXternal MAS). The second picture of figure 4, represents our approach of conforming a MAS to a FIPA-compliant one. The actual architecture of the system remains the same as before but two FIPA-compliant gateways (in grey) have to be added to the system, that work as *adaptors* to ensure interoperability with other FIPA-compliant multi-agent systems (EX MAS). Interoperability in this sense applies at both the communication and application levels, as specified by FIPA specifications. The communication level comprises the connection and communication layer, where as the application level comprises the ontological and agent service layer[4].

The two gateways are the FIPA-compliant part of the system. Each of those has all of the mandatory, normative components of the FIPA architecture. A representation of the FIPA-compliant gateway architecture is depicted in figures 6 and 7 of section 5, where we demonstrate the adoption of the FIPA-compliant gateways by the SARA multi-agent system. Each gateway contains three agents; the Agent Management System (AMS), the Directory Facilitator (DF) and the gateway agent. The AMS and DF are the FIPA agents, as defined by FIPA specifications. The gateway agent is the only agent of the system registered by both AMS and DF, which acts as a wrapper between MAS 2 and any external MAS. All the available services of the system are represented by this agent. It is like having an ordinary FIPA compliant system with only one registered agent capable of providing services. The Directory Facilitator (DF) and Agent Communication Channel (ACC), which manage the gateway agent and communication channel, support the required infrastructure for enabling service interoperability and are part of the FIPA specifications. The communication between an EX MAS and MAS2 is accomplished through the Agent Communication Channel (ACC) and the protocols that are supported (concerning the connection layer) are reflected thought the platform address. The gateway agent communicates with agents from EX MAS using the FIPA Agent Communication Language (ACL). Its responsibility is to translate the incoming messages to a form understood by its internal agents i.e. the agents that are hidden by the EX MAS. Likewise, the internal agents’ requests have to be also converted by the gateway agent into ACL messages, in order to be understood by an EX MAS. The gateway agent has a list of the agents of the system along with the registered services (with DF) that each of them can provide. Therefore, based on the service requested by an EX

---

**Figure 4. Two different approaches of conforming an agent platform into a FIPA-compliant one**
MAS, the gateway agent knows to which system agent the message should be forwarded, after it has been translated into the form understood by the appropriate agent that receives the request.

Hence, the external MAS does not see anything else apart from the gateway agent. The agent, after receiving a request from an external MAS (on the left side of MAS2) is responsible for transferring the request to the agents of its system, which are hidden by the external MAS, for processing the request. Once the request is accomplished, response is sent back to the external MAS through the gateway agent. In the case where agents from MAS 2 need to communicate with an external MAS (on the right side of MAS2), their request is passed through the gateway agent and translated into ACL; the results gathered by the external MAS are returned to MAS 2 agents through the gateway agent as well.

An example of an ACL message send to the gateway agent by an agent from an EX MAS, requesting a SARA image of specific coordinates may be:

(request
  :sender agent_from_EX MAS
  :receiver EXSA
  :content (coordinates 50,50,85,75,105,60,150,150)
  :language PlainText
  ...
)

The message that is generated after the translation of the gateway agent into a form understood by its internal agent (with reference to the SARA project described on section 5) may be:

<?xml version="1.0">
<!DOCTYPE message SYSTEM “message.dtd”>
<Message>
  <querydef>
    <trackquery>
      <Condition>
        <and>
          <GreaterThanOrEqual><left>latitude.upperleft</left> <right>50</right></GreaterThanOrEqual>
          <GreaterThanOrEqual><left>longitude.upperleft</left> <right>105</right></GreaterThanOrEqual>
          <LessThanOrEqual><left>latitude.lowerleft</left> <right>75</right></LessThanOrEqual>
          <LessThanOrEqual><left>longitude.lowerleft</left> <right>60</right></LessThanOrEqual>
          <GreaterThanOrEqual><left>latitude.upperright</left> <right>85</right></GreaterThanOrEqual>
          <LessThanOrEqual><left>longitude.upperright</left> <right>150</right></LessThanOrEqual>
          <LessThanOrEqual><left>latitude.lowerright</left> <right>150</right></LessThanOrEqual>
          <LessThanOrEqual><left>longitude.lowerright</left> <right>150</right></LessThanOrEqual>
        </and>
      </Condition>
    </trackquery>
  </querydef>
</Message>

In the SARA MAS the agents’ messages are encoded in XML. Therefore, the above XML message, which is a translation of the ACL message, can be understood by the SARA agents (in this case, MAS 2).

The interoperability of the FIPA-compliant gateway may be further extended by defining extra layers. For instance, the utilisation of a security layer will enable heterogeneous MAS to interoperate using different security certificates. In addition, an agent mobility layer, will allow the realization of agent migration between heterogeneous MAS build on the same agent platform; we are currently developing the mobility layer.

4.1 Advantages of Adopting the FIPA-Compliant Gateway

The alternative approach of using FIPA-compliant gateways for conforming a MAS into a FIPA-compliant one, yields the following advantages:

- System’s architecture remains the same as before. Implementation is only needed for the FIPA-compliant gateways and the interaction between the gateway agents with the other agents of the system. Unfortunately, there are no standard specifications set by FIPA yet and most of them are in experimental stage. Although, based on FIPA, developers should conform to the latest specifications to guarantee a 100% FIPA-compliant system. The continuous improvement of FIPA specifications have a direct affect on the developer’s systems since they should conform to the latest specifications. Consequently, developers can save time in terms of design and implementation by applying the new standards (the FIPA revised specifications) only in a specific part of their system i.e the FIPA-compliant gateways; avoiding the complexity of amending the whole system.

- Security is increased. Specifications pertaining to security within the context of the FIPA specifications were started at the beginning of 1997, the FIPA 97 agent management specification[6] and the FIPA 98 agent management security specification[7]. There is still no coherent agent security details from FIPA at this time. In fact both of these specifications have now been declared obsolete by FIPA: the management specification has been superseded by new specification but which contains no reference to security. Although, FIPA is planning in the future to investigate security related issues within FIPA architecture and formulate a long term strategy for the integration of security features into FIPA specifications[2][14]. There is currently debate as to whether a generic or default level of agent security ought to be specified. It is also required that such security criteria be applicable to different types of agent infrastructures and application domains[20] since:

  - security is a complex issue in the context of MAS and generally system level security.
  - security is part of the software infrastructure in which the agent platform is embedded and is outside the scope of an agent architecture.
  - security is domain and platform (implementation) specific, there is no general agent security architecture which is suitable for all applications and implementations.
  - the focus has been the development of collaborative, rational agent services within Intranets. Some agent systems do not need security.
Based on our approach, isolating the interoperable part of the architecture (i.e., the gateways) from the rest of the system; increases security. The policy of the architecture remains hidden to the foreign Agency due to the FIPA-compliant gateways. The interaction between the system and a foreign agency is managed by the gateway agent; the rest of the agents, hardware/software resources cannot be accessed. Securing the FIPA-compliant gateways, from where foreign malicious agents can enter into the system, implies minimum security for the rest of the system. Therefore, the FIPA-compliant gateways can also act as a shield for the core system. Requirements and design issues for adding security to FIPA agent systems can be found in [20].

- Performance is improved. FIPA specifications exist for the intercommunication between heterogeneous agent systems i.e. agents that are hosted on different platforms. Consider an agent system which does not need to communicate with a foreign one. The conversion of such a system into a FIPA-compliant one would be useless, since the agents which belong to the system can obviously interoperate between themselves. Since interoperability can be achieved with the use of the FIPA-compliant gateways without actually affecting the actual system, it is unnecessary to conform the whole system to FIPA specifications. For instance, the existence of the Directory Flicitator (DF), Agent Management System (AMS), Agent Communication Channel (ACC) and Internal Platform Message Transport (IPMT), which are mandatory, normative components of the FIPA architecture, impose extra complexity and delay in a system constituted by homogeneous.

It is likely that a legacy agent system will not utilise the FIPA Agent Communication Language (ACL), especially if the agents within such a system are identical. An important role of the gateway, in this context, is to translate messages from a FIPA compliant to legacy system. Most agent languages such as KQML and FIPA have been designed to minimize the size of the message and to function more as a data-passing protocol. Little emphasis has been placed on the flexibility or the transparency of the semantics of the message. Many other agent communication languages use the basic KQML/FIPA style, but replace or extend the sets for special purposes. The eXtensible Markup Language (XML) is becoming the standard for data interchange on the Internet, and enables a new generation of Web services that are not meant for humans to use directly, but rather to be used by other software e.g. agents. Its flexibility and ability to clearly express intentions in a predefined ontology can be found in [23].

In addition, the FIPA-compliant gateways have a direct affect on the security of the system and therefore on its performance. They more secure the FIPA-compliant gateways are, the less security is needed for the rest of the system. For instance, the cost of encrypting the messages transmitted between the agents, apart from the gateway agent, can be avoided. Consequently, the minimization of security (apart from the FIPA-compliant gateways) increases the overall performance of the system.

5. INTRODUCING INTEROPERABILITY IN SARA ARCHITECTURE USING FIPA-COMPLIANT GATEWAYS

Our research is based on the Synthetic Aperture Radar Atlas (SARA) active Digital Library. In order to achieve interoperability between our system and a foreign one, we have adopted the FIPA-compliant gateways approach. In the following section we give a brief discussion of SARA project and we demonstrate how interoperability can be achieved by using the approach outlined previously.

5.1 The SARA Active Digital Library

SARA is an active digital library of multi-spectral remote sensing images of the earth from the SIR-C Shuttle mission, which provides web-based online access to a library of data objects at Caltech, the San Diego Supercomputer Center, and the University of Leicester in Italy. The objective of the SARA project is to develop an infrastructure for a high-speed, high-volume, multi-protocol, distributed database, together with a means to attach distributed computing resources for data conversion, visualization and knowledge discovery.

A prototype MAS, which comprises both intelligent and mobile agents, has been developed to manage and analyse distributed multi-agency remote sensing data; more information can be found on our web-site. The SARA architecture is composed of a collection of information and web servers, each of them having a group of agents, Local Interface Agents (LIA) and User Interface Agents (UIA) accordingly.

We separate mobile agents from stationary service agents. Our approach is to localize the most complex functionality in non-mobile LIAs, which remain at one location, providing resources and facilities to lightweight mobile agents that require less processor time to be serialized and therefore quicker to transmit. LIAs are stationary agents that provide an extensible set of services. LIAs provide a level of abstraction between resource servers and requesting mobile agents, namely:

- LAA (Local Assistant Agent) supports interaction with any visiting URAs and assists the completion of the task carried by the URA. It also performs a resource-check on the user’s file-space. Each user has a fixed amount of physical storage on each server, where their files are being stored. The objective of LAA’s resource-check is to maintain the file-space of each user, and prevent a user from exceeding the fixed amount of physical storage space that he owns on a given information server. Finally it informs LMA for the availability of resources.
- LMA (Local Management Agent) coordinates access to other LIAs and supports negotiation among agents. It is responsible for optimizing itineraries to minimize the bottlenecks inherent in parallel processing and ensuring that
the URA is transferred successfully. It also informs UMA for the status of its local server.
- LIGA (Local InterGration Agent) provides a gateway to a local workstation cluster, or a parallel machine.
- LSA (Local Security Agent) is responsible for authenticating and performing a validation check on the incoming URAs. The URA will be allocated an access permission level. Agents from registered users may use have access to more information resources than the agents from unregistered users.
- UMA (Universal Management Agent) Its task is to optimize the overall system’s performance. Based on its interaction with each LMA, it is capable of optimizing mobile agent migration from the beginning; apply cash techniques and balance the distribution of agents between the information servers. This is due to its information concerning the system status i.e. the status of each server, the availability of resources, the distribution of agents on the network and their activities, any conflicts/failures or updates taking place on the system.
- URAS (URA’s Servant) is the FIPA-compliant gateway agent of each information server. Its task is to perform interoperability between SARA system and a FIPA-compliant one.
- EXSA (External Service Agent) is the FIPA-compliant agent of each web server. Its task is to perform interoperability between a FIPA-compliant system and SARA.

UIAs provide a front end to the end user, for checking the user
input and displaying the results, namely:
- UAA (User Assistant Agent) manages the information of the user and provides control functions for him. It launches URAs on behalf of the user, tracks their progress and location, and provides the dispatched URA with a contact point to which the results can be returned. It also enables the visualization of results according to the user’s choice.
- URA (User Request Agent) is responsible for holding a user’s request, carrying it to the appropriate local archive site(s) interacting with LIAs at each remote site visited, fusing the results into a single result that answers the user’s query and returning the results to the UAA.

5.2 Agent Collaboration Support Mechanism
In this section we describe in detail the interaction scheme of the SARA agents. Figure 5 illustrates a simple example of query processing, and shows the interaction and collaboration of the agents. The process of agent execution is as follows:

Step 1: The user visits the SARA web-server where he enters his information i.e. the desired query, username, password. Its information is gathered by UAA agent, which is in the form of a servlet.

Step 2: UAA launches URA by supplying it with the user’s information.

Step 3: URA communicates with UMA which is responsible of

![Figure 5. The FIPA interoperable SARA architecture](image-url)
constructing URA’s itinerary according to the information provided by the former and the current status of the system (known by UMA), i.e. availability of resources, server failures, number of agents on each server. UMA may also direct URA to collect the results of its query from a server which have already been stored by a previous agent having a similar query. Note that the management agent’s (UMA, LMA) interaction in not described in this paper.

Step 4: Once the URA’s itinerary is constructed, it communicates with the LSA of the first server of its itinerary.

Step 5: After URA is authenticated and accepted by the server that it needs to migrate to (through LSA), it migrates to it.

Step 6-7: Interacts with LAA and LRA which act as wrappers, wrap up the information source and make it thus accessible in a standard form. For instance, LAA connects to the server’s database using JDBC, then LRA executes the URA’s query and converts the results into XML. Finally, the results are send back to URA.

Step 8: URA reports its activities on the local server to LMA. If URA needs to migrate again and there is a change in the systems status that affect URA’s task, LMA is responsible of informing URA and amending its itinerary.

Step 9: As in step 4, before URA migrates to the next server of its itinerary, it needs first to communicate with the LSA of that server.

Step 10: Once LSA has granted access to URA, URA moves to the foreign server to continue its task.

When URA accomplish its task, it sends a URL reference with the results of analysis to LAA. LAA is then able of presenting the results to URA.

5.3 SARA and FIPA Compliance

The introduction of FIPA interoperability into the SARA system enables it to communicate with other MAS and vice-versa. The union of SARA system with other MAS extends its capabilities by providing users with further information. For instance, information retrieved from the SARA system can be further enhanced by additional information gathered from a GIS system that is capable of interoperating with SARA. The longitude and latitude of a particular area of the earth can be used as parameters on a GIS (Geographic Information System) to retrieve land information such as street names, which can then be combined with the image based on geographical coordinates in SARA, resulting in a detailed map of the particular area. Likewise, a foreign MAS can interoperate with SARA and use its information.

The interoperability of the SARA system is based on the use of FIPA-compliant gateways which are implemented using FIPA-OS toolkit. FIPA-OS is an open source implementation of the mandatory elements contained within the FIPA specification for agent interoperability. In addition to supporting the FIPA interoperability concepts, FIPA-OS also provides a component based architecture to enable the development of domain specific agents which can utilise the services of the FIPA Platform agents. The primary aim of FIPA-OS is to reduce the current barriers in the adoption of FIPA technology by supplementing the technical specification documents with managed open source code.

The architecture of the SARA system with added FIPA interoperability is depicted in figure 5. An external multi-agent system (EX MAS) can interoperate with SARA through the FIPA-compliant gateway (outlined by the dashed box) which is placed on every Web-server, where SARA can interoperate with an EX MAS through the FIPA-compliant gateway which is placed on every Information-server. The architecture of the FIPA-compliant gateways which is a slight variation of the architecture of FIPA-OS configuration case 3 is depicted in detail in figures 6 and 7.

The EXSA (EXternal Service Agent) agent is the gateway agent of the FIPA-compliant gateway placed on every Web-server. This agent is responsible for receiving a request from an external MAS i.e. a foreign agent, and passing it to the URA. EXSA can be considered similar to UAA; based on the fact that, as a client is
represented by a UAA, an external MAS is represented by an EXSA. When URA receives the appropriate information from EXSA it processes its request (i.e. by starting its itinerary) as it would be instructed by a UAA agent. When URA finishes its job, it sends the results back to the EXSA which then passes this to the foreign agent from where the request has been initially placed. The resource access level and request’s priority level is according to the EX MAS that accesses SARA.

The URAS (URA Servant) agent is the gateway agent of the FIPA-compliant gateway placed on every Information-server. The purpose of this agent is to server URA with information gathered from foreign MAS. When URA needs to access an EX MAS, it passes its request to the URAS which is responsible of fulfilling URA’s request. Ones, URAS has came in contact with the foreign agent of the appropriate EX MAS and has the results requested by the URA, it sends them back to URA. Until URAS has not acquired the results requested by URA, URA is free to continue with its next task (if it has one), migrate to another information-server or wait for URAS agent’s response.

6. CONCLUSION AND FUTURE WORK
In this paper we have presented an alternative approach to conforming a MAS into a FIPA-compliant one with the use of gateways, which behave like wrappers between the non-FIPA compliant systems and FIPA ones. The advantages of adapting this technique have also been stated. The SARA Active Digital Library, has been used as a test-bed for our approach. In the future, we would like to test and evaluate the interoperability of our system using AgentCities. Secondly, we would like to further extend the interoperability of the FIPA-compliant gateways to provide support for mobility between heterogeneous MAS build on the same Agent platform.

7. REFERENCES
http://www.agentcities.org


http://www.comnets.rwth-aachen.de/~cameleon/

http://leap.crm-paris.com/agentcities/Resources/resources.html


http://www.fipa.org

http://www.fipa.org

http://www.labs.bt.com/profsoc/facts/

[9] FIPA-NET.
http://casbah.ee.ic.ac.uk/fipanet

[10] FIPA-OS.


http://www.fipa.org


http://www.cs.umbc.edu/kqml/

http://www.omg.org


