Towards a FIPA Approach for Mobile Ad hoc Environments

Michael Berger[1], Michael Watzke[1], Heikki Helin[2]

[1]Siemens AG, Corporate Technology , Intelligent Autonomous Systems Group, Otto-Hahn-Ring 6, 81739 Munich, GERMANY

[2]Sonera Corporation, Emerging Technologies and Innovations, P.O.Box 970, FIN-00051 Sonera, FINLAND

Email: {M.Berger, Michael.Watzke}@siemens.com, Heikki.J.Helin@sonera.com

Abstract

FIPA (Foundation for Intelligent Physical Agents) is a standardization organization promoting development and specification of agent technologies. This paper describes an ongoing technical work within FIPA enabling FIPA-compliant agent systems to operate in mobile ad hoc environments. The goal of this work is to provide a high-level framework for efficient interoperability of software services provided by FIPA agents in these environments.

1. Introduction

FIPA (Foundation for Intelligent Physical Agents) [1] is an international standardization organization promoting development and specification of agent technologies. The emphasis here is on the practical commercial and industrial use of agent systems. By following the FIPA standards, implementations are able to guarantee the interoperability with other FIPA-compliant systems. FIPA has recognized the importance of mobile computing and provides agent standard solutions like a bit-efficient ACL and envelope encoding for connections with low bandwidth [2, 3]. These solutions have been proved to be efficient in environments where slow wireless networks are involved [4]. In contrast to that, until now FIPA has no solutions for agents interoperating in “mobile ad hoc computing” environments, a currently upcoming and even more exiting and promising agent application area.

Mobile ad hoc computing is possible because of new technologies for short-range wireless data communication technologies and enables new applications. Mobile devices, equipped with the same type of that technology, make the communication and collaboration of two devices or the establishing of an ad hoc group with more than two devices possible as soon as the devices coming in communication range. The resulting “mobile ad hoc network” (MAN) is very flexible because it has a dynamic topology where nodes are free to move arbitrarily and it allows a Peer-To-Peer (P2P) communication in an asynchronous manner without any pre-installed networking infrastructure.

In mobile ad hoc environments each of the devices may host agents offering specific services to the surrounding. These can directly be used or may be combined to more complex services after discovery by remote agents/services. Several different technologies were developed to describe as well as discover and share services. Some of them provide an API to infrastructure elements (e.g., Jini [5], Salutation [6]), others provide no infrastructure but specific protocol implementations needed on every device (e.g., UPnP [7], Bluetooth SDP [8]).

Working in an ad hoc and short-range area is independently of having a wide-range connection in parallel or not (e.g., there is no coverage or a user does not want to establish such a connection because of the cost).

In parallel to these developments, also dynamic service discovery technologies, such as SLP [9], JXTA [10], or Gnutella [11], were developed for fixed (Internet based) P2P-networks handling the dynamic availability of nodes. The variety reaches from approaches with central elements, over pure P2P solutions until advanced P2P systems which distribute/replicate the service directory entries in an intelligent way. Because of the same nature, technologies developed for fixed P2P-networks can in general also be used for mobile ad hoc networks.

The rest of this paper is organized as follows. Section 2 gives an informal overview of FIPA’s agent platform architecture approach. In Section 3 we outline possibilities for standardization in mobile ad hoc networking environments. In Section 4 we give a detailed description of one possibility followed by application examples. Finally, Section 5 concludes the paper.

2. FIPA’s Agent Platform Approach

The FIPA Agent Platform (AP) provides an infrastructure for deploying agents. FIPA does not specify the internal design of an agent platform, because FIPA's main concern is about achieving interoperability between APs. However, FIPA requires that every FIPA-compliant AP implements three capabilities: The mandatory Agent Management System (AMS, white pages), the mandatory Agent Communication Channel (ACC), and the optional Directory Facilitator (DF, yellow pages). The AMS is responsible for agent creation and deletion, maintenance of a white-page directory service where every agent residing on the agent platform has to be registered with, and the agent life-cycle management.

Based on the traditional DF definition, the search of remote services is accomplished by using DF federations: DFs—besides registering services offered by local agents—may also register other DFs. This allows them to extend the search for services to remote platforms. This mechanism is not efficient, even less for mobile ad hoc environments, for example, because the searcher first has to find the remote DF and afterwards to look if the service he is searching for is registered there. Allowing registering and discovering agent services using existing P2P/ad hoc discovery technologies that are specifically developed for these environments can enable a more efficient management of service descriptions and directories. Furthermore, ad hoc and P2P technologies can also be used transparently as mechanisms for agent (platform) societies in the fixed network. Nevertheless, mostly the DF implements a very efficient search and matching mechanisms by using specific high-level agent descriptions.

An example of a distributed FIPA-compliant AP is JADE-LEAP [12, 13], a scalable AP that allows running agents hosted on PCs as well as on small mobile devices such as PDAs or mobile phones. The JADE-LEAP agent platform spans several hosts distributed over the network. On each host, there is running an agent container, which provides agent communication within the platform and, if the host is powerful enough, agent communication between different APs. One of these containers has to be a main container that is emphasized against the other containers. The main container hosts the AMS and a default DF. Therefore, a container hosted on a mobile device that wants to communicate in an ad hoc manner with an agent living in another AP, may need a non-ad hoc connection to the main container of the platform.

MicroFIPA-OS [14] is a similar FIPA-compliant AP than JADE-LEAP. But it is targeted to more powerful devices, yet supporting PDA devices.

3. Aims and Options for Agent Standards in Mobile Ad hoc Networks

The production of standards is a very difficult process, because a standard is all about system interoperability and should be widely accepted by the community. The acceptance of a standard depends on the applicability and flexibility of the specifications as well as its compliance to already existing standards. That means, the standard should reflect just a minimal set of specifications which allows the interoperability of systems but does not restrict application developers. To find out the right balance is very difficult. It depends on the variety of options we have for a standard. The options within FIPA ad hoc will be shown in the following section.

3.1 Options to Consider for Specification

The service discovery may be executed on the FIPA level, meaning after a very FIPA specific protocol. On the other hand, the discovery may rely completely on existing P2P/ad hoc mechanisms, in case they provide the necessary functionality. Also a mixed solution could

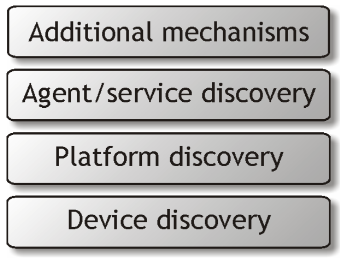

Figure 1: Discovery levels

be possible. Concerning the existing mechanisms, the problem is that the development of dynamic service discovery technologies is still an ongoing research topic. It is not yet presumable, which technology will finally be widely adopted and be the leading one. All of them have specific advantages and disadvantages and do not completely fit all requirements.

Both, a FIPA-specific approach as well as a solution based on existing P2P/ad hoc mechanisms has to consider the following topics:

a) Discovery: devices, agent platforms, agents/services, and additional mechanisms

In the agent domain we can differentiate between several levels to describe and discover entities in the network (see Figure 1). Firstly, we can *discover devices* in communication range. Mechanisms providing that functionality are for example Bluetooth, IrDA and WLAN. But we are not focusing on low-level device discovery. Secondly, a *discovery of agent platforms* is possible. APs are to discover for example for mobility reasons in case that one agent wants to move to another side or in order to connect platforms for service discovery. Thirdly, we can *announce and search for agents*. Agents have a specific role and are needed by other agents in order to cooperate and execute complex behavior. Services are much more basic than agents. For example, they provide simple functionality like a specific calculation. Agents may provide services or may wrap legacy services. Finally, we can define *additional features*, which for example comprise mechanisms to support policies and security in ad hoc environments. In this paper we are focusing on the agent and platform discovery.

b) Resource description/matching

A discovery should get a result which is exactly the one the requestor expects. The quality of the result depends on the ability of matching request and supply in an intelligent way. *Meaningful description of platforms, agents, and services* are required. While the final matching cannot be standardized, FIPA should focus on the information for describing “resources”. In case of using existing mechanisms, there are big differences in the possibility of supply and request descriptions. A description may be based on pre-defined IDs, on simple strings or on more complex descriptions using attribute-value pairs (see also Table 1).

c) Handling Mobile devices and mobile communications

Other parameters can be extracted based on the consideration of mobile devices. The existence of mobile devices has two main impacts on the decision. Firstly, mobile devices may have very hard *restrictions on local resources like memory and processor speed*. In that case it could be possible, that there will be no local DF and the agents know each other directly. In that case, the search over the network cannot be based on a DF federation. A compromise could be that a local DF can be launched depending on the amount of local agents. Furthermore, the installation of complex mechanisms on the mobile device, for example several connectors to underlying discovery mechanisms, is not possible. Secondly, the *communication costs* of the discovery should be considered. Minimal communication amount and short messages are needed. The better the mechanism fits to these requirements, the faster the mobile community will accept the solution.

d) Scalability

The scalability of the solution should be considered as well. There are two scalability factors. One is, that the solution may have *many participants*, including devices, platforms, agents, and services. In an extreme case, all of these resources may be announced and mechanisms should be available to announce only selected resources. A second scalability factor is the *dynamics of the networked participants*. This can be very high (there will be no stable network because participants always join and leave) or it may be quite low meaning the participants are not moving. Certainly, there are different discovery mechanisms for both extremes optimal. For example, some are not dealing well with MAN's spontaneity of the peer communication and fast changing service provisioning while others are not dealing well with the scalability for a huge amount of resources. The problematic here is, to decide whether one or more mechanisms are suitable or to define a FIPA specific approach.

3.2 Description of Derived Architecture Proposals

3.2.1 Discovery using existing P2P/Ad hoc Platform

The discovery using existing underlying P2P/ad hoc platforms can be based on three different models: AP discovery, DF discovery, and agent discovery.

a) Architecture using AP discovery

In a first model (see Figure 2 (a)), an AP_1 registers itself to the underlying architecture (1) and a different platform can discover it by requesting the underlying architecture (2). After receiving the search result (3) the AP_2 can proceed in the way defined in FIPA by federating DFs and searching for agents directly.

The registration is done using the functionality provided by the underlying architecture. The information the AP_1 registers is the "AP description" specified by FIPA [15] mapped to the native service description (of the underlying platform). This mapping obviously is needed for each P2P/ad hoc platform that FIPA will support. The AP description contains enough information so that the resource is identified unambiguously as FIPA AP. Other APs can use this information to discover the address of the AP. However, there is no need to map arbitrary FIPA service descriptions to native service descriptions, which is potentially more complicated. The advantages of this approach include, that the FIPA-level search functionality will be preserved, i.e., the agent and service search is done using powerful FIPA methods, while the approach still leaves open how the "FIPA-level" service discovery is actually done (e.g. by DF federation, or direct search in more dynamic environments). The model requires access to the search functionality of the underlying platform, but only few FIPA related registrations are needed in the underlying platform.

b) Architecture using DF discovery

In a second model, local DF(s) of all APs register themselves to the underlying platform. This is done similarly than AP registration in the first model, that is, mapping

Table 1: Comparison of discovery models using underlying P2P/ad hoc platform

	Model 1	Model 2	Model 3	Model 4
Mapping to service description of underlying platform	Easy, AP description	Easy, DF description	Complicated, Agent description	Combination of 2 and 3
Requested search quality	Simple underlying mechanism for AP search, DF for advanced agent discovery	Simple underlying mechanism for DF search, DF for advanced agent discovery	High requirements for underlying search mechanisms	Combination of 2 and 3
FIPA specifications needed	Interfaces to underlying technology + search mechanism on DF/agent level	Interfaces to underlying technology + search mechanism on DF/agent level	Interfaces to underlying technology	Combination of 2 and 3
Optional DF	No	No	Yes	Yes
Mobile communications	Yes	Yes	No	Yes
Many agents	Yes	Yes	No	Yes
Dynamic environment	Yes	Yes	No	Yes

(a) (b)

Figure 2: (a) Architecture using AP discovery, (b) Architecture using FIPA-defined AP discovery

between DF description [15] and the native service description of the underlying platform is needed. As in the first model, this mapping is needed for each underlying platform supported by FIPA. Other DFs can search DFs by using the search functionality provided by the underlying platform and then federate. This model is almost that simple as the first model and all the advantages from that model apply also to this model except the following extensions: The additional step of AP discovery is not needed by direct discovery of DFs. But the approach requires a DF on each AP (while according to the FIPA specification the DF is optional).

c) Architecture using agent discovery

A third model considers that agents residing in any FIPA AP register themselves directly to the underlying platform. Further, agents use the search functionality provided by the underlying platform to search other agents and services. This requires a mapping between the FIPA service description [15] and the native service description.

Compared to the first two models, there are some differences. Firstly, although this model does not rely on a DF any more, one can still use a DF. Secondly, the mapping between a FIPA service description used by the agent and the native service description is potentially hard. Furthermore, this model relies on search functionality provided by the underlying platform. Especially, it is not clear how well the search can be actually done using the information provided in FIPA service description after mapping that to the native service description. For example, more intelligent and semantics based search like *"give any agent implementing ontology Foo that can speak FIPA-SL0"* are not yet available in existing P2P/ad hoc platforms. On the opposite, FIPA's DF is able to handle that complexity. Finally, the number of agent registrations to the underlying platform is (potentially) large, which is not very useful in case of mobile communications.

d) Architecture combining AP, DF and agent discovery

The fourth model is a variation of models 1, 2 and 3 (AP, DF and agent discovery). This model is needed, when different circumstances occur: e.g., a DF is not available on every platform, some platforms have too many local agents but have only a mobile connection and want to register just the local DF with the underlying platform. Others are having an unstable connection and a DF federation takes long in such dynamic environments with many DFs. So they are registering all agents directly to the underlying platform. Otherwise, there is also a communication overhead using DF federation (first federate message and then search message), which is not useful for mobile communications. So the use of DF registration and federation depends on the communication channel, the dynamicity of the environment and the number of agents. Table 1 summarizes the features of the four models described above.

3.2.2 FIPA-defined Discovery

The discovery in a "FIPA-defined" way is a variation of models one to four in the previous section, with the difference that it does not rely on an underlying P2P/ad hoc platform. This implies that FIPA must define all the details related to platform, DF and agent discovery. Nevertheless, some existing discovery mechanisms can be used as a basis. Already at that point we can make the statement, that it is not a good way to go, because we should prevent to reinvent the wheel. Furthermore, it this case, we have to consider too low-level details from FIPA's conventional viewpoint and that is potentially complicated. But in contrary, there are some advantages:

The FIPA-defined solution is independent of any existing P2P/ad hoc platform, although we may have to make some assumptions like existence of broad/multicast support. Furthermore, in case of AP discovery (see Figure 2 (b)), there is no need to use ACL since platforms are talking to each other; not agents. In case of DF and agent discovery/announcements, potentially the discovery is more complicated because we work on the agent and ACL level, which can run into problems with its semantics (e.g. broadcast/multicast of an ACL is not defined).

3.2.3 Comparison of Existing Discovery Mechanisms

The previous sections described possible architectures, some of them with a reference to existing P2P/ad hoc

Table 2: Comparison of existing P2P/ad hoc platforms

	Bluetooth	UPnP	JXTA	Gnutella
Resource description/matching	UUIDS, name-value pairs	Service type XML	name-value pairs (XML)	simple string
Mobile devices	Yes	Yes	Yes	Yes
Mobile communications	Yes	No	No	No
Scalability (amount)	Yes	No	Yes	No
Scalability (dynamicity)	Yes	Yes	Yes	No

platforms. Table 2 gives a short comparison/evaluation of four existing and pre-selected P2P/ad hoc platforms.

The evaluation shows, that no single technology fulfills all requirements. Although JXTA provides the best support, FIPA cannot decide for just one technology, even if it is not a widely accepted standard like Bluetooth. The requirement to FIPA here is to provide a generic solution supporting all or several of these technologies.

4. Current Work and Prototypes

Currently there are different approaches known as proposals to FIPA (see [1]). One of them will be described below, followed by application examples.

4.1 FIPA Specific Approach

The proposal by Siemens assumes that a mobile device hosts a complete agent platform obeying the guidelines of the FIPA abstract architecture [16], and tries to minimize the impact on existing FIPA specifications. In their model, each agent platform periodically broadcasts an *agent platform announcement message* of itself. In this model, a specific ad hoc management service agent sends out the AP announcement messages periodically. Simultaneously, it listens for incoming announcement messages. If at least two mobile devices are in range for ad hoc communication, both agent platforms receive the AP announcement messages and become aware of each other. Having discovered each other, the APs form a logic AP federation. During that AP federation it is not necessary to federate the AMS of each platform, but the DF of each platform have to be federated in order to make agents of each platform accessible by the services they provide.

For sending AP announcements on ACL message level, it is necessary to specify a broadcast/multicast-capable message transport protocol, i.e., a protocol that both can send out and receive AP announcements messages to/from a given broadcast/multicast channel. HTTPMU (UDP multicast of HTTP messages) as specified by UPnP Forum [7] is one possibility that could be used here. This is especially suitable since the FIPA standard already defines a message transport protocol for the HTTP specification. Further, compared with broadcast, multicast seems to be more flexible, because there is a large number of IP multicast addresses (Class D IP addresses) available which could avoid interference between several ad hoc communication groups, each of which would have agreed to send out and listen for agent platform announcement messages on a specific channel. Even more important, multicast provides a time-to-live counter and is not restricted to a certain sub-net.

Technically, the AP announcement message is an ACL message with a REQUEST performative. The content of the message contains an action "notice-ap-description" as well as necessary AP information. One important aspect of the AP announcement message is, that an AID of a receiver agent has to be specified in the receiver slot. In case of multicasting ACL messages it is a priori unknown. For this reason, a particular surrogate AID ("*any-AID*") is introduced, which matches the ad hoc management service agent in each remote agent platform in ad hoc communication range. The surrogate AID is also used as the actor in the action expression.

The ACC of an AP must be able to dispatch a received AP announcement to the local ad hoc management service agent. Thus, the ACC has to map the surrogate AID (matching *any* ad hoc management service agent) to the local ad hoc management service agent's AID. The current FIPA ACC specification has to be extended in order to deal with such surrogate AIDs.

In order to set up a DF federation, the ad hoc management service agent registers the remote DF with the local DF. The name of the remote AP description can be evaluated from the AP description received with the announcement message.

In order to decide whether a remote AP is no longer in ad hoc communication range or not, for each remote AP that has been identified, the ad hoc management service agent maintains a lease table. If a new remote agent platform has been detected, a lease with an initial age is created for it and put to the lease table. The lease table is updated periodically. For every lease it is checked, whether it has expired. If a lease has not yet expired, its age is increased. The age of a certain lease is reset to its initial value whenever an AP announcement from the corresponding remote AP has been received again. If a lease has expired, it is removed and the remote DF is deregistered from the local DF. It is important to note, that until lease expiration, there can occur inconsistencies between the DF federation and the ability to communicate in an ad hoc communication range, i.e., agents living in a remote AP may not be able to communicate with in spite of their remote DF is still registered with the local DF. Similarly, this condition may hold during DF search. Thus, implementations of DF search as well as communication between application agents have to take into account, that a remote agent may not be reachable anymore. To avoid such inconsistencies as far as possible, an additional mechanism could be added, that deregisters a remote DF, if one of its registered agent is detected to be no longer in communication range. But this would require to locally maintaining information whether an agent is registered with the local or the remote DF.

4.2 Implementation, Application Examples

Siemens has implemented a first demonstrator as a proof-of-concept, which follows the FIPA-specific proposal. It is based on JADE-LEAP. As for the hardware, the demonstrator runs on mobile devices that support ad hoc communications based on WLAN (IEEE802.11) and provide a Personal-Java platform, such as the Siemens SX45 or Compaq IPaq. Along with the given proposal, each mobile device has a complete JADE-LEAP AP running on it. As for the AP itself, only minor modifications of the ACC have been done in order to handle surrogate AIDs. The ad hoc management service component has been realized as an agent.

Additionally, an implementation of HTTPMU is provided. One class D IP address is used as pre-defined multicast channel, on which all mobile devices are broadcasting its own agent platform announcement message and are listening for incoming ones. For reliable inter-platform communication, HTTP is applied with non-routable (private network) IP addresses.

As for an application, there has been implemented a simple demo showing that a DF of a remote agent platform is available and has been federated with the local DF. On top of that, we implemented a second demonstrator "Mobile Ad" which shows mobile advertising. People having that application on their device may edit a profile which is describing which goods they advertise for which price and which they are looking for. During walking on the street, the mobile application detects other users and matches the profiles. If there are common interests in selling and buying, the contact information will be exchanged.

In addition, as a prototype of a demonstrator for the Cebit exhibition in February 2002 we developed an application called "Jukebots" giving a mobile phone (SL45i) the capability of generating and adapting automatically a music profile for its user. With the profiles of users for example in a bar scenario, their mobile phones will decide which songs are to be played by the disc jockey (human or computer) through a partly automated voting process. Currently we port that example to mobile devices (IPaq) with WLAN communication facilities and used it in an ad hoc environment, which is much more near to bar environment.

All demonstrators showed the capabilities of ad hoc communication with mobile devices in combination with agent technology. The LEAP platform enabled the application programmer to use high-level agent development on small devices together with PC/Laptop devices and ad hoc communication to build a reliable agent environment. Further examples can be seen in office environments (e.g. control of printing devices), home environments (control of TV) or in hospitals for managing medical devices.

5. Conclusions

We described an ongoing technical work within FIPA standardization organization enabling FIPA-compliant agent systems to operate in mobile ad hoc environments. This work is towards providing a high-level framework for efficient interoperability of agent-based software services in these environments. Given the highly dynamic and complex nature of that domain, software agent technology seems to be ideal for implementing adequate services and applications.

Agents operating in mobile ad hoc networks need support for discovery on several layers. We outlined two general approaches: solutions using existing P2P/ad hoc platforms and solutions in which FIPA defines all details regarding the discovery. The evaluation shows that no single approach fulfills all requirements and is appropriate for all mobile environments. Standardizing one existing underlying P2P/ad hoc platform has the disadvantage to focus on just one restricted technology, which is hard if not possible to decide. On the other hand, defining the discovery details on FIPA level is not appropriate for a standard organization. We conclude that mapping FIPA concepts to several existing P2P/ad hoc architectures is the best possible way to solve this problem. However, further research is needed in order to select the appropriate FIPA strategy according to the requirements of the application environment.

6. Acknowledgements

The authors are grateful to members of FIPA TC/WG Ad hoc, especially to Celeste Campo, Heimo Laamanen, Jamie Lawrence, Michael Pirker, and Olga Ratsimor.

7. References

[1] FIPA. FIPA Home Page, 2001. <`www.fipa.org`>
[2] FIPA. *FIPA ACL Message Representation in Bit-Efficient Specification*, Dec. 2002. Specification number SC00069.
[3] FIPA. *FIPA Agent Message Transport Envelope Representation in Bit Efficient Specification*, Dec. 2002. Specification number SC00088.
[4] H. Helin and M. Laukkanen. Performance Analysis of Software Agent Communication in Slow Wireless Networks. In Proc. of The 11th International Conference on Computer Communications and Networks (ICCCN 2002). pp. 354-361, Oct. 2002.
[5] K. Arnold, *et al. The Jini Specification*. Addison-Wesley, 1999.
[6] The Salutation Specification. <`www.salutation.org`>.
[7] The Universal Plug and Play Forum Home Page. <`www.upnp.org/`>.
[8] The Bluetooth Specification. <`www.bluetooth.com/dev/ specifications.asp`>.
[9] E. Guttman, *et al.* Service location protocol, version 2.

Request for Comments 2608. 1999.
[10] Project JXTA Home Page. <www.jxta.org>
[11] Gnutella Protocol Specification v0.4. <www.stanford.edu/class/cs244b/gnutella_protocol_0.4.pdf>
[12] M. Berger, *et al.*, Towards an Agent-Based Infrastructure for Distributed Virtual Organizations. In Proc. of the 10th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (Wetice 2001), pp. 354–355. 2001.
[13] F. Bergenti, *et al.* Porting Agents to Mobile devices, IEEE Internet Computing, 2001.
[14] M. Laukkanen, *et* al. FIPA-OS Agent Platform for Small-Footprint Devices. In *Intelligent Agents VIII, Proceedings of the Eighth International Workshop on Agent Theories, Architectures, and Languages (ATAL-2001),* pp. 447-460. Springer-Verlag: Berlin, Germany, 2001.
[15] FIPA. *FIPA Agent Management Specification*. Dec. 2002. Specification number SC00023.
[16] FIPA. *FIPA Abstract Architecture Specification*. Dec. 2002. Specification number SC00001.