# Requirements for Disconnected Mobile Agents within Cellular Mobile Telecommunication Systems

Jens Hartmann
Ericsson Eurolab Deutschland GmbH
Ericsson Allee 1, 52134 Herzogenrath
Jens.Hartmann@Ericsson.com

Carsten Pils
Informatik 4 (Communications Systems)
RWTH Aachen, 52056 Aachen, Germany
pils@i4.informatik.rwth-aachen.de

14th September 2000

## Abstract

Mobile agents offer unique opportunities for structuring and implementing distributed communication systems. With their special ability to operate autonomously and disconnected they are identified to suit mobile environments well. Standardisation bodies like the Object Management Group (OMG) and the Foundation for Intelligent Physical Agents (FIPA) have addressed agent management in general, but unfortunately implementation details are lacking. Not discussed at all is the management and control of mobile agents within an intelligent communication system supporting both nomadic users and mobile devices. This paper defines 14 requirements for building agent-based systems in a mobile environment taking into account terminal mobility and personal mobility.
**Keywords:** Mobile agents, disconnected operations, terminal mobility, personal mobility.

## 1 Introduction

The current telecommunication market is shifting from providing basic teleservices towards the use of value added services through multiple heterogenous communication networks. Competition and deregulation cause a growing demand of operators and service providers to increase their service portfolio to gather additional revenue. This may be achieved by differentiating and enlarging the offer of personalised end-user services. However, as information technology (IT) and telecom platforms converge, operators and service providers need to plan their networks to adapt to these new services. Flexibility and responsiveness have to be built into the network and this is an area where service solutions are a great help.

Agent Technology [WJ95], especially Mobile Agent Technology (MAT), is often seen as an enabler for future communication concepts [HGF98] and might pave the ground for flexible environments, where known and trusted agents serving real user demands. Hence, network and service provider are investigating scalable agent-based communication concepts in depth in order to offer their customer adaptable environments allowing nomadic communication, because MAT is advantageous when it comes to transporting or processing data. The transport of an agent instead of the transport of the data which are to be processed can lead to less network load and less costs for operators as well as customers. Furthermore, agents are best suited for future packet-switched telecommunication networks such as GPRS and UMTS [HEG+99].

A mobile agent is a program roaming the network under its own control on behalf of its owner. It can migrate from host to host and interact with other agents and resources on each host. The agent should be able to execute on any machine within a network, regardless of the processor type or operating system. The agent system itself should not have to be installed on each machine the agent could possibly visit, which means that an agent system should be implemented on top of a mobile code system (example: Java Virtual Machine).

A mobile agent operates independently of the application from which the agent was invoked. The agent operates asynchronously, meaning that the client application does not need to wait for the results. In particular this is important for users who could not always be reached by the network. Remote programming allows a user to delegate a task to an agent. The communication device must be connected to the network only long enough to send the mobile agent on its way and, later, to take it home. It does not need to be connected while the agent carries out its assignment.

The remainder of this paper is organised as follows: in the next section we introduce an service scenario model which is the basis for our requirement analysis in sections 3, 4, and 5 . In section 6 we summarise our results and give a brief outlook of our future work.

# 2 Agent System Model

The agent system model investigated in this paper complies with the OMG's MASIF specification [OMG98] and enhances it to model disconnected operations. MASIF specifies:

*An agent system is a platform that can create, interpret, execute, transfer and terminate agents.*

Each agent system has access to communication infrastructure, e.g. for agent transfer. The MASIF specification does not distinguish between wired and wireless communication infrastructure since the MASIF specification focuses on agent system interoperability. But with our application model we want to utilise the major advantage of mobile agents with respect to mobile networks which is disconnected operations [CHK94]. Now, considering an infrastructure consisting of a wired core/backbone network and a wireless access network we must distinguish between agent systems which are connected to the agent system world by a wireless link and those which are connected by wired links. The presence of agent systems which are connected to the agent world by a wireless communication infrastructure has a severe impact on agent system fault tolerance since these systems are prone to frequent disconnections.

## 2.1 Application model

Apart from the distinction between wired and wireless connected agent systems we distinguish also between agent systems which are rarely being shut down and those which frequently shut down. Systems which are rarely shut down include those agent systems which are, for example, maintained by commercial service providers or commercial users. We suppose that these agent systems are connected by wired links and being only down for administrative purposes. Systems which are frequently down are in general maintained by service users who switch on their device on service need. Based on these assumptions we distinguish four agent system types:

- **Type A** systems are connected to a highly reliable wired network to the agent world and are rarely down. These systems are maintained by commercial users and service providers.
- **Type B** systems are connected to reliable wired networks but they are frequently down. Type B systems are used by service users who use these systems just on demand.
- **Type C** systems are connected to wireless access networks. Type C agent systems are highly reliable and rarely down.
- **Type D** systems are connected to wireless access networks and are frequently down. These systems are for example used by subscriber of cellular mobile networks such as GSM. Thus, agent transmission is currently barely possible without building up a dedicated traffic channel, which is not always feasible.

In figure 1 different agent system types and their relationship to each other are depicted. We assume that type A systems form the core agent world and

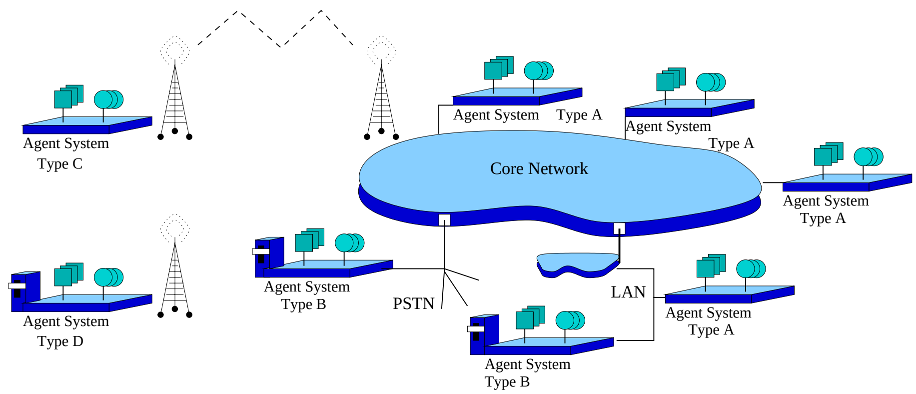


Figure 1: Agent system types

are connected through reliable high speed links or attached by LANs and reserved dial-in connections.

## 2.2 Agent system concepts

We distinguish two agent system concepts for disconnected operation support. The distinction is related to the level of agent migration. The first agent system concept which we address allows mobile agents to migrate to every agent system in the agent world. Thus, we call this concepts in the remainder of this paper *unrestricted migration concept*. Unrestricted migration causes problems in respect of type C and D agent systems. It is questionable whether it is useful to allow mobile agents to migrate to type C, D agent system: First of all it is not certain whether a mobile agent gets the chance to leave a type D mobile agent system once it has arrived there. Second most of the type D mobile agent systems reside on mobile devices with limited resources. Hence, agent system owners might prohibit mobile agents to access their scarce resources. This situation leads to a *restrictive migration concepts* where mobile agents can migrate only to type C, D agent systems to return results, or are not allowed to migrate to type C, D agent systems at all. In this paper we focus on the *unrestricted migration concept*. There are three reasons which lead to this decision:

- First of all we expect that the resources of mobile devices will be sufficient in the near future.
- Apart from agent systems residing on PDA's also systems residing on notebooks which have a wireless Internet access belong to the type D agent system class. These devices have enough resources to maintain an agent system and allow unrestricted migration.
- Restrictive migration is a special case of unrestrictive migration. Therefore, if problems are solved for unrestrictive migration problems of restricted migration are solved, too.

# 3 The impact of terminal mobility on mobile agent migration

Terminal mobility addresses the issue that users roaming around globe with mobile terminals while making use of communication services. In an agent world consisting of type A and B agent systems the likelihood of disconnection or shutdowns of agent systems is quite low. Currently most MATs are addressed to this scenario [CAM00], [LPP99], [Obj00], [IKV99], [IBM00], [IPV00]. Since these events are rare, there are no special mechanism to handle them. Only network errors are considered in these agent system technologies and thus most of them utilise a two-phase commit protocol to ensure an error-free agent migration. Type C and D agent systems introduce a new level of uncertainty to an agent world. This means that it is uncertain whether a migration can be completed or an agent systems which is connected to the agent world now will not disappear in future. Note that a type D agent system can shut-down while disconnected and thus has no possibilities to publish its shut-down to the agent world.

This uncertainty influences not only the agent system management, but also agent programmers, mobile agent users and type B, C, D agent system owners.

## 3.1 Agent programmers requirements

Agent programmer have to consider that a migration might fails due to a temporary disconnect of the destination system, or the destination system is down, or does not exist. If the agent system indicates a migration failure reason to mobile agents then they can decide how to handle the situation best. Obviously, there are two alternatives to handle migration failure: First the agent can decide to wait until the destination agent system is available again and second the agent can decide not to migrate to the destination but do something else. To wait until the destination system is available is not the best solution in all cases: Waiting for an agent system getting available again which does not exist is useless while waiting for an agent system which is temporary unavailable for a certain time might sometimes be useful. Not to wait until the destination is available again is also not the best solution in any case: If the destination agent system is disconnected for some seconds due to interference then the mobile agent should wait for a certain amount of time. It follows that an agent system which supports disconnected operations should indicate the agent the migration failure reason which can be disconnect, down, or not existing. Surely, these failure reasons are also uncertain, a disconnected agent system can be down, but it helps the agent programmer to handle migration failures. Thus, we define our first requirement of an agent system technology supporting disconnected operations:

Requirement 1
**(Migration failure reason indication)**
*On migration failure the agent system must indicate the mobile agent whether the reason of the migration failure is that the destination is disconnected, down, or does not exist.*

As mentioned above depending on the failure reason there are two alternatives to handle the migration failure: waiting and other actions. Since it depends on the agents task which alternative to choose agent systems should always indicate the reason to the agent and let it up to the agent and its programmer respectively to decide what the agent should do. Thus, an agent system supporting disconnected operations should allow the agent to take individual actions on migration failure. This leads to our second requirement:

Requirement 2
**(Individual migration failure handling)**
*The agent system allows the agent to take individual actions on migration failure.*

Another important point is that the agent system technology should support agent waiting for each system. We discuss this point in section 3.4 agent system management requirements since waiting agents influence the agent systems resource and connection management.

Apart from failure reason indication and individual migration failure handling the agent system can support the agent programmer by indicating the destination systems type. That means that the agent system should provide a mechanism which allows the mobile agent to check the system type of any agent system any time. Agent system indication would allow agents to prevent migration to the critical type C, or D agent systems. By simply checking whether the destination system is a type C, or D system in advance of a migration the agent can prevent migrating to these systems and take alternative actions.

Requirement 3
**(Agent system type indication)**
*The agent system provides a mechanism indicating the agent system type to mobile agents.*

*Agent system type indication* allows mobile agent to prevent migration to type C, or D agent systems but preventing type C, or D agent systems is not a solution in general. If an agent cannot prevent migrating to type C, or D agent system then the agent should know in advance of migration whether the migration is expected to be successful in a given time limit. This information would help the agent selecting between alternative type C, or D agent systems, or the agent could adapt to network conditions by reducing its data-base and data structure. Since network conditions change frequently it is not possible to make precise predictions. Therefore, it might happen that a mobile agent decides to migrate to a type C, or D agent system assuming appropriate network conditions, but while being in transfer the network conditions are getting worse. If the agent would have known the network conditions in advance it would not have decided to migrate, but while being in transfer and being inactive the agent has no choice but waiting until the transfer completes. A solution to this problem would be that the agent could specify transfer requirements to the agent system. If the agent systems detects that the specified requirements cannot be met then the agent transfer is aborted and a message is sent to the agent indicating the migration failure.

We summerise these observations in the following requirement:

REQUIREMENT 4
**(Network sensitive migration)**
*An agent can ask the agent system for the expected transfer times in advance of migration. To compensate frequently changing network performance agents can specify a requirement set in advance of migration. If the agent system detects that the requirements cannot be met then the agent system aborts the transfer and indicates the migration failure reason to the mobile agent.*

The requirements outlined above allow the agent programmer to take individual actions for each migration of a mobile agent. But for some agents this freedom is not necessary and therefore agent programmers would like to use default handlers or default policies. A default handler could handle each migration exception. If a migration exception occurs the default handler is called which examines the failure reason and performs actions depending on the failure reason. A default migration policy could be, for example, that the agent should never migrate to a type D agent system, or is only allowed to migrate to a type D agent system if the agent has reached a pre-defined state, or an agent is only allowed to communicate via RPC calls with an type D agent system. Default handler and policy definition are not necessary requirements, because with the requirements set described above each agent programmer could implement these features himself.

REQUIREMENT 5
**(Default handler and migration policy)**
*The agent system supports the definition of a default handler to handle migration failures and allows defining migration policies and rules.*

## 3.2 Agent owner requirements

An agent owner launches his agents to perform a task and expects them back after a certain period with results. Annoying for mobile agent owners are mobile agents which are lost somewhere in the agent world. Disconnected operations add another danger for mobile agent loss: Suppose an agent successfully migrates to a type D agent system. After performing its task the agent wants to migrate to another agent system, but in the meantime the agent system is disconnected due to interferences. Since the agent has no alternatives it decides to wait until the agent system connects to the agent world again. While the agent waits the agent system shuts down. The agent is trapped on the agent system. *Agent system type indication* allows preventing agent trapping by avoiding migration to type D agent systems, but in some cases avoiding migration to type D agent systems is not an appropriate solution. Instead there should be a mechanism which starts an agent copy if an agent is trapped. Thus, the agent owner requires that the agent systems support mechanisms handling agent trapping.

REQUIREMENT 6
**(Trapped agent handling)**
*The agent systems provides a mechanism to re-start a copy of a trapped mobile agent and makes sure that there is only one instance of a mobile agent. Hence, a management entity is needed, which tracks each mobile agent migrating to a type D agent system and restarts a copy if the agent is trapped (decision based on watch dog messages or time outs).*

Since the mechanism must make sure that there is just one instance of a mobile agent the question arises what happens to trapped mobile agents which are released again. A simple solution would be that the mechanism makes sure that these agents are killed. This simple solution has the disadvantage that the information which has been processed on the trapping agent system is lost. Another solution could be that the formerly trapped agent provides information to its current copy before being killed.

## 3.3 Agent system owners requirements

We assume that an agent system is used to interact and communicate with agents within its domain. Thus, the agent system provides services and information to the agent world which should be always available to represent the owners interests. A mobile user running a type C, or D agent system on a mobile device is not always connected to the agent world and therefore these services and information are not constantly available. To guarantee that this information is always available a mobile user could hire resources on a type A agent system which holds these services and information and synchronises them with the data on the mobile device.

If a mobile agent wants to communicate with or migrate to an type C, or D agent system and communication or migration fails because the agent system is disconnected then the agent systems user wants that the mobile agent gets a re-routing information to the proxy agent system which holds the synchronised information. Thus, we define our seventh requirement:

REQUIREMENT 7
**(Proxy agent system support)**
*The agent system technology allows definition of a proxy agent system and provides proxy agent system routing information if migration or communication fails.*

The advantage of mobile agent systems in respect of disconnected operations is that mobile agents perform their task autonomously. This allows the mobile device to disconnect and save connection costs. To gain from autonomous agent operations and save connection cost the agent system owner requires that mobile agent systems disconnect if the wireless access channel is not used.

REQUIREMENT 8
**(Connection on demand)**
*Type C and D agent systems should disconnect if the mobile access is not needed.*

Note, that agent system which use GPRS/UMTS packet-oriented access technologies have not monitor the link since the problem is already solved by the system.

Talking about agent system owners requirements we must also take core network providers requirements into account. Core network providers provide disconnected mobile agent management facilities to the agent world like the agent waiting support discussed in the next section. These management facilities consume core network providers resources. Therefore, core network providers need accounting mechanisms for the offered management facilities. Furthermore, there is a need for policies which allow the core network provider to block mobile agents if the core network and its management facilities are overloaded. Accounting and system overload avoidance are no special disconnected mobile agent management requirements. Therefore, we do not add this requirement to our requirement list, but nevertheless we must take it into account when developing management concepts.

## 3.4 Agent system requirements

Agent system requirements addresses the scalability of the agent system in respect of disconnected operations. In section 3.1 we outlined that an mobile agent might decide to wait for a temporary disconnected agent system. If this waiting is a busy waiting then the agent consumes agent system resources. Since the time until the destination agent system is available again has no upper limit, agent systems are in danger of getting blocked by waiting mobile agents. Thus, the agent system should provide a mechanism which allows suspending agents while the destination agent system is disconnected. This would allow the agent system to store waiting agents persistently and free up memory and processing resources. To guarantee scalability the agent system should be able to move waiting agents to other agent systems if a certain number of waiting mobile agent is reached. The agent system should also limit the waiting time of mobile agents to an upper limit to prevent that agents wait the entire life time of the agent world. If there is no upper limit then the agent world is in danger of being blocked by entirely waiting mobile agents.

In some cases users require that agents either return results within a predefined time or send a status reports. Thus, it is required that mobile agents also can define a maximum waiting time themself.

REQUIREMENT 9
**(Agent waiting support)**
*The agent system should provide an agent suspension and wake-up mechanism for mobile agents which wait for a type C or D agent system to connect to the agent world. Furthermore, it is required that the mechanism:*

- *wakes the agent if the destination agent system connects*
- *restricts the maximum waiting time of a mobile agent to an upper limit*
- *allows the agent to specify a maximum waiting time itself*

Since type C and D agent systems are prone to frequent disconnections, mobile agent which migrates from a type C, D agent system to another type C, D system have a high probability of migration failure. Therefore, it is required that a mobile agent which migrates from one type C, or D agent system to another one transparently migrates to an intermediate type A agent system.

REQUIREMENT 10
**(Disconnected migration support)**
*If an type C, or D agent system detects that a mobile agent migrates to another type C, or D agent system then the agent system transparently starts migration via an intermediate type A agent system to enhance the probability of a successful migration.*

# 4 The impact of personal mobility on mobile agent migration

Personal mobility allows a user to access communication services through different devices, e.g., at home, at office, mobile phone, notebook, PDA. This means that the user has several different access points. Communication directed to the user should be presented to him at the device he is currently using. Two aspects related to personal mobility are number mobility and service mobility. Number portability allows a user to originate and receive data and calls at any location by using the same destination number. The success of the 2nd generation mobile communication standard GSM relies, among other things, on the possibility to roam between networks and thus between countries by using a single subscription. This means that subscribers are reachable using a single number and receive a single bill from their home service provider. Service mobility allows a user to access his personal service portfolio independent of the end-device he uses.

Personal mobility like terminal mobility has a severe impact on mobile agent systems. An end user starting a mobile agent expects that the agent returns its results to the user wherever he is and whatever device he uses at the time. Thus, mobile agents need the routing information of the agent system where the user is currently logged in. But the routing information is not sufficient for handling personal mobility: It is senseless if a mobile agent migrates to an agent system to return its results which offers no service to process them. Consider, for example, that some device are not capable to meet the resource requirements of some services, or services are not available on all devices due to licensing policies. Thus, a mobile agent should check in advance whether the agent system where the user is currently logged in can utilise the results.

A mobile agent migrating to an agent system which cannot utilise the agent's result data fails, but a mobile agent migrating to an agent system which cannot handle its resource requirements can even cause agent system failure. If the agent system cannot meet the resource requirements of the mobile agent then the agent fails. But, in case of a Java-base agent system, if the memory requirements of the mobile agent exceed the memory resources of the agent system then the agent system fails too. This problem can be solved by a resource negotiation between the mobile agent and the agent system in advance of migration [KRL+99]. The agent sends a resource request to the agent system and the latter responds with a resource accept message. The agent migrates if the resource requirements of the mobile agent are satisfied. Resource negotiation requires also that an agent system constantly keeps track of the negotiated resource restrictions.

Based on these observations we identify three agent system requirements:

Requirement 11
**(User-routing)**
*The agent system should provide a mechanism to identify the users current agent system.*

Requirement 12
**(Service associated user-routing)**
*A service associated routing mechanism is an enhancement of the user-routing mechanism. The mechanism indicates whether a specific service is currently available at the users current agent system.*

Requirement 13
**(Resource negotiation)**
*In advance of each migration the agent must negotiate with the destination agent system whether the latter can meet the agents resource requirements. Furthermore, it must constantly check the negotiated resource restrictions.*

# 5 Requirements of a user friendly information society

Apart from more technical requirements we must also address the user's need for information selection. In a user friendly information society subscribers should have the opportunity to access information every time and everywhere. We expect that the user does not want to receive each information any time. One can imagine a user who starts several agents during his office hours which finish their task during his spare time. Now, if the user starts an agent system during his spare time the agents try to return to the current agent system. But during his spare time the user just wants to receive very important agents or agents related to his spare time activities; any other agent should return its results during the users office hours. On the other hand, the user does not want or even is not allowed to receive agents which are related to his spare time activities during his office hours. Thus, we assume that an agent system which supports user mobility should provide selective agent blocking. Selective agent blocking is tightly associated with user-routing because an agent returning its results uses user-routing to find the users current agent system and must check afterwards whether the user allows migration. Thus, we do not call this requirement selective agent blocking but selective user routing.

Requirement 14
**(Selective user-routing)**
*Selective user-routing is an enhancement of the user-routing mechanism. Selective user routing allows the user to block or re-route mobile agents migrating to his current agent system.*

# 6 Conclusions

Mobile agents might pave the ground for future communication concepts. Focusing on the unrestricted migration concept and investigating the impact of terminal and personal mobility on agent migration we derived 14 requirements for building agent-based systems with respect to management of disconnected mobile agents in mobile communication systems. Beside the global need of solutions for disconnected operations, we motivated these requirements with the help of a service scenario, in which we distinguish between four agent system types with different classes of physical network connection (wired/wireless) and network reliability.

Current cellular mobile communication networks suffer under the limited bandwidth and high communication costs. Future systems will offer more bandwidth, but resources will still be restricted and compared to fixed networks additional efforts are necessary for a guarenteed quality of service. Thus, these systems should only be used for data transport to the fixed backbone communication systems, if the communication is really needed. This implies that disconnected operations have to be seriously taken into account for mobile telecommunications systems. However, non of todays agent technologies meet all of our requirements. In this paper we have outlined the most important requirements for a future, agent-based mobile telecommunication infrastructure. Moreover, these requirements have been classified to the aspects of terminal and personal mobility, as well as of a user friendly information society. For a suitable mobile agent system for cellular mobile systems a management concept, which meet all these different requirements for disconnected operations, is currently under development by the authors.

# References


[CAM00] CAMELEON - ACTS European research project 341. *Communication Agents for Mobility Enhancements in a Logical Environment of Open Networks.* http://www.comnets.de/~cameleon/, May 2000.

[CHK94] David Chess, Colin Harrison, Aaron Kershenbaum. *Mobile Agents: Are they a good idea?* Technical Report RC 19887, IBM, October 1994.

[HEG+99] Jens Hartmann, Rune Evensen, Carmelita Görg, Peyman Farjami, Hai Long. *Agent-based banking transaction and information retrieval - what about performance issues?* In *European Wireless'99 - Mobile Kommunikation, ITG-Fachbericht*, pp. 205–210. VDE, Berlin, Germany, October 1999.

[HGF98] Jens Hartmann, Carmelita Görg, Peyman Farjami. *Agent Technology for the UMTS VHE Concept.* In *First ACM International Workshop on Wireless Mobile Multimedia (WOWMOM'98)*, pp. 48–57. ACM SIGMOBILE, Dallas, Texas, USA, October 1998.

[IBM00] IBM Research Laboratory. *Aglets Workbench Homepage.* http://www.trl.ibm.co.jp/aglets/, Tokyo, Japan, May 2000.

[IKV99] IKV++ GmbH Informations- und Kommunikationssysteme. *Grasshopper Development System, Light Edition Release 1.2: Basics and Concepts*, February 1999. http://www.ikv.de.

[IPV00] University of Stuttgart IPVR. *Mole.* http://mole.informatik.uni-stuttgart.de/, May 2000.

[KRL+99] Ernoe Kovacs, Klaus Roehrle, Hong-Yong Lach, Bjoern Schiemann, Carsten Pils. *Agent-based Mobile Access to information Services.* In *4th ACTS Mobile Communications Summit*, pp. 97–102, June 1999.

[LPP99] Stefan Lipperts, Anthony S. Park, Carsten Pils. *Mobile Agents at the Boundary of*

*Fixed and Mobile Networks*. In *2nd International ACTS Workshop on Advanced Services in Fixed and Mobile Telecommunication Networks*, Singapore, November 1999.

[Obj00] ObjectSpace Inc. *Voyager Homepage*. http://www.objectspace.com/voyager/, May 2000.

[OMG98] OMG - Object Management Group. *Mobile Agent System Interoperability Facaility (MASIF)*, February 1998.

[WJ95] Mike Wooldridge, Nick R. Jennings. *Intelligent Agents: Theory and Practice*. The Knowledge Engineering Review, Vol. 10, No. 2, pp. 115–152, 1995.