Software Agent Technology in Mobile Service Environments[1]

Ralf Sesseler, Alexander Keiblinger, Nicolas Varone

DAI-Lab at Technische Universität Berlin
Sekretariat FR 6-7, Franklinstraße 28/29, 10587 Berlin, Germany
{sesseler, keiblin, nvarone}@dai-lab.de

Abstract. The demand for mobility in modern societies leads to the development of smaller and wireless networked computational devices. The resulting service environments for future mobile service networks will be complex. The individual character of future services requires highly flexible service infrastructures and development frameworks. We propose agent technology for the implementation of mobile services to reduce the complexity of mobile service infrastructures. We present JIAC IV, a Java-based agent framework designed with the service as a modelling abstraction. JIAC IV delivers a highly integrated set of tools for the analysis, design, and development of agent-based software services. Utilising the JIAC IV agent toolkit, a service developer focuses on application logic using agent services and the co-operation between agents. This will lead to faster service development cycles despite the increasing complexity of the environments. Encapsulation and communication between independent functional units can be a way to manage future complexity of service networks, and software agents are its technological implementation.

1 Introduction

The digital economy has been driven by two main phenomena: the tremendous success of networked services powered by the Internet protocol family and the need for mobility in modern societies, addressed by the equally successful mobile telephony. Mobile services (m-services) are positioned just at the crossroads of these two heterogeneous worlds. The next generation of electronic services does not only rely on the smart integration of technologies, it also needs identification and handling of a new array of arising business opportunities. Making the most of these opportunities will require new approaches and tools.

Agent technology [1] is a promising way to realise the basic infrastructure for the supply of m-services. Agents perform their tasks autonomously and flexibly by integrating reactive, goal-driven, and interactive behaviour. Agent societies are open and dynamic and may be distributed logically as well as physically. Thus, multi-agent systems provide the flexibility and openness required for m-commerce. Our approach extends agent technology by a service concept that enhances the interoperability of agents and maps services to the agent level. In addition, our agent toolkit JIAC IV

[1] This work was funded by T-Systems-Nova Deutsche Telekom Innovationsgeselschaft mbH.

(Java Intelligent Agent Componentware, Version IV) offers a comprehensive framework for the development and deployment of multi-agent systems.

In this paper, we consider the particularities of m-services compared to e-services and examine the requirements for m-services successively from a user and a provider point of view. We will introduce the JIAC IV agent framework and outline how its features meet m-service specific requirements. In more detail, we describe the mechanisms of JIAC IV that enable a high degree of flexibility and dynamics in providing and using services.

2 Scope and Definition of M-Services

A common and broad definition of m-services is the act of providing and consuming services through wireless handheld devices such as cellular phones, personal digital assistants (PDA), laptops, or any other wireless enabled device. Usually expected to be the next wave of electronic services, mobile services (m-services) enable users to access networked services anywhere and anytime. From all the numerous aspects characterising any e-service solution, such as availability, reliability, ubiquity, instantaneity, and usability, we have identified the latter two as the main properties differentiating between e- and m-services.

Usability describes the comfort in the utilisation of devices and services regarding size (device, keys, screen), multimedia capabilities (colour, sound, CPU), and user interfaces (input capabilities, menus). Usability is also affected by the access method (dial-in or always-on) and the network performance parameters. Handheld devices used to access m-services have to get along with a smaller set of resources in terms of memory, processing power, storage capacity, and display size in comparison to the devices used in e-services. Technology is pushing the limits in the field of stationary and mobile technology simultaneously, while bandwidth differences persist.

The second main aspect to consider is instantaneity, which indicates how quickly and efficiently a transaction can be performed. It also enables impulsive and unplanned user activities. Consider a car driver listening to his favourite song in the radio. At this moment, the user may have the desire to buy the CD of the artist. In a conventional e-commerce scenario, the user has to delay his transaction until he has access to a networked computer. In the upcoming m-services world, the user will be able to place an order directly, resulting in a higher factor of instantaneity.

Office computers are likely to have a broader bandwidth and to be continuously on-line, but suffer from restrictions on privacy depending on the employer's policy with respect to usage and system configuration. Home computers provide a higher degree of usability because the user has full administrative control. Public Internet access points have no or few advanced personalisation functions, but tend to become more and more widely available enabling a higher degree of instantaneity. Laptops and PDAs cope with these notions of privacy and personalisation, but have a narrower bandwidth due to air network access constraints and limited multimedia capabilities. Handhelds have even more restricted multimedia capabilities, but they offer a higher degree of portability and can be used instantly in many situations.

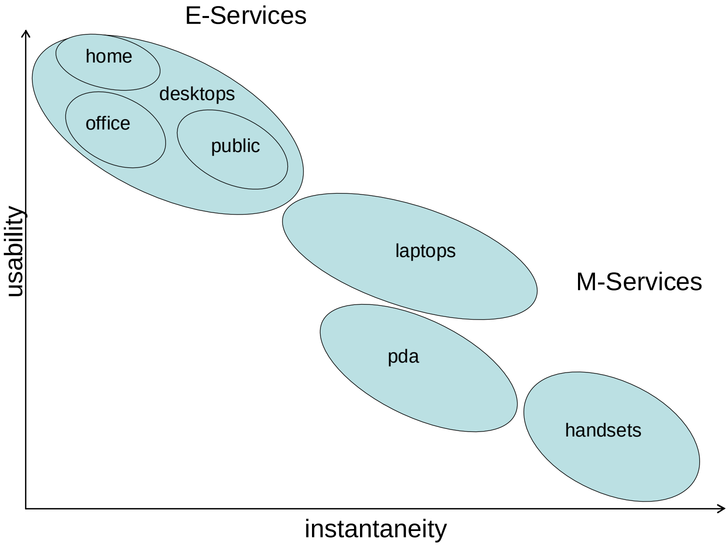

Figure 1. Relationship between usability and instantaneity

M-services are usually represented as a subset of all e-services, implying that any e-service could be made available from a wireless device. Instead, we see an inverse relationship between usability and instantaneity spanning a gradual transition from e-services to m-services (Figure 1). Therefore, m-services should be recognised as a unique business opportunity with its own unique characteristics and constraints and not just an extension of an organisation's Internet-based e-services channel.

2.1 M-Services: A User Perspective

A typical e-service usage takes more time than a typical m-service usage. Usually, an e-service user wants to compare competing offers and to switch between them. He processes much more information before proceeding. M-services fulfil a spontaneous need or desire, accomplishing a well-defined task in a well-defined context. The respective process and the user interface should be designed to enable a simple utilisation. Automated mechanisms that are easily accessible have to be provided to complete the task efficiently.

Certain similarities between e- and m-services, especially the following common issues should be addressed: security, personalisation, service aggregation, high availability, and network and terminal independence. Security is a fundamental challenge any m- or e-business project is faced with. It is a topic extensively documented in the literature with solutions implying the utilisation of standardised encryption, authentication, authorisation, and non-repudiation mechanisms. Personalisation mechanisms should drive and assist the user experience and save time in the navigation and input of data. New business models require the flexible assembling of services. The goal is to enable a smart collaboration between actors of the value chain and to lead the creation of transversal services. The aggregation of services will provide the user with a homogenous view on his m-commerce experience.

Another common constraint is availability. Systems have to guarantee high availability, be robust with respect to hardware and software failures, and also enable smooth maintenance operations such as on-the-fly reconfiguration or component replacement. Maintaining a high level of reliability will create trust and loyalty relationships between business partners. The user interfaces have to be flexible and adaptive enough to fit the targeted device. Applications have to be aware of the kind of terminal used and provide user interfaces and content accordingly. Due to the possibility of intermittent wireless network coverage, applications have to be able to work in an off-line mode and to make use of methods to restore the user context after an interruption.

2.2 M-Services: A Provider Perspective

The user experience with current Internet-based e-commerce will set the basic demand for the level of security, usability, personalisation, and availability of m-commerce services. The service providers in the mobile environment will have to catch up with these standards, but also integrate the location and situation awareness in a consistent manner. This awareness needs to be invisible to the end user or be easily accessible from the user interface of the end user device. The service providers need to integrate information regarding the location, price, availability, and content of services as selection criteria into the service infrastructure. These constraints result in a demand for context sensitivity and user adaptivity of mobile devices.

There are some fundamental commodities an m-service provider infrastructure needs to offer. The upcoming world of dynamically selected services needs support for short and overall savvy service development cycles to reduce costs.

Capabilities for dynamic maintenance, management, and extension of the infrastructure as well as of single services have to be supplied. Distributed accounting and billing mechanisms are needed which are easily scalable in a highly dynamic service environment. Special precautions have to be taken with respect to the security of services, accounting, and user data.

Additional data sets with user and community profile data and location data have to be merged into an integrated, user-adaptive data model. Given the high cost for content, infrastructure support for content exchange or recycling between different media channels is required. Acting on the assumption that complex services will become aggregations of sub-services from different providers, mechanisms for billing between service providers are needed. These have to be supported within the overall mobile network infrastructure.

Considering the technological and social challenges described earlier, we recommend agent technology as a well-suited software paradigm for the upcoming service environments. Agent technology has been designed for complex, distributed, and open environments. Decentralisation helps to cope with the complexity problem of service infrastructures. Software agents are encapsulated entities, which interact at a level of formal semantics. Mobile agents are able to travel between platforms to fulfil their tasks at different locations. The agents can reside within the provider network infrastructure or on the end user device. Stemming from decades of research in the field of artificial intelligence, agent technology has the additional benefit of operating at a higher level of abstraction than procedural or object-oriented techniques.

3 The JIAC IV Agent Toolkit

Before going into detail about some specific aspects of JIAC IV, we will give an overview of the whole toolkit in this section and motivate some of its features by their relation to the requirements for the realisation of e- and m-services we stated above.

JIAC IV is intended as a comprehensive toolkit for developing and deploying multi-agent systems. The metaphor of a service is used as an abstraction of interactions between agents as well as of interactions between the user and the system. It allows a service developer to focus on his business logic in terms of modelling the content of a service, of the processes of providing and using a service, and of the relationships between agents. The level of the operation of the services, the interactions between agents, and the end user access is realised by the capabilities of the agents, the functionality of the agent infrastructure, and the generic access scheme, which are part of the JIAC IV system. Already integrated are also mechanisms for security and management functionality. In addition, the toolkit delivers an integrated set of tools for developing agent-based software services. The implementation of the toolkit has reached a stable working state and will be publicly available soon.

The toolkit covers design methodology and tools, component and control architecture, agent languages, a FIPA compliant infrastructure, management and security functionality, and a generic scheme for user access (Table 1). For a detailed overview, see [2]. Though it is a generic agent toolkit, JIAC IV has many features suitable for the realisation of platforms for e- and m-services. These features are outlined in the following. We also refer to [3] regarding the relation between our toolkit and electronic marketplaces.

Table 1. The JIAC IV Agent Toolkit

JIAC IV Agent Toolkit					
Architecture		Development		Run-Time	
component architecture	*control architecture*	*methodology*	*tools*	*infrastructure*	*administration*
components	knowledge	analysis	configuration	platforms	configuration
roles	reactivity	design	compilers	migration	fault management
messages	deliberation	implementation	debugger	white pages	
	interaction	evaluation		yellow pages	end user access
				security	

A crucial aspect of e- and m-services is the interoperability in the context of an open and dynamic environment. The service scheme of JIAC IV allows interactions to be established and performed dynamically. Required information and support for interactions are provided by the agent infrastructure based on the specifications of FIPA [4]. To exploit the dynamics of the environment, the agents require the ability to act in a flexible and autonomous way. Therefore, the control architecture of agents in JIAC IV integrates reactive, deliberative, and interactive behaviour [5]. Reactivity allows dealing with opportunities and other suddenly changing circumstances that may occur. The deliberation process uses the set of available capabilities to flexibly perform the given tasks. The integration of interactivity includes the capabilities of other agents into these single-agent control processes. By combining dynamic selection of services and flexible control of agents, services may be provided, accessed, and assembled from more basic services automatically or in collaboration of the user and his agents. Thus, particular e- and m-services do not remain separated, but are integrated into larger interoperating service environments. This reduces the development effort and increases the benefit for the user.

The special requirements of commercial applications on the reliability, trustworthiness, and maintenance of the system are addressed by several management and security functionalities. Management services include configuration, fault management, and the logging of system processes, which may be used for analysis as well as for accounting. Security issues are handled by an integrated infrastructure supporting authorisation, authentication, and privacy mechanisms [6].

Additional functionalities for management, security, and others may be integrated and adapted easily due to the overall modularity of JIAC IV. At the agent level, the modularity is attained by a component system allowing the rapid creation and reconfiguration of agents, since they are assembled of reusable and exchangeable components and knowledge elements [5]. It is for example possible to add new communication transport facilities, to exchange the planning mechanism, or to replace a component by an improved version at run-time. At the society level, modularity is a result of the openness realised by the agent infrastructure. Thus, the implementation of a service may be modified and extended without affecting its permanent availability.

Finally, the process of developing and deploying complex systems is supported by an agent-oriented software engineering model. It is supported by tools for agent specification and tools to analyse and debug the running system. Furthermore, the toolkit includes a generic scheme for user interfaces, which translates agent functionality into services accessible by humans in multiple formats. Thus, JIAC IV allows a rapid development of complex, distributed systems for service provisioning.

4 Dynamic Interactions Based on Services

The main objective of the service concept of JIAC IV is to gain a high degree of flexibility and dynamics of interactions between agents. A crucial prerequisite of this is interoperability, which is achieved by standardisation at different levels allowing mutual understanding. For that purpose, all interactions are described as services in a uniform manner and are performed by a generic protocol. Apart from interoperability, this also enables the integration of interactions into the knowledge-based control scheme according to the service descriptions. The agent infrastructure facilitates communications and interactions between agents by providing information about agents and their services dynamically. Finally, the user access translates services from the agent level to the user level by mapping services and protocols onto user interfaces. The details of this service concept are the topic of the next two sections.

4.1 Interoperability by Standardisation

Standardisation is a common technique to attain interoperability between agents. A well-known example is the effort of FIPA [4] to provide a framework for the interoperability between different agent systems. But in open, dynamic, distributed systems, interoperability is already required within an agent system of a single architecture, because it lacks fixed, predefined boundaries and dependencies.

Since they use communications as the basis of interactions, most agent systems employ standards for the format, the content, and the course of communications. Agents in JIAC IV communicate by exchanging speech acts according to the FIPA-ACL format [4]. The content of these speech acts is expressed in the knowledge representation language of the agents using ontologies to define shared vocabularies. Protocols facilitate the processing of communicative acts by prescribing the possible courses of valid conversations in advance.

At the society level, interactions between agents are facilitated by an agent infrastructure. Again, this infrastructure has to rely on standardisation to ensure access to its functionality for all agents. Therefore, the agent infrastructure of JIAC IV utilises the corresponding FIPA management specifications [4].

To bridge the gap between the agent level and the society level, the service concept of JIAC IV introduces an additional layer of standardisation for describing and performing interactions. From the society point of view, services specify interdependencies between abstract roles of agents in an open manner. From the point of view of a single agent, services allow flexible access to additional capabilities and resources provided by the society. Thus, services standardise interactions in a similar way as speech acts and protocols do for communications and as the infrastructure does for agent societies.

4.2 Integrating Agent Control and Interactions by Services

As in many agent systems, the behaviour control mechanisms of JIAC IV [5] are based on well-known knowledge-based techniques in the style of BDI [7]. Reactivity is ensured by a rule-based situation assessment and goal-oriented behaviour is realised by planning mechanisms. The usage of formal logics for knowledge representation allows the description and the control of agent behaviour at an abstract level in a uniform manner.

The first step to integrate interactions into the behaviour control mechanisms of the agents is to describe interactions in the same way single-agent actions are represented. Agents that rely on AI planning techniques usually describe actions in the form of operators stating the conditions and effects of their execution to allow a goal-directed behaviour control. In JIAC IV, interactions are therefore also expressed by operators, while the execution of these operators is delegated to other agents. These operators are called services because they express an action that the provider of the service performs in order to reach an effect for the customer. Therefore, the service operators have to describe the interactions from the point of view of the customer who uses them in its decision process. The service provider handles services in a more reactive manner, since it is permanently waiting for requests.

To represent interactions as operators has many advantages. Conditions and effects of the usage of a service are described explicitly in a familiar way. Interactions may be selected by the agents dynamically as any other action, and the combination of services or their integration into sequences of actions exploits the planning capabilities of the agent. Thus, service operators allow flexible and automated interactions between agents.

5 An Open, Generic Protocol for Service Access

A service operator only describes the circumstances of an interaction, but not how to perform the interaction. The latter part is done by conversation protocols. Since all interactions between agents in JIAC IV exploit services, they may be performed by a common scheme, which is prescribed by the service meta-protocol with the provider and the customer as roles.

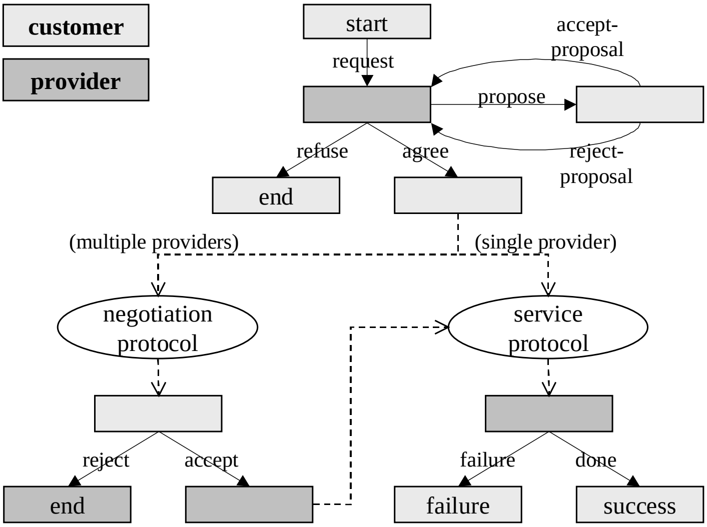

Figure 2. The Service Meta-Protocol

The meta-protocol of JIAC IV (Figure 2) for accessing services consists of three phases. The customer initiates the protocol by requesting the service from one or several providers. After negotiation of generic parameters, the provider agrees or refuses the request. Then, the customer chooses the best provider based on service-specific negotiations. Finally, this provider supplies the service. To retain openness, negotiation and service supply are handled by embedded protocols that may be freely defined for a service. For negotiation, common auction protocols may be used.

The meta-protocol is tightly bound to the concept of a service operator. On execution of such an operator, an agent initiates the meta-protocol as the customer and hands over to the provider the instantiation of the operator as given by its conditions. The result sent back by the provider contains the achieved effect of the operator. Thus, the meta-protocol implements the process of delegating the execution of a service operator from the customer to the provider.

Operators and protocols for services provide the agent only with the ability, but not the opportunity to interact. The latter is the purpose of the agent infrastructure, which constitutes a society of agents and facilitates interactions. As mentioned before, the agent infrastructure of JIAC IV draws from concepts of the FIPA management specification [4]. Agent platforms are the organisational unit that constitute societies of agents. Each platform has an Agent Management System (AMS) to administer its agents, a Message Transport System (MTS) as the default communication channel, and one or several directory facilitators (DF) to manage available services. Several platforms may be connected by mutual acquaintance.

The combination of service operators, meta-protocol, and agent infrastructure allows highly dynamic interactions between agents. The DF provides up-to-date information about available services and providers. It may be accessed on demand and in a goal-driven manner due to the operator character of services. In addition, the customer may choose a provider in the selection phase of the meta-protocol. Thus, the service concept enables a dynamic access to available services and providers as well as a dynamic selection of appropriate services and providers. Agents share their capabilities and may utilise the total functionality of an agent society to fulfil their tasks. The only prerequisite is the ability to use the required ontologies and embedded protocols that have to be standardised in advance.

Another advantage of modelling interactions as services is the possibility to map services from the human level to the agent level and vice versa. In the former direction, service parameters like costs, quality of service, and security requirements can be integrated into the service descriptions and used as selection criteria. Functionality such as accounting and billing may be supported by additional infrastructure services.

In the latter direction, services at the agent level may be made accessible to humans by translating the user role of the protocols to a user interface. The JIAC IV toolkit encloses a generic scheme for these translations based on intermediary agents.

By separating the user interface and the service functionality in that way, some specific restrictions of m-services may be addressed. The user interface can be adapted to the capabilities of the device appropriately. The limited resources of the mobile device are only used to run the interface and to handle data exchange to the agent in the network.

6 Summary

We identified usability and instantaneity as the main aspects to be considered for m-service solutions and found an inverse relationship between e-services and m-services that form a gradual transition. Therefore, it is not sufficient to simply extend the concepts and business models of e-services into the m-services world. M-services have to be recognised as technically challenging but also as interesting new business opportunities with unique characteristics and constraints. Aspects of services like security, personalisation, service aggregation, availability, and network/terminal independence have to be considered in a new way. The individual character of future services demands for highly flexible service infrastructures and development frameworks.

Designed with complex service environments in mind and the service as a central concept, the JIAC IV agent framework constitutes a toolkit for developing and deploying multi-agent systems covering design methodology and tools, agent languages, component and control architecture, and an infrastructure with integrated management and security functionality.

We described the service concept that bridges the single agent level and the multi agent level by providing a standardised but open way to describe and perform interactions. At the single agent level, the integration of interactions into the behaviour control allows the dynamic selection and combination of services. The service concept of JIAC IV enables the agents to profit from the openness and dynamics of the service supply. The usage of agent technology for the realisation of e- and m-services allows the transition from the static, stand-alone services of today to the dynamic, interconnected service environments of tomorrow.

7 Related Work

BDI-style agent architectures are based on the work of Bratman [8]. Like other BDI architectures, JIAC IV draws from the conceptual framework of BDI in a pragmatic way without implementing a belief-desire-intention logic. The modular structure of JIAC IV is comparable to component-based architectures (e.g. [9]). The use of toolkits in order to support the development process of agent-based systems is widespread (e.g. ZEUS [10], AgentBuilder [11]).

The basic idea of software agents as mediators has been described by Guttman et al. [12] for the domain of e-commerce. We focused mainly on the user adaptivity as the main difference to e-services. The model of [12] consists of six steps: Need identification, product brokering, merchant brokering, negotiation, purchase and delivery, and finally product service and evaluation. They argue that agents are of benefit mainly for product and merchant brokering and for negotiation because of their autonomy.

Our view of mobile services has been inspired by the European perspective of current mobile GSM services and the upcoming UMTS services as represented in [13].

Although security issues are commonly regarded as very important in the context of mobile agents, architectures with enhanced security features are sparse. Mole [14], another mobile agent platform, deals mainly with life-cycle aspects, localisation, and transactions. Access control lists are a well-known feature of Aglets [15]. In the Ajanta framework [16] security is managed similar to our approach by use of credentials and authentication protocols. Additionally, it provides a proxy-based mechanism for protected resource access.

References

1. Wooldridge, M., Jennings, N.R.: Intelligent Agents: Theory and Practice. Knowledge Engineering Review, 10 (1995) 115-152
2. Fricke, S., Bsufka, K., Keiser, J., Schmidt, T., Sesseler, R., Albayrak, S.: Agent-based Telematic Services and Telecom Applications. Communications of the ACM 44, 4 (2001) 43-48
3. Sesseler, R., Albayrak, S.: Agent-based Marketplaces for Electronic Commerce. International Conference on Artificial Intelligence, Las Vegas (2001)
4. FIPA 2000 Specification. www.fipa.org
5. Sesseler, R., Albayrak, S.: JIAC IV - An Open, Scalable Agent Architecture for Telecommunications Applications. First International NAISO Congress on Autonomous Intelligent Systems, Geelong, Australia (2002)
6. Bsufka, K., Holst, S., Schmidt, T.: Realization of an Agent-based Certificate Authority and Key Distribution Center. IATA'99, Lecture Notes in Artificial Intelligence (LNAI) Vol. 1699. Springer-Verlag, Berlin (1999)
7. Rao, A.S., Georgeff, M.P.: BDI-agents: From theory to practice. First International Conference on Multi-Agent Systems (ICMAS-95), San Francisco (1995)
8. Bratman, M.E.: Intentions, Plans, and Practical Reason. Harvard University Press, Cambridge, MA (1987)
9. Hayes-Roth, B.: An Architecture for Adaptive Intelligent Systems. Artificial Intelligence 72 (1995) 329-365
10. Azarmi, N., Thompson, N.: ZEUS: A Toolkit for Building Multi-Agent Systems. Proceedings of Fifth Annual Embracing Complexity Conference, Paris (2000)
11. Reticular Systems: AgentBuilder - An Integrated Toolkit for Constructing Intelligent Software Agents. Reticular Systems, San Diego, CA (1999)
12. Guttman, R.H., Moukas, A.G., Maes, P.: Agents as Mediators in Electronic Commerce. EM - Electronic Markets, Vol. 8, No. 1 (1998)
13. IZT, SFZ, IAT: Entwicklung und zukünftige Bedeutung mobiler Multimediadienste. IZT WerkstattBericht Nr. 49, Berlin (2001)
14. mole.informatik.uni-stuttgart.de
15. Karjoth, G., Lange, D., Oshima, M.: A Security Model for Aglets. IEEE Internet Computing 1, 4 (1997) 68-77
16. Karnik, N., Tripathi, A.: Security in the Ajanta Mobile Agent System. Technical Report, Department of Computer Science, University of Minnesota (1999)