Exploration Programmes:
Corporate Technology Explores Future Telecommunications

Intelligent Software Agents: The Next IT Revolution

The incredible growth of the demand for Internet data services has led to the introduction of new network communication devices allowing large bandwidth capacity and facilities for mobile services. Managing such components turns out to be a complicated task because of the inherently distributed nature of the applications and because of the emergence of new business models which allow service providers to deal with several network operators by using negotiations based on Service Level Agreements. Software agents can perform these negotiations in a fully automatic way, taking into account customer profiles. This is supported by encoding languages such as XML, which make the usage of small messages that contain knowledge information very efficient.

The Programme "Network Architectures & Technologies" explores the emerging IP functionalities supporting fixed and mobile services, and identifies key solutions to engineer and plan next generation packet-based networks. The objective is to achieve cost reduction in network investment and operation and to transfer new network capabilities into revenue generating converged network services.
With its Exploration Programmes, Corporate Technology is exploring telecommunication technologies and new service possibilities with a long-term view of 2–5 years. Further, the expertise built up in the course of this activity enables active support of business innovation projects.

Software agents will play an important role for the control of UMTS, MVNO (Mobile Virtual Network Operator) and future intelligent optical networks. They will also enhance peer-to-peer Web-based applications such as Napster or Gnutella [1] with service personalisation, information filtering, mobile services and a lot of helpful customer-centric services.

DANIEL ROSSIER

Intelligent agents technology results from the fusion of software engineering and distributed artificial intelligence mechanisms which are particularly adequate to deal with our heterogeneous, dynamic environment. It is regarded as a natural extension of object-oriented technology and relies on the two main fields of computer science research mentioned above. Internet networks and Web-oriented technologies like Java have now made the implementation of large-scale experimental agent-based systems possible. An overview of software agent technology can be found in [2]. A variety of software agents is depicted in fig. 1.

An Intelligent Agent is a Software Entity

An intelligent agent is a software component which exhibits specific characteristics such as *autonomy* – having its own execution context (code + data), *reactivity* – capability to react to notifications –, *pro-activity* – capability to make a decision based upon its own internal knowledge – and *social ability* – capability to exchange knowledge with other agents by using specific communication channels (ACC) and a specific language (ACL).
An agent-based solution therefore consists in designing and implementing several agents which co-operate in order to reach their internal goals. The concept of intelligence means that the agent is provided with knowledge of the user's wishes and makes use of this knowledge; it also refers to the intelligent behaviour emerging from the agents societies. Developing multi-agent systems requires a bottom-up approach. This has an advantage: multi-agent systems can solve problems that have not been envisaged during system development. Consequently, they are not developed for a specific task, but are designed for the general solution of problems.

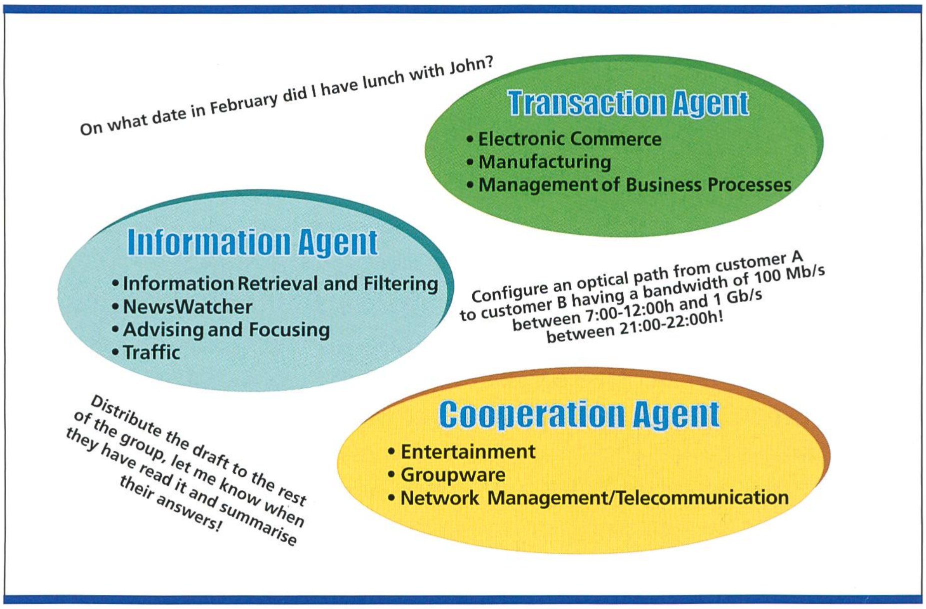

Fig. 1. A variety of Software Agents.

An Agent might decide to move itself

Thanks to mobile code techniques, agents are able to move from one computer to another. Mobile agents bring about a number of benefits. In particular, they *lower the communication costs* in a significant manner and solve tasks locally in a more flexible and elegant way, thus *reducing bandwidth needs.* In a network management application, for example, a mobile agent can be launched by a manager within the network, visit node after node, collect relevant information, or perform a task, and then go back to the manager.
Although research in the field of mobile agents is still at an early stage, mobile agents already promote bio-inspired approaches like ant colony based optimisation (ACO), that sometimes outperforms traditional heuristics [3].

The Foundation for Intelligent Physical Agents (FIPA) established in 1996 in Switzerland to produce software standards for agents and agent-based systems facilitates the inter-working of agents and agent systems across platforms issued by multiple vendors. The major telecommunication actors are members of FIPA (Lucent Technologies, BT [4], France Telecom, Nortel Networks, etc.) – see http://www.fipa.org
Object Management Group (OMG) has recently decided to create the Agent Platform Special Interest Group which deals with multi-agent system design evolution (see http://www.objs.com/agent/index.html). The Mobile Agents Service Interoperability Facility (MASIF) allows the inter-working of mobile agent platforms.

How do Agents communicate? Coordination and Cooperation

The ability to exchange complex knowledge constitutes one of the major strengths of multi-agent systems. Since

FIPA Specifications
http://www.fipa.org

Nortel FIPA-OS agent platform
http://sourceforge.net/projects/fipa-os

Mobile agent platform
http://www.grasshopper.de

Europe's IST-funded Network of Excellence for agent-based computing
http://www.agentlink.org

Agent Society
http://www.agent.org

Cluster for Intelligent Mobile Agents for Telecommunication Environments
http://www.fokus.gmd.de/research/cc/ecco/climate/

Nortel OpenetLab
http://www.openetlab.org

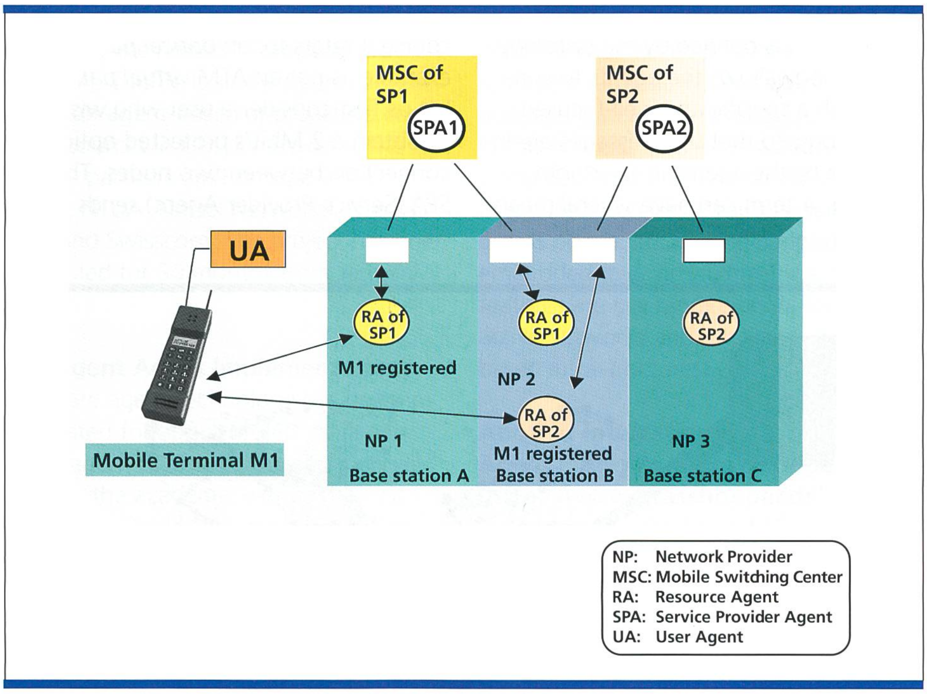

Fig. 2. Future 3G networks will introduce new interactions between SP, NP and the customers.

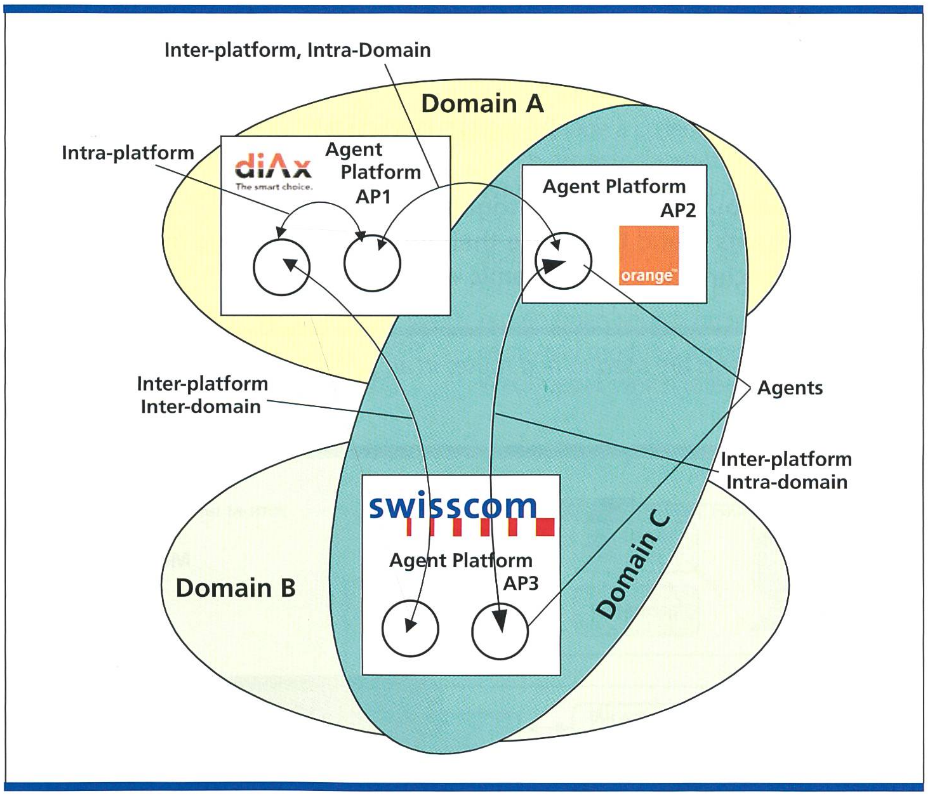

Fig. 3. Software Agents for multi-operators interactions.

the operations are performed in a distributed way, the support of cooperation and coordination between agents represents a necessary challenge for every agent platform. As for the 3G/4G mobile networks, typically, the number of entities involved in a transaction can be expected to increase. The customer will be able to choose the best service provider, which in turn will initiate negotiations with several network operators (fig. 2). From the interoperability viewpoint, standards in agent technology consequently constitute a key issue; a multi-agent system can be regarded as powerful as long as agents in different domains, i.e. from different network operators or different transport layers, are able to communicate *without ambiguity* and in a *secure way*. Fig. 3 shows the general architecture of inter/intra-domain communication.

Mobile agent-based systems do not always require direct communication: their mobility allows them to change the environment and to influence other agents through these changes. Agents are also able to exchange knowledge as they meet during their travel. In that case, mobile agents resort to advanced coordination mechanisms in order to upgrade their internal knowledge by picking up knowledge from other agents. Routing in an ad-hoc network or peer-to-peer virtual network (fig. 4) is an example of co-ordinated mobile agent-based application.

Agents need a clear Language: Communicative Acts & Ontology

The mechanism of communication between agents relies on the speech act theory. Readers interested in speech act theory can refer to [5]. The communication principle consists in defining a collection of highly structured messages initiating concrete action in the receiving agent and then changing its mental state. Examples of messages are request, *agree, inform, refuse, query_if*. The message content can be described with SL0 or RDF, two possible content languages, and is then encoded with XML to produce small messages. The collection of terms and rules embedded in the message content is defined by the *ontology*. The ontology allows the agents to converse with a specific and well-focused terminology, so that the messages are interpreted by the agents in their right meaning. A term can have several meanings depending on the domain of discourse it refers to: an *optical path,* for example, is not an ATM *virtual path.* Let us first consider a user who wishes to obtain a 2 Mbit/s protected optical connection between two nodes. The SPA (Service Provider Agent) sends a *request* message to the RA (Resource Agent) which replies by *agree* or *refuse* according to the wavelength availability. In case of agreement, the message informs the client of the discovered path (fig. 5). It has to be noted that ontology can be stored in a remote host. Suppliers could use common standard-based ontologies issued by an Internet site in the future.

The following example is based on a meeting scheduling scenario (fig. 6): the coordinator agent is expected to schedule a meeting suitable for all attendees. Having been activated by a human user, the coordinator agent proceeds to issue a call for proposals to the invitees. According to their availability, the attendees reply by a *propose* or *refuse* message. The coordinator looks at each incoming message and works out the best time to hold the meeting, sending the *accept-proposal* or *reject-proposal* message. Each invitee which can attend the final meeting then responds with *inform* after it has scheduled the meeting details in its calendar.

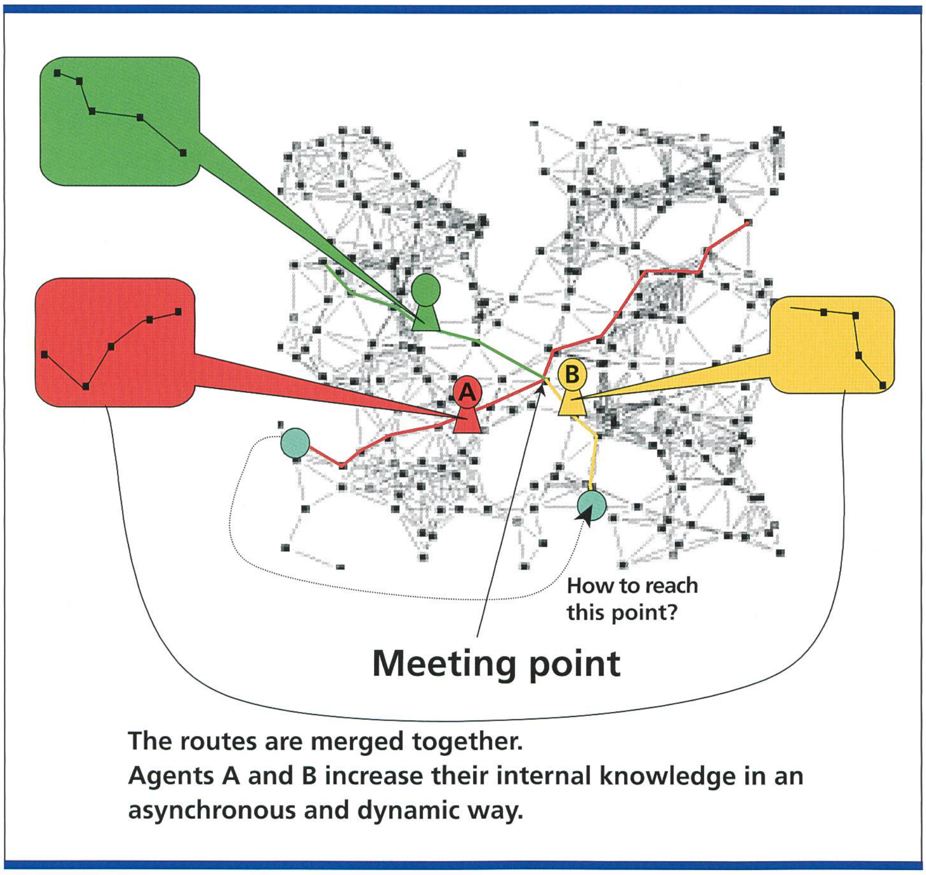

Fig. 4. Mobile Agents are used to find routes in an ad-hoc network.

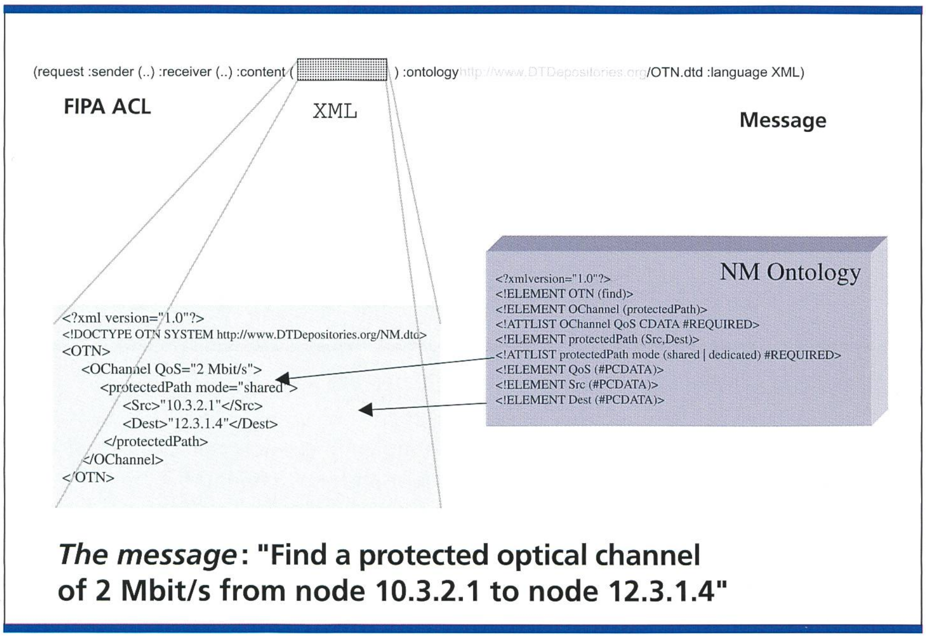

Fig. 5. Example of a FIPA message using XML as encoding language.

References

[1] "The Impact of Napster on Swisscom", Christoph Rytz, CT Outpost USA, monthly report, October 2000

[2] Walter Brenner, Rüdiger Zarnekow, Hartmut Wittig, "Intelligent Software Agents", Springer-Verlag Berlin Heidelberg 1998

[3] Alex L.G. Hayzelden, John Bigham, "Software Agents for Future Communication Systems", Springer-Verlag Berlin Heidelberg 1999

[4] http://www.labs.bt.com/projects/agents.htm

[5] Austin, J.L., "How to do things with words", Harward University Press, Cambridge 1962

[6] Open source community – http://www.opensource.org

[7] European R&D Framework Program ACTS – Advanced Communications Technologies and Services IST – Information Society Technologies

Abbreviations

AMS	Agent Management System
ACC	Agent Communication Channel
ACL	Agent Communication Language
CORBA	Common Object Request Broker Architecture
DF	Directory Facilitator
FIPA	Foundation for Physical Intelligent Agent
IIOP	Internet Inter-ORB Protocol
IST	Information Society Technologies
MA	Mobile Agent
MASIF	Mobile Agents Service Interoperability Facility
MVNO	Mobile Virtual Network Operator
OMG	Object Management Group
OO	Object-Oriented technology
ORB	Object Request Broker
RA	Resource Agent
RDF	Resource Description Framework
RMI	Remote Method Invocation
SL0	Semantic Language
SPA	Service Provider Agent
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunications System
XML	eXtensible Markup Language

Agent Activities at Corporate Technology

In the past, Corporate Technology has developed a Java-based agent framework in the scope of IMPACT, a European ACTS project [7] involving ASPA, Flextel (Italia), National Technical University of Athens, Queen Mary & Westfield College (QMW), Teltec (Ireland), Tele Danmark and Swisscom. IMPACT aimed at studying an agent-based architecture to allocate bandwidth in an ATM network. In this context, intelligent agents were used to perform market-oriented negotiations to take advantage of the best offer among various virtual providers. A description of this approach is presented in the article "Bandbreite auf dem Markt erhältlich", Comtec 11/99.

Swisscom is now involved in Shuffle, a project of the new IST European program [7]. Shuffle is studying and developing an agent-based approach to control resources in UMTS networks. The Shuffle partners are Portugal Telecom, QMW, NTUA, Nortel Networks, Martel (CH) and Swisscom. The project has been scheduled for 30 months from January 2000.

Intelligent Agent Implementations

Software agent technology has been investigated for about fifteen years and many agent platforms have been developed in the academic world. The first publicly available FIPA implementation, FIPA-OS, came from Nortel Networks and was released as *open source* in October 1999. Quotation: "The basic idea behind open source is very simple. When programmers on the Internet can read, redistribute, and modify the source for a piece of software, it evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing" [6].

With the help of the tutorials and agent templates which are part of the distribution, intelligent agents can be developed rapidly. The core architecture is made of three agents – AMS, DF and ACC (fig. 7). The communication between two platforms relies on traditional distributed systems technologies such as CORBA/IIOP, RMI, VoyagerORB, or on sockets only.

The DF Agent is basically a "yellow page" service which provides facility to retrieve information about the running agents registered on a host.

In this context, Nortel Networks are currently developing a new family of products called Accelar (IP routers, wireless access points, etc.) which support an embedded Java virtual machine. Such features are key requirements for future active networks and intelligent agent-based infrastructures.

Another interesting Agent Platform is Grasshopper (http://www.grasshopper.de).

Grasshopper is the first OMG MASIF and FIPA compliant platform. The MASIF architecture is different from FIPA and consists of hierarchical entities able to host and manage mobile agents (fig. 8). An agent can move from one place to another. The place is managed by an agency which provides services for migration, localisation, registration, etc. The agency in turn belongs to a region. In the case of UMTS networks for example, the region delimitates a set of interacting service providers. A service provider agency can manage a collection of services which are associated to specific places. When service roaming is needed, a mobile agent can then negotiate the best agreements between service providers. Grasshopper is a free product – but is not an open source project – composed of several modules among which are Webhopper which integrates the mobile

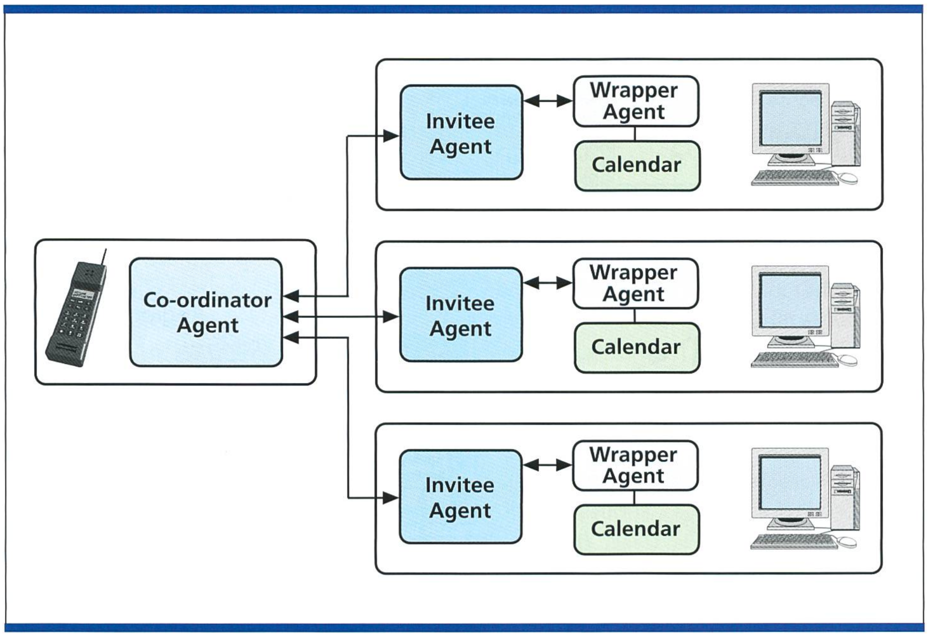

Fig. 6. Meeting scheduling scenario with a Coordinator Agent.

agents technology and World Wide Web using the *servlet* technology. Grasshopper is also available for *Windows CE Pocket PC.*

Conclusions

The future telecommunication landscape will encompass the development of sophisticated mobile devices, the creation of new Web-based services and the deployment of terabit optical networks. Managing future services and networks will obviously be a complicated task.

Although intelligent agent technology is promising indeed, there are still a number of unresolved crucial issues. First of all, a widely adopted definition of an intelligent agent still remains to be determined. The term "agent" itself has been used in many areas. Standards like FIPA or OMG are constantly involved in the specification of concepts based on intelligent and mobile agents (architecture, language, ontologies, interaction protocols) and should facilitate their integration into the telecommunication world.

Moreover, ongoing research activities are currently concentrating on security aspects, and solutions begin to appear.

Agent technology also suffers from the absence of methodology. OMG has concentrated its efforts on object-oriented technology and supports UML as modelling language. Based on observations made in both the OO and agent world, OMG has decided to create an agent working group in order to examine the convergence of these two technologies.

A working group is presently focusing on AUML (Agent UML), the adaptation of UML to support the modelling of agents and their interactions.

From the point of view of infrastructure, Corba, Java, Jini, or Enterprise JavaBeans and the World Wide Web provide excellent means of implementing intelligent and mobile agents. Despite performance problems which will be solved gradually thanks to new releases of the Java virtual machine and J2ME (Micro Edition), the development of new multi-agent systems will take a more serious turn in the coming years. The number of collaborative communities in agent technology – AgentCities, NIST, FIPA-NET – is continuously growing. With Internet, it is now becoming easier to deploy a multitude of agents with different purposes and to study their interactions.

Ongoing research work in the field of mobile computing, evolutionary computing, active networking and bio-inspired algorithms is now bringing about innovative agent-based solutions. The success of these approaches will strongly depend on the capability of future communication devices to embed software agents.

In conclusion, from a network and service provider viewpoint, it is crucial to consider the opportunities to use intelligent and mobile agents *now*.

Outlook

In the Shuffle project, Swisscom is interested in deploying a small IP-based network using Accelar products from Nortel Networks with an embedded Java virtual

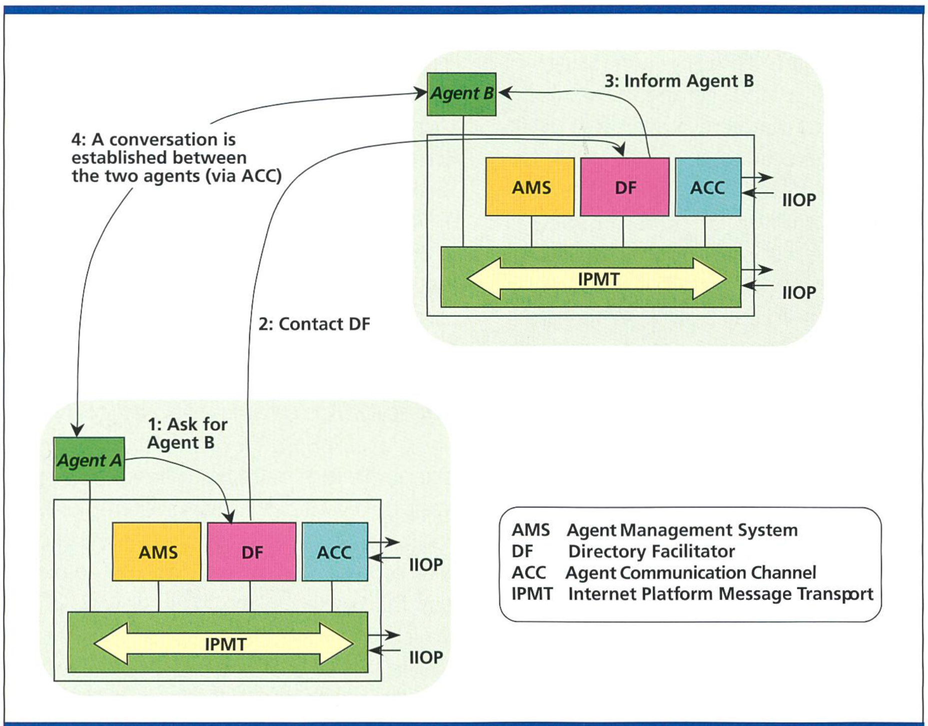

Fig. 7. FIPA architecture for an Agent Platform.

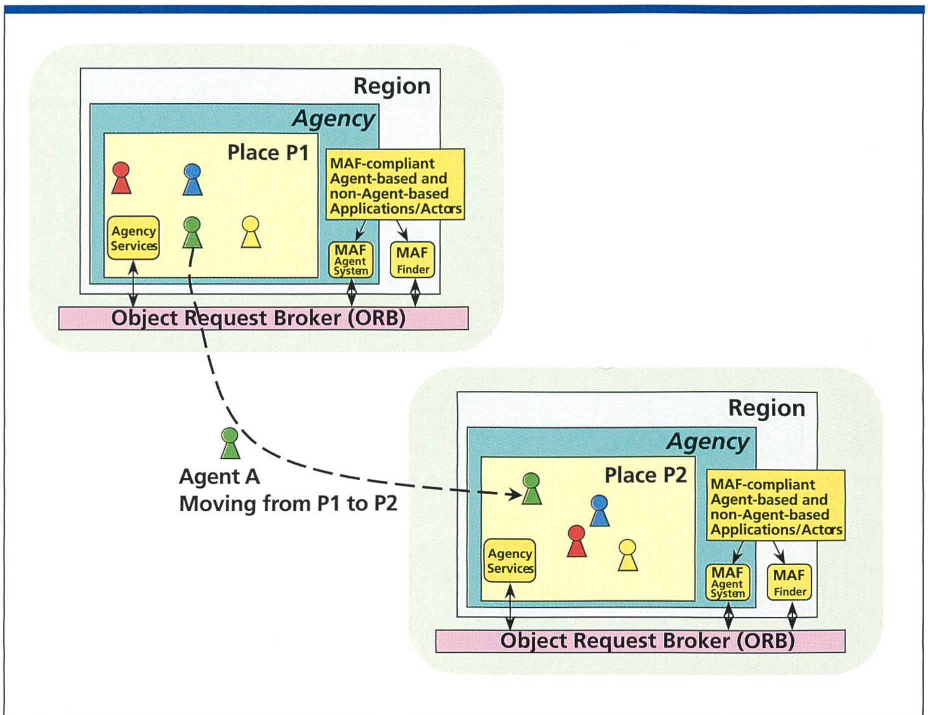

Fig. 8. OMG MASIF architecture for a Mobile Agent Platform.

machine. It is therefore possible to install FIPA-OS in the router itself. We will then implement our agents and let the network run as if it were a UMTS network. Experiments will start in January 2001 and the first results are expected by the middle of that year.

One of the main requirements of the 5th European framework program is the elaboration of dissemination plans to ensure the transfer of our results to business units.

We have also proposed to start an "Intelligent Agent Technology" project aiming at the study of opportunities to develop agent-based solutions for Web-based applications, E-business, security management in firewalls and network management within the scope of a future exploration programme. 8.3

Zusammenfassung

Die zunehmende Zahl von internetbasierten Applikationen und Mobildiensten verlangt nach einer Infrastruktur, die in der Lage ist, all die Interaktionen einer zukünftigen Telekommunikationslandschaft zu steuern. Intelligenten Softwareagenten kommt dabei eine Schlüsselrolle zu. Dank ihrer spezifischen Eigenschaften wie Autonomie und Gesellschaftsfähigkeit (Sociability) können Agenten komplexe Informationen zur Erfüllung ihrer Aufgaben austauschen. Die Einführung von Softwareagenten in Kommunikationseinrichtungen wird neue Dienste mit intelligenter Verarbeitung von Informationen und persönlichen Daten ermöglichen.

Daniel Rossier *completed his studies at the EPFL and acquired a computer science engineer degree after spending four years working as a software developer for hierarchical and relational database applications. After his studies, he worked during 2½ years as a software engineer in the field of digital television, conditional access system and network management. Having started to work at Swisscom in October 1998, he has undertaken a PhD in collaboration with the University of Fribourg in the field of intelligent and mobile agents-based full optical network management. He is currently involved in the IST-Shuffle European project in the scope of the Exploration Programme "Network Architectures & Technologies".*

Richard W. Bloch verstorben

Bloch? Nie etwas von ihm gehört? Vielleicht doch: Wer immer in der Vergangenheit mit Datenübertragung zu tun gehabt hat, ist an Bloch nicht vorbei gekommen. Bloch war zuletzt der Chef der Rechenzentren an der renommierten Harvard Universität in Cambridge, Massachusetts. Er hatte dort auch studiert und war später Vice President unter anderem bei den Unternehmen Honeywell und General Electric. Bekannt geworden ist er durch eine Standardprozedur zur automatischen Fehlererkennung bei der Übertragung von Daten sowie deren Ein- und Ausgabe: Den so genannten Parity Check, der bis heute in praktisch allen Rechnern zu finden ist. Für diese Paritätsprüfung werden den zu übertragenen Zeichen Prüfbits hinzugesetzt, sodass die Summe aller «0» oder «1» innerhalb des Datenblocks eine gerade Zahl (oder auch ungerade Zahl, je nach Vereinbarung) ist. Stellt sich bei der Überprüfung fest, dass dies nicht der Fall ist, muss ein Fehler vorliegen. Durch entsprechende Verfahren kann der Fehler auch automatisch korrigiert werden. Bloch hielt das Patent auf diese Erfindung. Er starb im Ruhestand im Alter von 78 Jahren.

Ein Roboter macht Riesensprünge

Roboter stellt man sich ja eher als schwerfällige und langsame Objekte vor, wenn sie sich frei durch den Raum bewegen. Treppensteigen ist dabei beispielsweise eine schwierige Art der Fortbewegung, weil das Gleichgewichtsproblem zusätzliche Anforderungen stellt. In so weit überrascht eine Entwicklung am Sandia National Laboratory, von der die US-Zeitschrift «EE Times» berichtet. In Albuquerque, NM, wurde ein Roboter gebaut, der wie ein Grashüpfer über Hindernisse hinwegspringt. Auch das könnte man noch zur Kenntnis nehmen, wenn nicht die Sprunghöhe spektakulär wäre: Der Roboter überwindet bis 9 m Höhe. Dann aber geht ihm die Puste aus. Die Energie für die Brennkammer bezieht der Roboter aus einem kleinen Gastank mit nur 20 g Treibstoff. Diese Energie kann man natürlich auch besser verteilen. Auf 4000 Hüpfer zum Beispiel, jeder Sprung 1 m hoch und 1,8 m weit. 100 Hüpfer kann der Roboter absolvieren, wenn er dabei 6 m hoch springen soll. Der einfüssige Roboter hat einen eingebauten Mikrocomputer, der den Roboter nach dem Sprung wieder sicher auf die eigenen Füsse stellt. Wer braucht so etwas? Das Militär natürlich, denn das Pentagon will damit eines Tages Geländeerkundungen durchführen.

Sandia National Laboratory
Public Relations
P. B. 5800
Albuquerque
NM 87185
USA
E-Mail: http://www.sandia.gov