UDC 621.397.2:681.3.06

An Agent-based Service Brokering Architecture for Multiservice Next Generation Networks

●Francisco Navarro ●Keith Jones ●Sagar Gordhan ●Nigel Garnham
(Manuscript received February 6, 2001)

This paper describes how advanced distributed software techniques based on agent technology can be used to provide brokering within a complex service delivery environment in order to deliver services to end users in a consistent and simple manner. It also describes the benefits of separating and distributing service control from service transport in next generation networks (NGN), a principle inspired by the MSF architecture. An agent-based multiservice platform, called *Agent Enhanced Service Portal (AESP)*, has been designed and a prototype implemented to control broadband service sessions in Fujitsu's NGN testbed. The prototype has been implemented using a web computing platform enhanced with agent technology called Phoenix. It consists of five service providers offering high speed Internet access, Voice over IP (VoIP), Video on Demand (VoD), streamed video and games services delivered over a broadband core and access network. A description of this novel platform and its related technologies is presented in this paper, highlighting the suitability of agent technology to provide brokering facilities and distributed control in NGN.

1. Introduction

In recent times the telecommunications environment has seen both a rapid increase in the number of services on offer and the number of network, service and content providers that are involved in the chain of supply of these services. The trend is set to continue given the emergence of next generation networks (NGN) able to deliver voice, data, broadcast and broadband interactive services. This has resulted in a complex service delivery environment and a requirement for mediation has emerged in order to support the scaleable and efficient deployment of services in this environment. Trading and brokering facilities are required that 1) manage the complex interactions between the various stakeholders that are involved in the service supply chain and 2) provide the end user with a simple and consistent facility for selecting, subscribing and connecting to these services.

Agent technology is one of the most interesting and fastest growing areas within the distributed software community. However, the definition of the term 'agent' is controversial. Our use of the term here refers to a relatively simple autonomous system component that exists in a community with other agents and co-operates with those other agents in order to achieve complex objectives. This approach results in flexible and adaptable 'plug and play' systems that are inherently scaleable, characteristics which are highly desirable in telecommunications systems. These are the benefits that we aim for rather than the more rarefied claims of intelligent behaviour.

The agent-based approach is hence a promising candidate for implementing the required trading and brokering facilities among service providers, network providers and end-users. In this paper, we describe how the various stakeholders and relationships involved in the chain of supply of services as shown in the TINA-C trading model,[1] can be mapped to a multi-agent architecture based on FIPA specifications.[2] The proposed architecture uses a community of distributed agents which represent each of the service providers, network providers and end-users. These agents advertise information about their associated service offerings, network spot capacity and user location respectively to an agent representative of a Service Portal, which acts as a broker between the various stakeholder agents. The Service Portal is then the gateway to the myriad of next generation services, since its agent has a global view of service and network availability and enables users to select, subscribe and activate all the available services in the system.

A prototype of this agent-based service brokering architecture, called the *Agent Enhanced Service Portal (AESP)*, has been implemented using the Fujitsu's Phoenix[note 1)] platform, a Java™-based web computing middleware enhanced with an agent-based library to provide federation and high-level communication between distributed Phoenix platforms.

This paper initially provides a description of the motivation, features and main benefits of an agent-based approach to provide brokering within a complex service delivery environment. The next section then presents the AESP prototype to verify the application and feasibility of the agent-based service brokering architecture. It first introduces a brief description of Fujitsu's Phoenix platform on which AESP is based. It then presents the system design and main features of the AESP, completing its description with the proof of concept demonstration testbed. Finally, we summarise the key advantages of AESP and point out the main conclusions reached.

note 1) Phoenix has been jointly developed by the Fujitsu TeamWARE Group (Findland) and the Fujitsu NetMedia Research Centre (Japan). Java™ is a trademark of Sun Micro Systems Inc.

2. An agent-based service brokering architecture

The fast growth of network, service and content providers has led to a complex service delivery environment. Therefore, trading and brokering facilities are required among these stakeholders to support the scalable and efficient deployment of services in this environment. The ultimate goal of the agent-based service brokering architecture described in this paper is to provide such facilities, introducing an architectural framework for the deployment of NGN. To do so, the foundations of this architecture are inspired in two main models: TINA-C business model[1] and the FIPA model for network management and provisioning.[2] Moreover, the idea underlying this novel architecture is to provide an efficient and scalable way of implementing the complex relationships between the various stakeholders of the TINA-C model by using a FIPA agent model and an intermediate broker, that is the Service Portal.

The TINA-C trading model inspiring this architecture is depicted in **Figure 1**. It illustrates a concise view of the various stakeholders involved in the service supply chain as well as their relationships.

As can be seen, there are multiple relationships that can become quite complicated when considering one-to-many relationships. For instance, complex mediation mechanisms will be required so that a subscriber can interact on-line with multiple service providers and network providers.

The agent-based service brokering architecture can be understood as an extension of this abstract model to physically realise the interactions between all these stakeholders in a multi-service and multi-network environment. Taking as a reference the FIPA specifications for network management and provisioning, we have defined a distributed agent-based architecture with a core element (Service Portal) responsible for mediating among all the stakeholders shown in Figure 1. In this approach, each service pro-

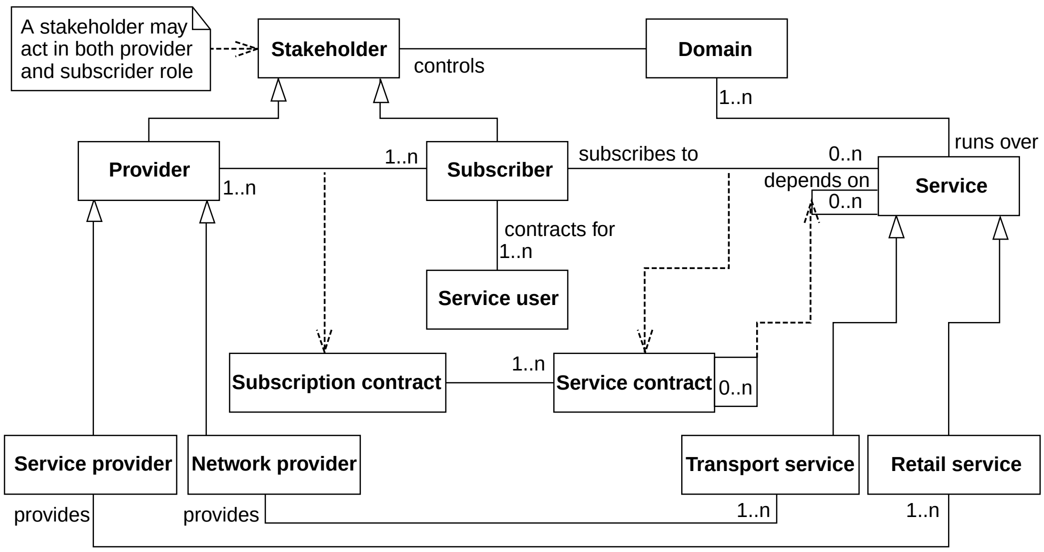


Figure 1
TINA-C trading model.

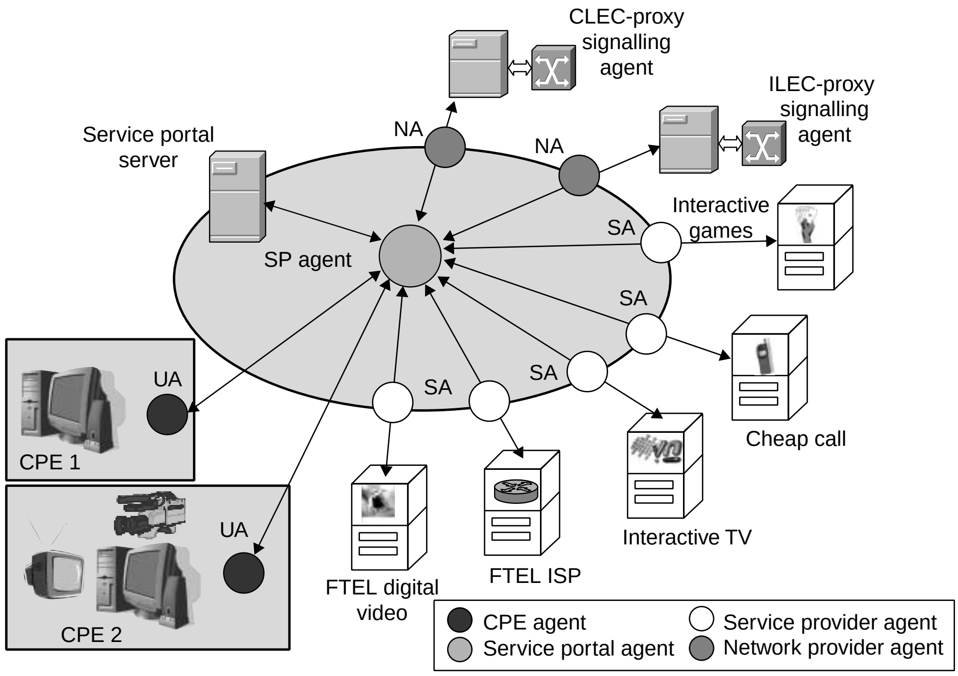


Figure 2
Logical view of the agent-based service brokering architecture.

vider and network provider has its own representative agent. Both providers can respectively inform the Service Portal about service offerings and network resources and prices. In addition, the end-user also has an associated agent that is able to communicate with the Service Portal to access all the available services. The Service Portal is then the key element added to the TINA-C trading model to mediate between providers and end-users (either subscribers or service users) and to allow an open marketplace with multiple service and network providers. A logical view of the agent-based service brokering architecture is illustrated in **Figure 2**:

One of the relevant concepts introduced in this architecture is to offer users the possibility of selecting on-line between various network providers according to the future open network marketplace introduced in.[3] To achieve such a goal, the agent-based service brokering architecture uses separation of service control from service transport following MSF principles.[4] Using a generic control plane independent from the underlying transport networks, this architecture can handle multiple networks as well as provide simpler network devices which are not overloaded with complex call control functionality specific to each network.

In addition, the control plane is made 'always available' and most configuration tasks can be carried out in such a control plane through inter-agent communication without the need for management. For instance, service provider agents can advertise new services to the Service Portal almost instantaneously through the control plane. Similarly, users can select, subscribe and activate services very fast through such plane, whereas service transport will be delivered via bearers specific to each network provider. Therefore, having an 'always available' control plane independent of service transport gives the architecture the advantages of isolating inter-agent control communication from service delivery enables rapid service configuration and provides support for multiple network providers.

Supporting multiple network providers in a flexible manner also means support for multiple network technologies. In fact, besides the independent control plane, this multi-network architecture has been designed with the goal of abstracting service descriptions from the underlying network technology. For instance, a mapping protocol has been designed to map generic service QoS requirements (bandwidth, delay and jitter) to specific parameters of ATM CBR, VBR-rt or UBR connections. Using the available control plane, service providers can inform the Service Portal about the QoS requirements of their services in an abstract manner such that every network provider can subsequently map these parameters to its specific network technology when it comes to setting up the service session.

Likewise, the inclusion of agents in each provider's domain together with the distributed control plane allows this architecture to be extended in a flexible and consistent manner. New service or network providers simply have to advertise their offerings to the Service Portal, which will make them available to end-users. This '*plug and play*' functionality makes it very easy to scale the system as the number of services and providers of those services increase, enabling and embracing Fujitsu's *everything on the Internet* strategic goal.

Finally, it is important to notice that using this distributed agent-based approach, all the stakeholders require relatively low complexity to be able to implement such a complex scenario. As will be discussed in the next section, the individual software agents that have been used in the Service Portal, service providers, network providers and end-users are programmatically quite simple, but they are still able to realise a truly complex business model through collaboration.

Summarising, this agent-based service brokering architecture describes a possible approach to integrate all the stakeholders of the TINA-C trading model, adding an intermediate mediator (Service Portal) and providing a distributed and separated control plane. This then provides the necessary trading and brokering facilities able to cope with the complex service delivery environment that we will face in next generation networks.

3. Agent enhanced service portal (AESP)

In order to verify the architecture described in the previous section, Fujitsu has developed the *AESP*. This section aims to provide a complete description of the system, ranging from its implementation technology and design to the deployed testbed.

3.1 Fujitsu's web computing middleware platform: Phoenix

The AESP has been implemented using a web computing platform enhanced with agent technology called Phoenix. This section provides some insights into the Phoenix architecture and describes some of its most relevant features to this application.

Phoenix, which is a deployment framework for distributed Internet-based services, comprises several important components that make it quite a powerful platform. First of all, Phoenix acts as a Web server, providing an HTTP responder and a runtime environment for Java Servlets, which are the software components used to build Internet-based services with Phoenix (**Figure 3**). In addition, it provides a service API to organise and customise services as well as a set of add-on system services (authentication, session management, billing support, user profile management, administration tools, etc.) that developers can use in their Phoenix-based applications.

Each Phoenix server also has a representative mediator agent developed with Fujitsu's agent programming library Pathwalker.[5] Each mediator agent is configured with information about the services running on the local Phoenix platform (via an add-on system tool called *Advertisement Manager*) and they advertise such information to other mediator agents (**Figure 4**). The key advantage is that the communication plane among mediator agents allows the federation of services distributed over multiple Phoenix servers. Since Phoenix services are implemented with Java Servlets, mediator agents enable local servlets to have sophisticated collaborations with remote servlets beyond the simple HTTP request-response paradigm, allowing a flexible and extendible service brokerage.

Phoenix administrators can dynamically register/de-register new services in the mediator agent using the Advertisement Manager, enabling *service plug and play*. For instance, when a new service is to be made available in the network, only the Advertisement Manager of the local Phoenix platform will have to be updated. The mediator agent will automatically advertise the availability of the new service to other mediator agents so that users can access the new service from any Phoenix platform. A similar process can be applied when a service needs to be withdrawn from the system for maintenance tasks (it is simply necessary to delete its information from the local Advertisement Manager).

Therefore if a user requires a service running on a different Phoenix platform, the HTTP request (from a web browser) will be translated through a gateway to a suitable format for the

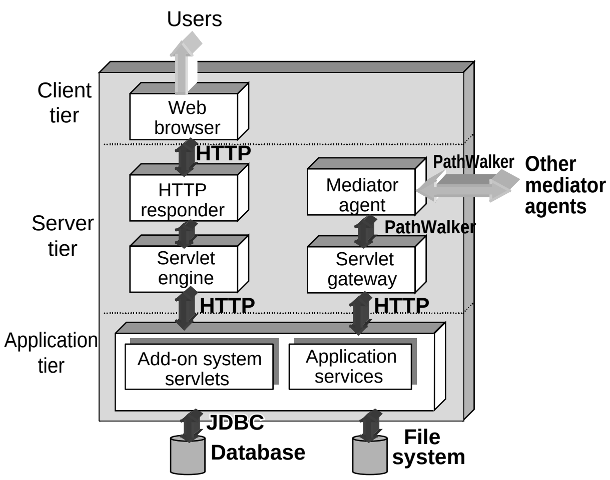


Figure 3
Agent Enhanced Phoenix multi-tier architecture.

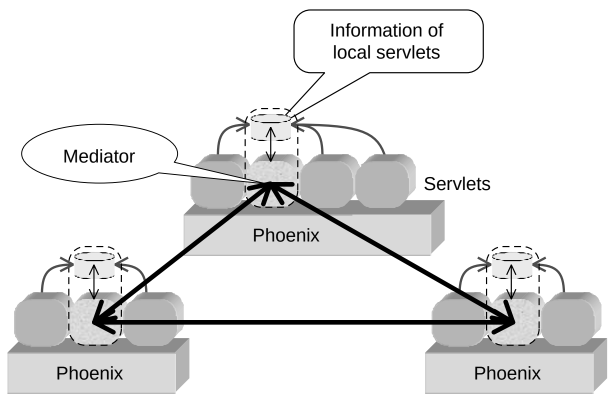


Figure 4
Federation of multiple Phoenix platforms via mediator agents.

mediator agent, which will in turn use its "advertisement" information to route client requests to the Phoenix platform that supports the requested service.

As can be anticipated from the previous paragraphs, the Phoenix platform together with the collaboration of its mediator agents fit reasonably well with the requirements to implement the business logic of the agent-based service brokering architecture introduced in section 2. Therefore, Phoenix and mediator agents have been used to implement the control plane that provides trading and brokering facilities among service providers, network providers and users.

3.2 System design

In order to achieve this aim, the business logic associated with each service provider, network provider, user equipment and the Service Portal itself is implemented as a set of Phoenix-based services. Therefore, each of these parties is provided with its own Phoenix server and associated mediator agent.

For instance, each service provider has been provided with a Phoenix server, which has implemented applications for service administration, such as creation of a new service registering all its associated information (name, type of service, bandwidth requirements, delay and jitter tolerance, etc.) and registration of such services in the Service Portal. When a new service is created and is to be registered, the administration application calls the *Advertisement Manager* of the local mediator agent so that it advertises the availability of the new service to the Service Portal agent (flow 2 in **Figure 5**). Once the availability of the new service is detected and a user tries to browse available services, the Service Portal will issue a request to the corresponding service provider via its agent to retrieve the necessary information to set-up the service account (flow 4 in Figure 5).

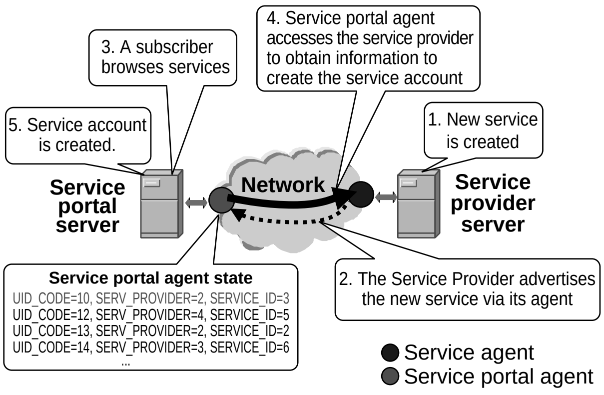


Figure 5
Advertisement of a new service and requesting information to the service provider to create the service account.

This communication plane among agents provides a flexible and scalable architecture to introduce new services in the system in a distributed fashion. Service providers can locally manage their services and service mediator agents will advertise them in order to update the available services in the Portal. Similarly, services can be dynamically de-registered for maintenance tasks. Hence, Phoenix mediator agents naturally enable the service 'plug & play' feature that was considered as a requirement in section 2.

The AESP has also been designed to allow user mobility, i.e.users can access the Service Portal from any Phoenix enabled CPE (Customer Premises Equipment). To do so, each CPE has its corresponding Phoenix server and mediator agent. Since a given CPE might not support all the available services in the system, a configurable profile associated to the Phoenix server of the CPE contains the type of services that can be supported (e.g. VoD, VoIP, Interactive Games) and their corresponding addressing information. This information is required by the Service Portal to know whether the service that the user is trying to activate can be supported in the current CPE and if this is the case, the network endpoints that network providers must use to connect the user to the requested service provider.

When the user logs on using a CPE, his credentials are sent to the Service Portal to be authenticated and the user agent (usually referred to as the CPE agent) is set-up with the identity of

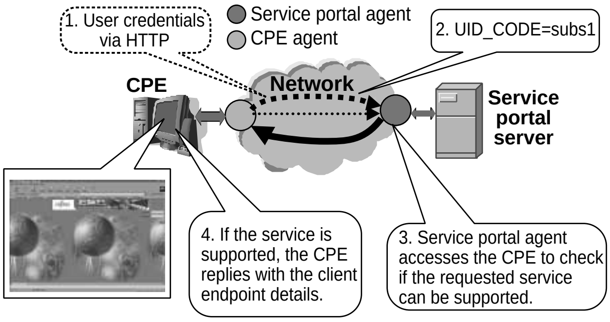


Figure 6
Accessing the Service Portal and the CPE to obtain network endpoint information.

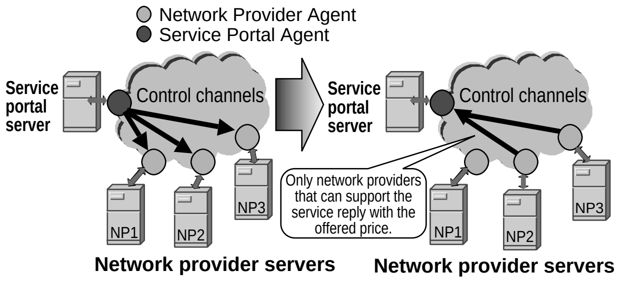


Figure 7
Finding network providers to support the service.

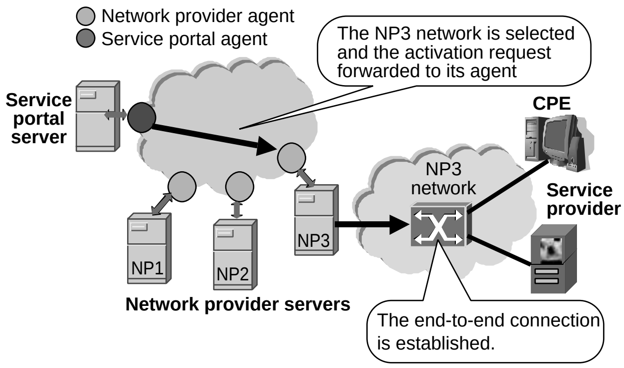


Figure 8
Service activation.

the currently logged on user (flows 1 and 2 in **Figure 6**). The CPE agent will then advertise that user identity to the Service Portal. This enables the Service Portal to know how to reach the CPE from where the user logged on and whether the service to be activated can be supported (flows 3 and 4 in Figure 6). The decision on whether a service can be supported is based on the capability of the CPE, the user's authorisation to access the service and network connectivity.

Finally, and according to the multi-network concept discussed in section 2, the AESP has been designed to support multiple network providers and technologies. In the same way that service providers advertise their services, network providers advertise their transport service. In fact, network providers are also provided with a Phoenix server with its corresponding administration application to register/withdraw their transport service. Therefore, when a user tries to activate a service and the CPE is accessed to ensure that it can be supported, the Service Portal will access all the available network providers to find out which of them can provide an end-to-end connection with the required quality of service. Network providers that are able to provide end-to-end connectivity for the requested service will reply with a price strategy which is made available to the user, who is now able to select the best available deal (**Figures 7** and **8**).

The following section will describe a complete service activation process to show all the interactions among agents and to illustrate how the AESP is able to mediate among users, service providers and network providers.

3.3 Demonstration testbed

The agent-based model described in the previous section has been implemented in a proof of concept testbed to illustrate and verify its capabilities. The logical architecture of this testbed is shown in Figure 2 and comprises five service providers offering Fast Internet access (FTEL ISP), streamed video (FTEL Digital Video), VoD over ATM (Interactive TV), Interactive Games and VoIP (Cheap Call). In addition, it comprises two CPE, referred to as CPE 1 and CPE 2. Whilst the CPE 1 is a PC running Windows 95, the CPE 2 consists of a Windows NT PC and a Fujitsu plasma screen connected to a set-top box that provides termination of the VoD over ATM service. Finally, the testbed has two network providers that are described in more detail in the following para-

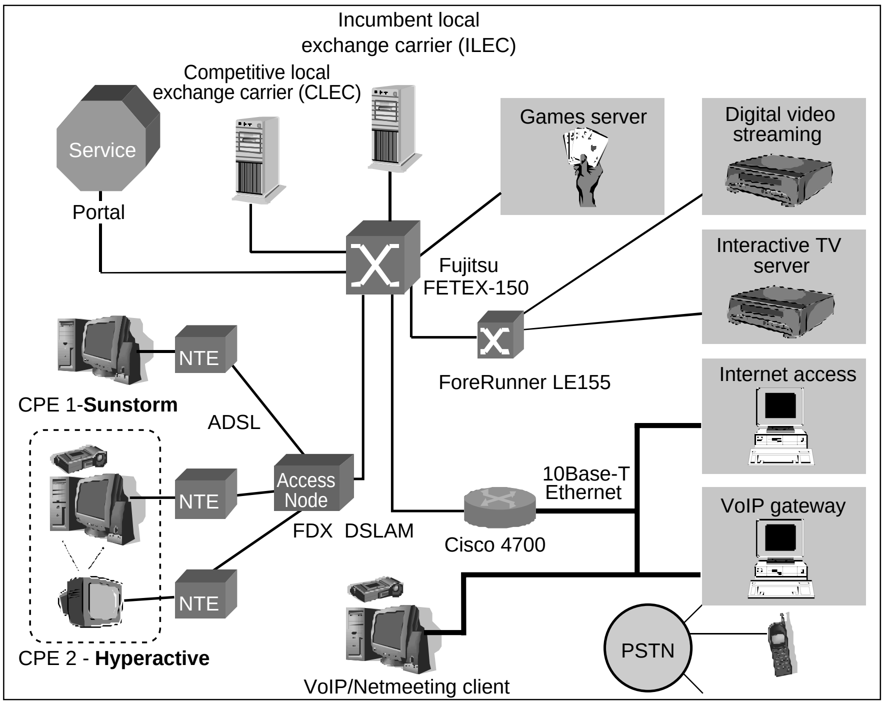


Figure 9
Physical architecture of the AESP testbed.

graphs. The logical architecture maps to the physical architecture as illustrated in **Figure 9**.

As can be seen, a small ATM network has been developed which consists of five service providers, various CPE connected via ADSL to the Fujitsu FDX DSLAM, network providers and the Service Portal. According to section 2 and MSF principles of separation of session control from session transport (in this case ATM transport), all the inter-agent communications take place via control PVCs between the Service Portal and each of the other components of the system. In addition, there are two network providers in the system called ILEC (Incumbent Local Exchange Carrier) and CLEC (Competitive Local Exchange Carrier). They both make use of a TPCC (Third Party Call Control) function based on ATM Forum UNI Proxy Signalling to control the Fujitsu ATM switch to establish service SVCs between CPEs and service providers. Due to the separation of session control from session transport, the AESP is made network agnostic. For example, it could be used to control an IP network with the replacement of the TPCC with an appropriate open call control interface such as COPS or RSVP.

Although in the testbed both the ILEC and CLEC share the same ATM network, they are configured with different QoS capabilities and costs to emulate a multi-network provider environment. In particular, each network provider includes the advertisement of network resources in terms of bandwidth, delay and jitter. This allows the Service Portal to match the capabilities of the networks to the quality requirements of a given service, and retrieving pricing information from only those networks able to provide the required quality. For example, in this testbed, the CLEC represents a high performance network able to support any service, whilst the ILEC represents a network with lower performance that cannot cope with real-time services such as VoD. Since the AESP must be applicable to any underlying next generation network (including IP based transport networks), service QoS requirements are expressed using network agnostic terms such as bandwidth, delay and jitter. Each network provider agent is provided with a translation protocol to map these generic parameters to the parameters specific to the type of network the agent supports. In this case, the network is ATM and hence the generic parameters of bandwidth, delay and jitter are mapped to the specific parameters of ATM CBR, VBR-rt or UBR connections.

As can be derived from the previous paragraphs, the AESP accomplishes all the requirements that were proposed to implement the agent-based service brokering architecture described in section 2, verifying thus the feasibility of such an architecture to manage the complex interactions between the different stakeholders involved in the service delivery framework.

In order to observe the operation and collaboration of the various agents in the system, the messages exchanged among them are captured and displayed in real-time using an applet. As an example of the system dynamics, the following figures depict different screens from this applet in a typical scenario of the AESP. The scenario consists of the following steps:

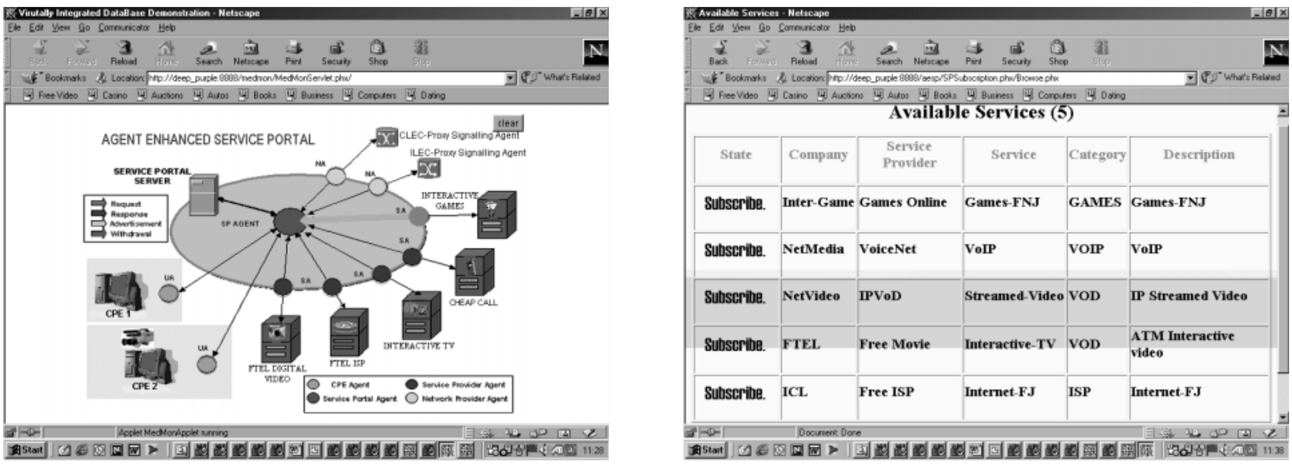
1) A service provider creates and advertises a new service.
2) A user logs into the service portal via a CPE.
3) The user browses a list of available services and subscribes to one of interest.
4) The user then activates the service which initiates a host of brokering activity within the system.
5) Some time later the user de-activates the service.

As the starting point, we will consider the moment in which the service provider Interactive Games advertises its service to the Service Portal. The process followed by a subscriber called *subs1* logging in from CPE 2, who wishes to subscribe and activate the 'Games' service, will be analysed next.

Once the service provider Interactive Games has set up its service, it will advertise it to the Service Portal using the local administration tools. The advertisement process is depicted in **Figure 10 (a)**.

The Service Portal is now aware of the new service. The next time that any user tries to subscribe to services, the Service Portal will issue a request to the service provider to get information to create the service account. Let's consider that *subs1* is the first user trying to subscribe to services since the registration of the Games service.

When the user *subs1* logs in from CPE 2, user credentials will be sent to the Service Portal to be authenticated. At the same time, the CPE agent will advertise to the Service Portal agent that



(a) (b)

Figure 10
(a) Advertisement of Interactive Games (b) List of available services to subscribe.

subs1 is logging in from CPE 2. If the user is successfully authenticated, he/she will be redirected to the subscribers' menu.

The user will then select the option "Subscribe Service" to see all the available services. Because it is the first time the subscription manager is called since the games service was registered, the Service Portal will issue a request to the service provider to retrieve the service information (service account details, network resources and service endpoint information). A list with all the available services and a link to subscribe to them will then be displayed as depicted in **Figure 10 (b)**.

Let's suppose the user decides to subscribe to the Games service. The subscription manager will carry out the subscription and will offer the user with a GUI to start the activation process selecting potential networks.

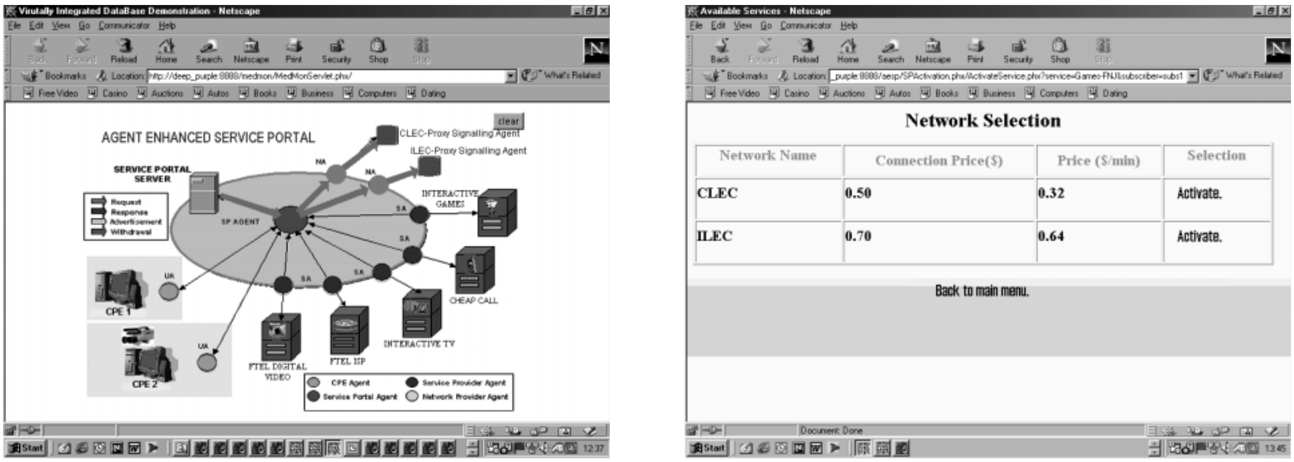
The user clicks "Select Network" and Service Portal agent will access the corresponding CPE to check whether the service can be supported. Since CPE 2 can support the Games service, the Service Portal will then seek network providers able to provide an end-to-end connection between the user and service provider with the required QoS (specified in terms of bandwidth, delay and jitter). All the networks able to support this service will reply with an offer (in terms of price) for the service transport. **Figure 11** shows this process:

The user will next select the cheapest option (in this case the competitive carrier), and the corresponding activation request will be issued to the CLEC to set up the end-to-end connection. From this moment, the service is available and the user could start playing the game. Once *subs1* wants to stop using the service, it can click "Deactivate" and the corresponding deactivation request will be issued by the AESP to the network provider representing the CLEC. The AESP also maintains an account of service usage statistics that can be used for billing purposes.

4. Conclusion

This paper has described the definition, design and implementation of a distributed agent-based architecture that is able to provide brokering facilities to manage interactions in a multi-service and multi-network environment. The resulting testbed demonstrates the feasibility of such an architecture and highlights its main benefits, which include:

- A Service Portal that acts as a broker for ser-



(a) (b)

Figure 11
Network selection process. (a) Both networks are offered the service (b) Both networks respond with a price to transport the service.

vices and is able to manage the complex stakeholder relationships that will be prevalent in next generation networks. It further mediates between end-users and the other stakeholders in the service supply chain in order to hide the complexities of configuring and accessing services.

- Separation of service control from service transport, allowing the control of multiple and heterogeneous networks using a common control plane and providing a path to simpler and cheaper network devices. This also enables the future open network marketplace with multiple network providers offering their transport service to the end-user.
- Support of multiple network technologies is also achieved by abstracting the service QoS requirements from the underlying technology. Network providers can offer their transport service and prices to the Service Portal in such a way that users can select the most appropriate option at any point in time.
- A scaleable and flexible architecture, enabling both *service and network plug and play*, i.e. new service and network providers can be dynamically registered through agent collaboration. In addition, service providers can also introduce new services at run-time and their agents will advertise the real-time availability of each service to the Service Portal.
- User mobility (*user plug and play*), i.e. users can access the system from multiple CPEs because CPE agents will collaborate with the Service Portal agent in order to provide location information on the user.
- Relatively simple distributed software entities (that we call agents) are able to co-operate to implement the complex trading model of the service supply chain.

If we contrast these key advantages with existing broadband networks where services are statically provisioned and that involve in-flexible management processes, where session control is tightly integrated into session transport, it can be seen that agent technology is a promising candidate for building future multi-service and multi-network brokering architectures. The results of this initial work are very encouraging and confirm the important role that agent technology will play in enabling the usability of NGN services and the development of scaleable and flexible systems that are cheaper to build, change and discard.

Further work is planned to compare the performance of the agent based prototype, in terms of resource usage (processing, memory and storage), with more conventional distributed software implementations. The application and comparison of the agent enhanced Phoenix middleware with other agent middleware together with the integration of AESP with the brokering of network resources described in[3] is also planned.

References

1) TINA-C Service Architecture Version 5.0, June 1997.
http://www.tinac.com/specifications/specifications.htm.
2) FIPA 97 Specification, Part 7: Network Management and Provisioning. OC00016, October 1997.
http://www.fipa.org
3) D. Chieng and I. Ho: Brokering Marketable Network Resources Using Autonomous Agents. The XVII World Telecommunications Congress/ISS2000, May 2000.
4) Multi-Switching Forum.
http://www.msf.org
5) Pathwalker Programming Library.
http://www.fujitsu.co.jp/hypertext/flab/free/paw/index.html

Francisco Navarro is a recent MSc. Graduate in Telecommunications Engineering from Valencia University (Spain), who received a Socrates-Erasmus grant to do his final year project at the School of Electronic and Electrical Engineering at the University of Birmingham (UK). His project was sponsored by FTEL and covered the research and deployment of the agent-based multiservice platform. Francisco has a solid background in agent technology, Java agent-based programming and ATM networks.

Keith Jones is Chief Engineer, Product Strategy, within Fujitsu Telecommunications Europe Ltd, Birmingham, UK. His responsibilities include definition of product and technology strategy, input to global product planning and international standards initiatives and providing direction to FTEL's research programmes. Keith has a background in the system design of narrowband IN and SS7 systems, his recent work has focused on definition of next generation xDSL fixed access products and service platforms for ATM and IP based next generation networks. Keith is a Chartered Engineer and is a member of the IEEE and the IEE and is on the editorial panel of the IEE's Electronics & Communications Engineering Journal.

Sagar Gordhan graduated from London University with an MSc in Telecommunications Engineering. Since then he has researched various artificial intelligence techniques, including software agents, and applied them to the complex problems of ATM and IP network management. Currently he is a Senior Systems Engineer in DSL Customer Solutions, within Fujitsu Telecommunications Europe Ltd., UK, where he is responsible for specifying and designing solutions for the delivery of multiple services over xDSL networks. He actively participates in international standards looking at next generation network services and architectures and provides input into FTEL's product evolution strategy.

Nigel Garnham gained his BEng Honours degree in Electrical & Electronic Engineering from Newcastle upon Tyne Polytechnic and Ph.D from the University of Nottingham. Having started his career in the custom microelectronics industry, he joined BT Laboratories in 1995 to investigate low bit-rate video coding, interactive television and Quality of Service performance in managed networks. Now Product Manager in the DSL Solutions Group at Fujitsu Telecommunications Europe Limited, he originally joined to work in Fujitsu's systems design team for the deployment of ADSL to the UK Network. He is a Chartered Engineer and a Member of the Institution of Electrical Engineers.