A generic component for managing service roles

Thanassis Tiropanis, Chris Malbon
University College London, Computer Science
Gower Street, London WC1E 6BT,
United Kingdom
Tel: +44-171-419 3687, +44-171-391 1327
Fax: +44-171-387 1397
E-mail: {t.tiropanis, c.malbon}@cs.ucl.ac.uk

Hervé Karp
Atos Integration
Thalassa B, 444 rte des Dolines - BP 67
06902 Sophia Antipolis, France
Tel: +33 4 92 96 88 48
Fax: +33 4 92 96 88 49
E-mail: hkarp@atos-group.com

Abstract

There are a number of architectures that describe how service providers can provide telecommunications services to their customers. Architectures like TINA address service control and service management issues for communication exchanges among human service users. In this paper we point out the importance of role-based exchanges in a telecommunications environment and we present a generic component that provides role management. We describe how this component was implemented and integrated in a TINA-like environment. We also describe why a role can be modelled as a mobile entity and how role mobility relates to role management. Finally, some conclusions are drawn.

Keywords

Service management, personal communications, role management, role mobility, TINA

1 INTRODUCTION

Continued global deregulation and the harnessing of new technologies has had an enormous impact on the telecommunications industry, leading to a dramatic increase in the number and type of services that telecommunications companies can offer. Research, too, within the industry has broadened to encompass all parts of the business and has been the subject of a number of ACTS projects sponsored by the European Community to look specifically at service management and the problems associated with providing network based services to a wider community.

A number of independent industry bodies such as TINA and the ITU have sought to define standard architectures for the control and management of telecommunication services, free of commercial interest and open to all. The Open Service Market (OSM) as envisaged by these bodies have identified rapid commercial service deployment as well as openness as a key requirement, and for this reason, the definition of reusable components for service control and service management is a common feature of the proposed architectures [TINA-SA][M.3400].

In particular, the TINA consortium has attempted to specify a service architecture [TINA-SA] addressing the key areas of service management (subscription, accounting, profile management) [TINA-SAA] and service control (setting up and controlling the exchanges among service users). This has been used as the basis for further investigation into service management and service control by a number of European projects [P-D5.1B][VITAL-V2].

In this paper we hope to show that the standards bodies and consortia have overlooked an important area of service management: role management. We observe that many communication exchanges are not user-based (as so far considered) but can be described as role-based. For this reason, role management has an impact on the service control level too. We present a generic role management sub-system and how it was deployed in a TINA-like environment. Finally we draw some conclusions based on our experience and look at further work that can be done to advance this interesting area.

1.1 Concept of a service role

Many of the telephony exchanges today are not on a personal basis. For example when a telephone user contacts the operator to get some information they are not contacting the operator as a person but as someone who can perform the tasks of the operator. The user would have contacted anyone who could perform the operator tasks. For this reason, this kind of exchange is between the *user* and the *role* of the operator who is outlined by the operator tasks that he performs. The tasks of the operator role express an obligation from the telephony provider to the user and they are related to the provision of the service. Therefore, we can call the operator role a *service role*.

We define a service role as a set of tasks, which express a contractual relationship between a service provider and a customer. In the wider context we consider a service role to be an entity that can take part in exchanges with users or other service roles. In this paper the terms role and service role mean the same thing and they are used interchangeably.

A service role is a virtual entity that can be carried out by a person. We call *role-holder* the person who can carry out a specific role.

Existing role theory [BIDDLE-79] defines a role as a set of rights and duties. Lupu and Sloman [LUPU-97] [LUPU-95] used roles to model an organisation's structure in terms of the access rights and obligations that different positions within an organisation have to a set of target managed objects. In our case, a service role describes mainly a set of duties, which derives from the contractual obligations of a service provider to a customer. We use service roles to encapsulate those positions within a service provider organisation that a customer can contact or be contacted by to fulfil the contractual obligations of the service provider [TIROP-98]. We also recognise that this is a two way process and that it is possible to have roles within the customer domain fulfilling customer contractual responsibilities to the provider.

It is out belief that every service provider needs a number of service roles. There are *service independent roles*, like the "service administrator" role or the "help-desk" role. We can also have *service dependent* roles like the "doctor on call" role in a medical advice service, or the "tutor" role in a Tele-education service. We state that as the number and type of services on offer diversify so the need for better and more effective service support and service provision grows. We believe that the concepts of service roles and role management can assist service providers to fulfil their obligations to their customers for service support and service provisioning more effectively. As such, we believe that a service provider must not only support person to person communication, but also role-person and role-role communication.

1.2 Role management

A role can be carried out by a number of persons at the same time. A person who can fulfil a specific service role we call a *role-holder*. A role-holder who is able to fulfil the role when requested is known as an *active role-holder,* of which there may be many at any given time.

A role management system simply allows service roles to be managed within a service. We have identified three areas of functionality in a role management system:

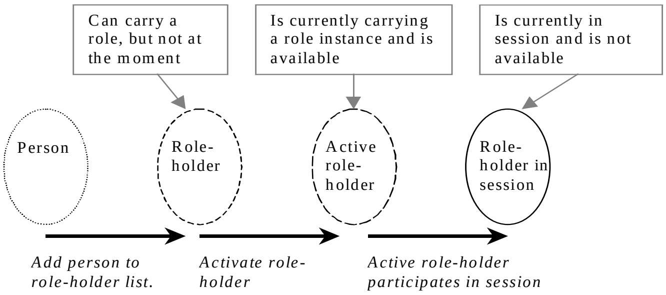

Can carry a role, but not at the moment
Is currently carrying a role instance and is available
Is currently in session and is not available
Person
Role-holder
Active role-holder
Role-holder in session
Add person to role-holder list.
Activate role-holder
Active role-holder participates in session

Figure 1: States of a role-holder.

1. *Role life-cycle management*: is concerned with creating/deleting /updating service roles.
2. *Role-holder management*: provides for adding/removing role-holders for a service role and for activating/deactivating role-holders. Figure 1 shows the three states that a role-holder can be in. The activation/deactivation of role-holders can be policy based.
3. *Call handling*: manages the forwarding of a call to a role. For example, when a call is addressed to a service role, the role management system decides which of the available active role-holders should service the call. Call handling can be policy based.

1.3 Role mobility

A mobile entity is a well-known feature of most topics to do with mobility in telecommunications. It can change location without any disruption to the communication environment or to the service on offer. The location of a mobile entity is another entity via which the mobile entity can be reached. For example, personal mobility features a person that is able to change location. The location of a person is a terminal [F.850]. Via this terminal a person can be reached. Similarly, terminal mobility features a terminal that is able to change location. The location of a terminal is a network access point via which the terminal can be reached.

In a similar way, a role is a mobile entity, which can change location. The location of a role is a person (the role-holder) [TIROP-98]. A role is reached via a person (a role-holder). The difference between role mobility and other kinds of mobility is that multiple instances of a role can exist at a time. A service role encapsulates a set of obligations and tasks, and it makes a virtual entity that can be attached to a number of role-holders. N.B. a service role is not a mobile agent, and role mobility does not imply agent mobility.

Role management is a kind of mobility management. In the modelling process of the role management component, we used techniques similar to those for managing personal or terminal mobility. For example, we used unique IDs to identify roles in the service provider's domain and we nominated an entity, a Role Agent to delegate for a role at all times, in a way similar to user agents or terminal agents. Modelling role management as mobility management proved a very useful exercise since it gave us another insight into mobility management. Our conclusions on this issue are presented at the end of this paper.

2 A ROLE MANAGEMENT COMPONENT

In this section we give a high level description of a role management component. First we give the use cases that outline the functionality of the component and then we look at certain design considerations with respect to the information and computational viewpoints.

2.1 Use cases for the role management component

The functionality for the role management component can be described with a number of use cases [TIROP-98]. The entities that interact in these use cases are:

- *The end-user*: who is the end-user of a service.
- *The role management system*: which is the computing facility that manages roles. The role management system resides on the SP domain.
- *The role-holder*: who is a user who can carry a role.

We distinguish between *active role-holders* and *inactive role-holders*.

The use cases for role management and role-based exchanges are divided into four groups:

1. *Role management use-cases that are initiated by the provider service administrator*: These provide for creating, updating or deleting roles. They allow the service administrator to add or remove role-holders for a particular role and to set-up policies for determining when a role-holder is activated (becomes available) or deactivated (becomes unavailable) or which of the active role-holders is to take a particular call to a role.
2. *Role management use cases that are initiated by the role-holder*: These use cases allow a role-holder to become active (available for calls) or inactive (unavailable), at will.
3. *Role browsing*: Use cases that allow the service provider administrator or end-users or role-holders to browse through the roles of a service provider and get details about them.
4. *Call use cases*: These allow an end-user to call an instance of a role. In other words, they allow an end-user to contact an instance of a service role carried by a role-holder. Also they allow for a role-holder to make a call to a person as the role; for example they allow a

person (role-holder) who is carrying the service administrator role to make a call to an end-user as the service administrator.

This is a high-level specification of the functionality of the role management component. In the reminder of this section we look at certain considerations that were taken into account during the design of this component.

2.2 Information level issues

A role is identified by means of a role ID. Since the scope of a role is the service provider domain, the scope of the uniqueness of a role ID is the service provider domain. From the use cases we can see that the following kinds of exchanges can take place in a communications environment with roles and end-users:

- A role (which is carried by a role-holder) contacts another role (on another role-holder).
- A role (which is carried by a role-holder) contacts an end-user.
- An end-user contacts a role (via the role-holder that is carrying that role).
- An end-user contacts another end-user.

The group of the different roles that are supported in a service provider's domain can be contacted by the end-users and the other roles of that domain. Similarly, the group of end-users of a service provider's domain can be contacted by the roles and the other end-users of that domain. This makes it clear, that in order to address someone in this environment, we must be able to specify first whether it is a role or a user and then to specify which exactly role or end-user is being addressed. The service control functionality supports this kind of addressing by means of a called party identifier that is composed of two fields:

<calledPartyIdentifier>=<calledPartyType><partyID>

The calledPartyType can either be *role* or *person* and the partyID identifies a person or a role *uniquely* in the set of persons or roles. It is the responsibility of the SP to make sure that a role ID is unique in the service provider domain.

There is certain information that is associated with a role. This information includes the role ID, the list of the user IDs of the role-holders, the list of active role-holders and a number of policies regarding the activation/deactivation of role-holders and determining which of the active role-holders will take a call to the role.

2.3 Computational level issues

A role is a virtual mobile entity, which is outlined by a number of tasks. This means that the role is not a physical mobile entity like persons in personal mobility, terminals in terminal mobility or agents in agent mobility [BREUGST-98]. Although it is not a physical entity, a role can still be delegated for by a *Role Agent* (RA) in a similar way like persons or terminals, which are delegated for by user agents or terminal agents.

The use of a Role Agent to delegate for a role has certain benefits. The RA of a role can be always available even if there is no instance of a role carried by a role-holder at some moment. Also, The RA can handle calls to a role by means of a role profile and it can also enforce access control to accessing the role. The RA can initiate more instances of a role on demand, if there are many callers to the particular role. Finally, the RA keeps track of where the different instances of a role are available (the role-holders who are currently carrying the role).

Of course there are some drawbacks. Firstly, a caller has to contact the RA before they can get through to the role-holder and the role. Second, if the RA is not available, the role cannot be contacted even if it is instantiated on a number of role-holders.

Assuming that the users are delegated for by user agents and since the role-holders are users who are delegated for by user agents too, we make sure that the user agents of the role-holders support the necessary exchanges with RAs (for example for activating a role-holder).

Although a role is a mobile entity, the RA does not need to be mobile. In order to get in contact with the RA of a role we need to use another entity the Role Agent Locator, or *Role Locator (RL)* which can return the location of a role agent given a role ID. We note that the RL locates RAs and not roles.

A service provider can offer a number of roles. In order to keep track of all the roles in a service provider domain we employ a Role Registrar (RR). This entity manages the life-cycle of roles and RAs, i.e. it can create/update/delete RAs and it can return information about all the roles of a service provider domain.

3 AN IMPLEMENTATION OF ROLE MANAGEMENT

After taking the considerations of the previous section into account, the European ACTS project Prospect designed and prototyped a role management system. This service management component was integrated with the service management functionality of some service providers in Prospect’s environment. The following sections describe the Prospect environment and the design and implementation of the role management component.

3.1 The service environment in PROSPECT

The main purpose of Prospect is to demonstrate service management by means of a trial of a tele-education service (TES). This high level service provides courses to the students of a customer organization who have been authorized to access it from the establishment of a contract between the TES provider and the customer organisation. The TES is based on an integration of multi-media tele-services (MMTS) [LEWIS-97]: a multi-media conferencing service (MMC), a WebStore service allowing to store/retrieve information from a Web server, a Hyper-Media service providing functionality above a Web server. The Prospect Management system is based on the one-stop-shopping paradigm, therefore the Customer needs to subscribe and interact only with the TES provider

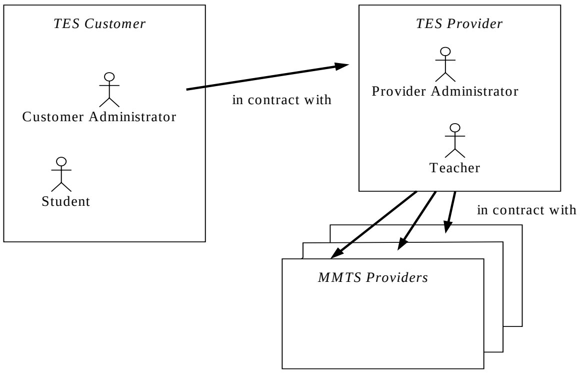


Figure 2: Prospect Enterprise Model.

The Enterprise Model of Prospect features a number of stakeholders with contractual responsibilities to each other (see Figure 2). The following actors and systems are employed to carry out the contractual responsibilities between stakeholders.

- The TES Provider system, which provides control and management for the TES.
- The TES Provider Administrator, who is responsible of the administration of the TES.
- The teacher, who provides the content of the course and delivers the course.
- The TES Customer Administrator, who is responsible of the Customer domain management regarding the TES.
- Students, who attend the course.
- The MMTS Providers systems, which provide control and management of the MMTSs.

The service management functionality of the service providers' systems in Prospect includes subscription and accounting management. During the last phase of the Prospect trial the TES service management has been enhanced with a Role Management subsystem.

3.2 Implementation use cases

At the design level three implementation scenarios were considered: a Support Desk, a TES centre, and a medical centre. Of these, we chose to implement the TES centre, primarily due to the ease with which the existing TES in Prospect could be extended. In particular we wished to show the management of roles and how a student of the TES could contact a tutor by role. Figure 3 illustrates the use cases considered and shows the interactions between the TES Provider Administrator, the TES Customer Administrator, the role-holders, and the end-users (TES students) in UML notation.

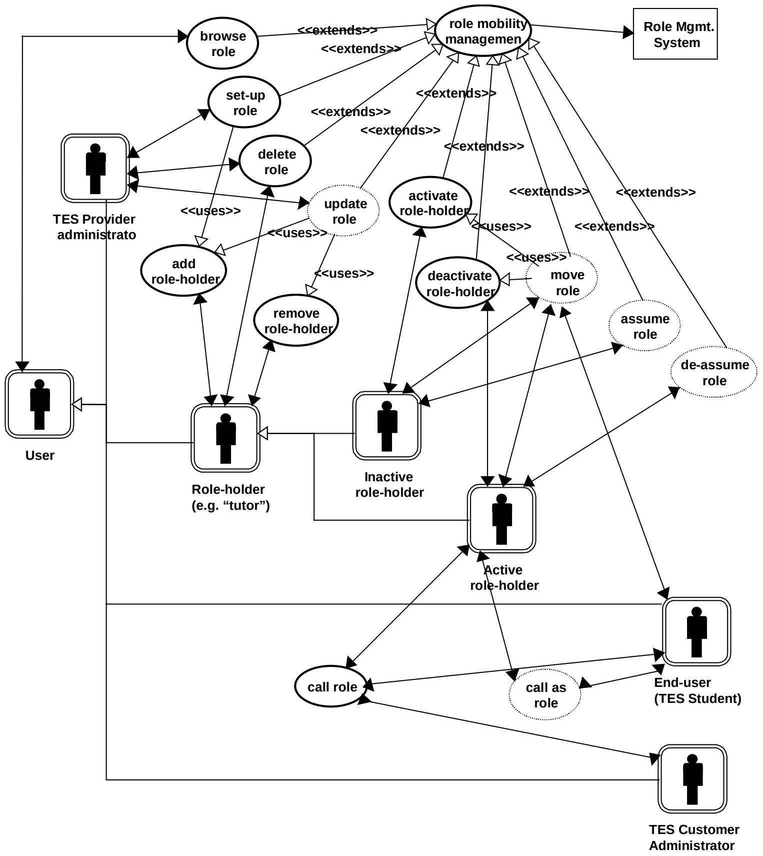


Figure 3: Role mobility use cases.

From the role management use cases that were initially considered, the ones that were eventually implemented and demonstrated are highlighted in Figure 3. These are:

- *Browse role*: the end-user receives a list of roles and can ask for details on a specific one.
- *Set-up role*: the service provider administrator creates a new role, including all role-holders and policies.
- *Delete role*: a role is deleted from the role management system.
- *Add role-holder*: a new role-holder is registered for a role.
- *Remove role-holder*: an existing role-holder is removed from a role.
- *Activate role-holder*: the service provider/administrator instructs an existing role-holder to become active (or the system activates a role-holder based on predefined policies).
- *Deactivate role-holder*: the administrator instructs an activated role-holder to stand down (or the system deactivates a role-holder based on pre-defined policies).
- *Call role:* a user wishes to contact a role (in our case, a student contacts the "tutor" role) and gets through to one of the role-holders who are carrying this role at that time.

3.3 Role Management sub-system

This section describes the implemented Role Mobility sub-system and includes descriptions of the following components: the Role Registrar (RR), the Role Agent (RA), the Role Locator

(RL) and the User Locator (UL). It includes additions to the User Agent (UA), and a brief description of the Provider Management User Application (PMUAP) and the User Application (UAP). The components of the role management system and their interfaces were implemented in Java with CORBA support [OMG]. A view of the components and their relationships is given in the diagram of Figure 4.

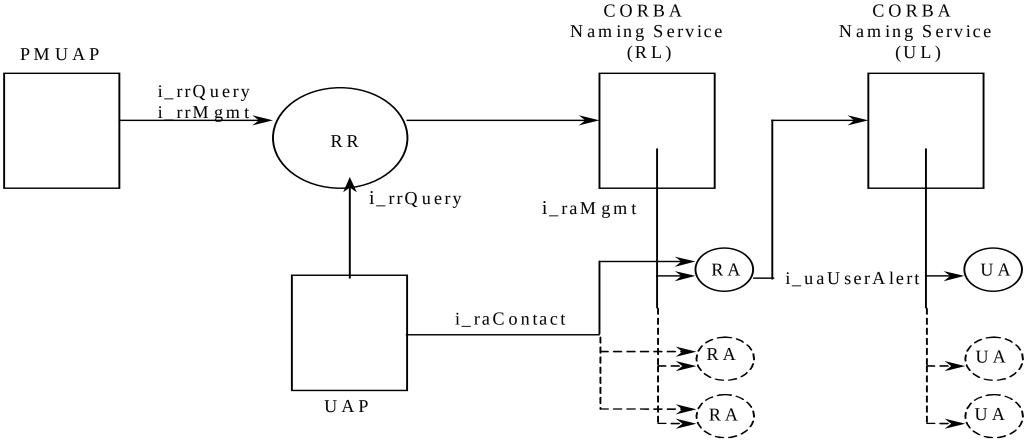


Figure 4: Component view of the role management system.

3.3.1 Role Registrar

The RR manages the life cycle of the roles of the service provider. RAs are manipulated through the RR's *i_rrQuery interface*, allowing roles to be browsed and queried, and *i_rrMgmt* interface. The *i_rrMgmt* interface is used by the PMUAP to create, modify and delete roles, to alter role-holder lists, to change role and role-holder policies, and to activate and deactivate role-holders.

When a role is created the management system instructs the RR to create a new RA. The RR assigns to the role a unique identification moniker, a role ID, which is used for unique identification. RAs are stored as a collection of objects. In the original design storage was managed by a Role Locator (RL), however, for expediency, we used our chosen vendor's implementation of the CORBA Naming Service to publish a reference to each RA object. Role Agent objects were stored persistently using Java's Object Serialization interface.

3.3.2 Role Locator

The RL is responsible for creating and publishing RA IOR's. Obviously there are security issues involved as it is unwise to make sensitive object information available to a wider community, however, in our model the Naming Service ran locally within the provider domain and was therefore as secure as any other information within the provider's domain.

3.3.3 Role Agent

The role agent contains a collection of role-holders that fulfil the requirements of that role. There may be one or more role-holders assigned to the role but only a subset of these will be available at any given time, these are known as Active Role-Holders. The role-holders' activation status is maintained by the RA and is based on some policy. A policy is a rule that determines when a role-holder becomes active or not, and how an active role-holder is chosen when a request from a user is received.

Policies are set up through the PMUAP and the *i_raMgmt* interface, and can either be applied at the RA level in which case they apply to all role-holders, or at the individual role-holder level when they apply only to that role-holder. It is the responsibility of the management system to ensure that in the case of policy conflicts that one policy is preferred over another.

When a user wishes to contact a role they do so by calling the *i_raContact* interface of the required RA directly from the UAP. The sequence of events is as follows: the user asks the RR, by way of the *i_rrQuery* interface, for a list of supported roles, which the RR obtains from the Naming Service; this list contains the role-id, a role description and an IOR for the *i_raContact* interface for the RA of each role; once the user has selected the desired role, the IOR is narrowed and an invitation can be made to a role-holder.

To optimise the search, active role-holders are posted onto an Active Role-Holder List. Role-holders can notify the list when they become available. Proactive notification saves time searching for active role-holders on each request. Once an active role-holder is identified the RA locates the role-holder's *i_uaUserAlert* interface, an extension to the interfaces offered by the role-holder's UA (see below). The role-holder is informed of the request and invited to respond. If accepted the call can commence, if rejected another role-holder is selected until all active role-holders have been exhausted.

3.3.3.1 Role-Holder

Each role-holder maintains a policy list onto which new policies are posted and ordered. Policies determine when and how a role-holder becomes active.

Our implementation upheld three role-holder selection policies: selection based on priority, whereby a role-holder held some rank and was selected based on that rank, the higher the rank the more likely the role-holder was to being selected; selection based on preference, whereby a user could stipulate a preferred role-holder, if available this role-holder was requested above all others; and selection based on equality, whereby all active role-holders were placed in a queue, once a role-holder had finished servicing a call they were placed to back of this queue.

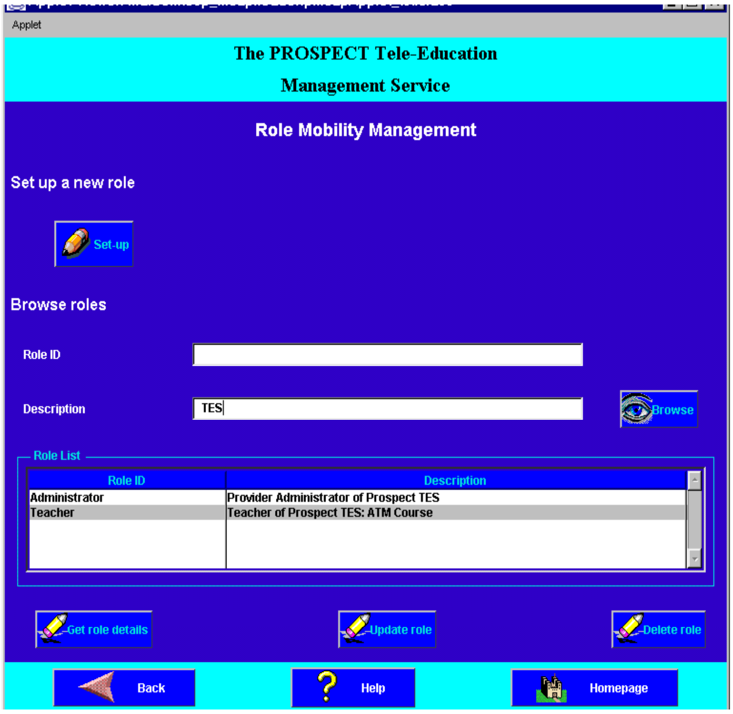
3.3.3.2 Policy Element

A policy is an entity that operates at either the Role Agent level over all role-holders, or at the role-holder level.

Policy elements are created and/or modified at the PMUAP level before being assigned to a role-holder/role, and determine how and when role-holders active. Although policy was pre-determined the use of a policy element allowed us the flexibility to create new policies. The only implementation headache then being one of presentation to the administrator.

3.3.4 User Agent Extensions

In order for a role-holder to physically be contacted by the RA or end-users the role-holder's UA required extending so that they could be notified of incoming requests for activation/deactivation/calls. The *i_uaUserAlert* interface enabled role-holders to be contacted in two ways: either by notification, requiring no role-holder interaction; or by invitation, requiring a response.



Role ID

Description TES

Role ID	Description
Administrator	Provider Administrator of Prospect TES
Teacher	Teacher of Prospect TES: ATM Course

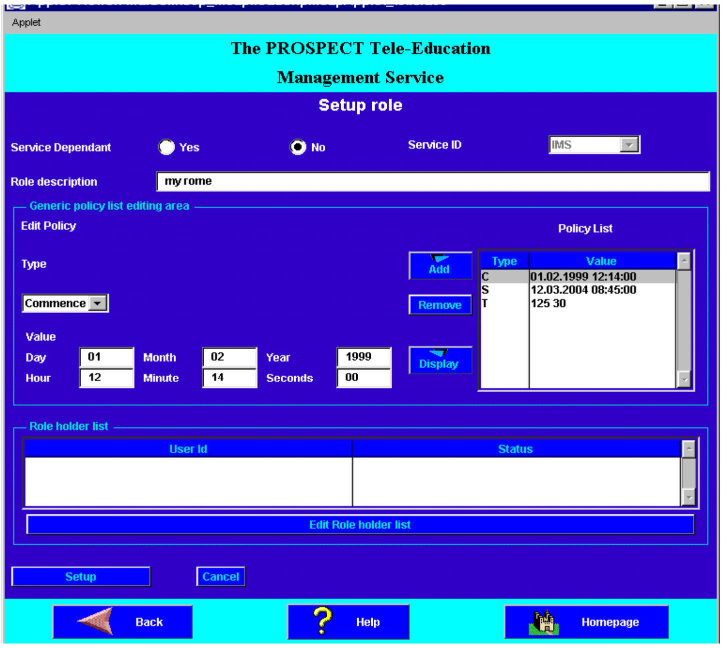
Figure 5: Main panel of the Provider Management UAP for Role Management.

The design stipulates the use of a User Locator (UL) computational object to bring in contact a RA to the UA of a role-holder. A RA queries the UL using a known user ID which would return the appropriate interface. Again, for expediency, we chose to implement the UL using the Naming Service, publishing the *i_uaUserAlert* IOR using a defined naming convention.

3.4 The Graphical User Interface

The Provider Management UAP allows the Provider Administrator, in charge of the role management, to manage the information relative to the roles. The main functions provided are:

- *Browse role*, allowing to get a list of roles from user defined criteria (see Figure 5).
- *Set-up a new role*, allowing to create a role including a generic policy list and a role-holder list (see Figure 6).
- *Get role details*, allowing to get all the information relative to a role and details about each role-holder.
- *Update some role information*, allowing managing the role-holder list and activating or deactivating role-holders.



Applet

The PROSPECT Tele-Education

Management Service

Setup role

Service Dependant ○ Yes ● No

Service ID: IMS

Role description: my rome

Generic policy list editing area

Edit Policy

Type: Commence

Value

Day: 01 Month: 02 Year: 1999

Hour: 12 Minute: 14 Seconds: 00

Policy List

Type	Value
C	01.02.1999 12:14:00
S	12.03.2004 08:45:00
T	125 30

Role holder list

User Id	Status

Figure 6: Panel of the Provider Management UAP for the set-up of a role.

4 CONCLUSIONS

From the observation that many exchanges in communications services are role-based and from the plethora of the service roles that we identified for a number of services in Prospect (although a few were implemented), it is apparent that role management is an essential part of service management. The advanced services that we expect to see widely available in the near future (such as multimedia conferencing, Tele-training, and others) require a significant number of service independent roles. Also, many of these services require service specific roles (such as the tutor role for the Tele-education service) which can be specified and deployed in a generic and efficient way. Requirements for reusability and modularity of service management functionality led us to the deployment of a generic, reusable service management component, which could be integrated in the TINA-like environment of Prospect next to the subscription and accounting management.

There are some requirements though, on the service control plane too. Role-based exchanges must be considered on this level and the service control should allow for service users to be able to address and contact service roles and vice-versa. We only required a minimum number of modifications to the TINA-like service control of Prospect in order to allow for role-based exchanges.

By modelling a service role as a mobile entity and by treating the concept of role mobility in a way similar to other kinds of mobility (such as personal and terminal mobility) we could reuse design patterns that we used for managing personal mobility in Prospect. For example, we used a RA and role profiles in a way similar to User Agents and user profiles. This exercise also gave us an insight into the nature of mobility. A service role as a mobile entity has some peculiarities, which may be encountered on other kinds of mobility too. One peculiarity is that a service role is always mobile. Since a role is carried out by a person, and since a person cannot be continuously available, the service role has to move from person to person frequently. Another peculiarity is that a service role can have many instances available each time. This is a quality that we do not encounter in personal mobility but which can be encountered in agent mobility with multiple agent instances [BREUGST-98].

The role management component that was deployed in the Prospect multi-service environment proved generic and useful to the service administrators. Roles can be rapidly defined and deployed. The automated interactions between the role management system and the role-holders (for adding/activating role-holders) were efficient and they saved time and effort for the service administrator. The graphical user interface provided a comprehensible abstraction of the concepts of service roles and role management to the service administrators.

5 FURTHER WORK

The deployment of role management in non-TINA environments such as the Internet is an area that we will investigate in the future. The reuse of the existing role management component in other TINA or TINA-like environments is also of interest to us. We are looking at possibly evolving the concept of service roles to the general concepts of role theory [BIDDLE-79]. We are also planning to investigate interactions between service roles (for example the "service administrator role" contacts the "help-desk" role).

6 ACKNOWLEDGEMENTS

The main part of this work was done under the auspices of the European ACTS project Prospect (contract number AC052). We are thankful to everybody in Prospect and to the PCS activity in particular for their useful feedback and support in this work. The views and ideas expressed in this document do not necessarily reflect those of the Prospect consortium.

7 REFERENCES

[BIDDLE-79] B. J. Biddle, E. J. Thomas, *Role theory: concepts and research*, Robert E. Krieger Publishing Company, New York, 1979.

[BREUGST-98] Markus Breugst, Thomas Magedanz, *On the usage of standard mobile agent platforms in telecommunication environments*, Proceedings of the 5th international conference on Intelligence in Services and Networks, IS&N '98, Antwerp, Belgium, May 1998.

[F.850] *Principles of universal personal telecommunication (UPT)*, ITU-T standard F.850, ITU-T, 1993.

[LEWIS-97] D. Lewis, T. Tiropanis, A. McEwan, C. Redmond, V. Wade, R. Bracht, *Inter-domain integration of services and service management*, Proceedings of the fourth international conference on intelligence in services and networks, IS&N '97, Cernobbio, Italy, May 1997.

[LUPU-95] Emil C. Lupu, Damian A. Marriott, Morris S. Sloman and Nicholas Yialelis, *A policy based role framework for access control*, First ACM/NIST Role Based Access Control Workshop, Gaithersburg, USA, Dec. 1995.

[LUPU-97] Emil C. Lupu, Morris Sloman, *Towards a role based framework for distributed systems management*, Journal of Network and Systems Management, vol. 5, no. 1, Plenum Press, 1997.

[OMG] *The Common Object Request Broker: Architecture and Specification*, Technical Report, Revision 2.0, July 1995.

[P-D5.1B] *ACTS Prospect deliverable D5.1B: Operational plan for trial 2*, The Prospect consortium (AC052), November 1997.

[TINA-SA] *Service Architecture version 5.0*, C. Abarca, P. Farley, J. Forslöw, J. C. García, T. Hamada, P. F. Hansen, S. Hogg, H. Kamata, L. Kristiansen, C. A. Licciardi, H. Mulder, E. Utsunomiya, M. Yates, TINA-C, June 1997.

[TINA-SAA] *Service Architecture Annex version 5.0*, C. Abarca, P. Farley, J. Forslöw, J. C. García, T. Hamada, P. F. Hansen, S. Hogg, H. Kamata, L. Kristiansen, C. A. Licciardi, H. Mulder, E. Utsunomiya, M. Yates, TINA-C, June 1997.

[TIROP-98] Thanassis Tiropanis, *Offering role mobility in a TINA environment*, Proceedings of the 5th international conference on Intelligence in Services and Networks, IS&N ’98, Antwerp, Belgium, May 1998.

[M.3400] *CCITT recommendation M.3400*, October 1992.

[VITAL-V2] *ODTA Validation Second Trial*, European-ACTS project VITAL Deliverable, number AC003/SES/WP2/DS/P/D10/b1, VITAL, October 1997.