

Offering Role Mobility in a TINA Environment

Thanassis Tiropanis

University College London, Department of Computer Science
Gower Street, London WC1E 6BT
United Kingdom
t.tiropanis@cs.ucl.ac.uk

Abstract. This paper presents the concept of role mobility and the value of role mobility in an open service environment. We outline the aspects of role management and role mobility and we describe an open, reusable component for role management. Then, we show how this component can be deployed in a TINA environment with a minimum number of changes to the TINA service architecture.

1 Introduction

With the deregulation of the telecommunications market and the expected introduction of a new range of services to the customers, mobility is becoming increasingly important. The success of mobile telephony indicates a demand from the telecommunications customers for increased flexibility in terms of service personalisation and mobility.

The Telecommunications Information Network Architecture Consortium (TINA-C) has been deploying a Service Architecture (SA) that defines how a range of diverse and possibly complex services can be provided and managed in a uniform way. The starting point for the TINA SA is a set of separations that among other aims they attempt to allow for mobility.

Much of the work done on mobility to-date is concerned with personal and terminal mobility. That is the case for TINA [TINA-SA], ITU [F.851], ETSI and others. However, other kinds of mobility can be identified and tackled in a way similar to those for personal and terminal mobility. Role mobility is a problem that offers an interesting challenge to re-address the concept of mobility and approach it from a new perspective.

1.1 Concept of mobility

The term mobility describes the ability of an entity to change location without any disruption in the facilities that it offers or receives. The entity is usually obvious, as it is included in the definition of a kind of mobility; for example, in personal mobility the mobile entity is the person. The location of an entity can be more

obscure; for example, is it the geographical co-ordinates or the attached network access point that describes the location of a terminal? From the connectivity point of view, which is the main problem that mobility sets, it is the network access point that identifies the location of a mobile terminal, although the scope of a network access point can map to geographical co-ordinates.

In this context, personal mobility is the ability of a person to move to another terminal without any disruption in the provided telecommunications services. For example, the personal mobility that UPT [F.850] describes allows a telephony user to make and receive calls from a set of different telephones (terminals).

Similarly, terminal mobility is the ability of a terminal to move to another network access point without any disruption in the provided telecommunications services. In GSM systems for example, a mobile telephone (terminal) can move to another cell without any disruption in the offered services.

1.3 Concept of role mobility

In a telecommunications service environment, exchanges take place not only on a personal basis but also on a role basis. An example of a role-based exchange is that between a telephony user and the operator. In this kind of exchange the user does not wish to contact the specific person who is implementing the role of the operator at that time, but any person who implements the role of the operator, at that time. Therefore, the exchange really takes place between a person and a role. The persons who implement a role can vary but the service that they offer is virtually the same.

We assume that a role is an independent and well-defined virtual entity that can be carried by one or more persons at a time. Role mobility allows for a role to move to another person without any disruption in the service provided by the role. In role mobility, the mobile entity is the “role” and the location of the role is the “person” or “role-holder”.

1.4 Objectives

The objective of this work was to define the concept of role mobility and to specify a generic, reusable component to support it. Role mobility is a concept that applies to any service environment. We tried to design this component so that it can be used in any service environment.

Our first priority however, is to examine how this component can be integrated with the TINA business model [TINA-BM] and the TINA call-model with a minimum number of changes to the TINA service architecture [TINA-SA][TINA-SAA]. It is in our future plans to see how the concept of role mobility and the design of this component can fit in environments other than TINA.

In section 2 we look at the business aspects of role mobility and at the scope of role mobility on the enterprise level. Section 3 and section 4 describe information and computational aspects of a of role management component which provides

role mobility. Section 5 shows how role mobility is deployed in the ACTS project Prospect. Finally, some conclusions are drawn in section 6.

2 Enterprise aspects

We assume a case where a service provider (SP) is supporting a service for a number of users. Some of these users can implement a set of well-defined tasks like service administrator tasks, service help-desk tasks, or other. Each set of tasks describes a role. One or more persons (users) can carry out the role; those we call role-holders. A role must have the following qualities:

1. A role always describes a set of tasks and it cannot be an attribute that some users may possess. For example, “user aged under 30” is not a role. Also, it is the task that justifies a role as a single entity.
2. These tasks should always derive from the contractual relationship between the SP and the customer. This justifies the need for the management of roles and role mobility.

There can be two kinds of roles:

1. Service independent roles, for example “service administrator”.
2. Service dependent roles, for example a “private tutor”, for a Tele-education service.

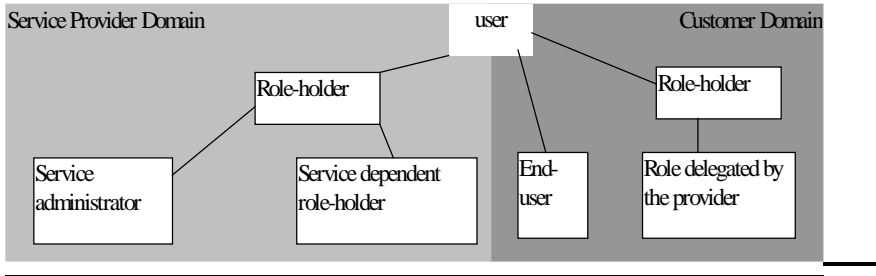


Fig. 1. Role mobility in the business model.

The scope of a role, service independent or not, is the domain of the service provider that provides that role. The possible locations of a role are limited among the users of that service. The term user includes all persons who use or manage a service, as shown in Figure 1. A role can reside in the service provider’s domain or in the customer’s domain. Since roles derive from the contractual relationship between the provider and the customer, the roles that reside on the customer domain represent tasks that can be delegated by the service provider to the customer.

Figure 1 shows the enterprise context of role mobility. A set of use cases is shown in Figure 2, in UML notation. The entities that interact in these use cases are:

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

We distinguish between two kinds of role-holders:

- *Active role-holder*: is a user who is carrying a role at a given point in time.
- *Inactive role-holder*: is a user who can carry a role but who is not active at a given point in time.

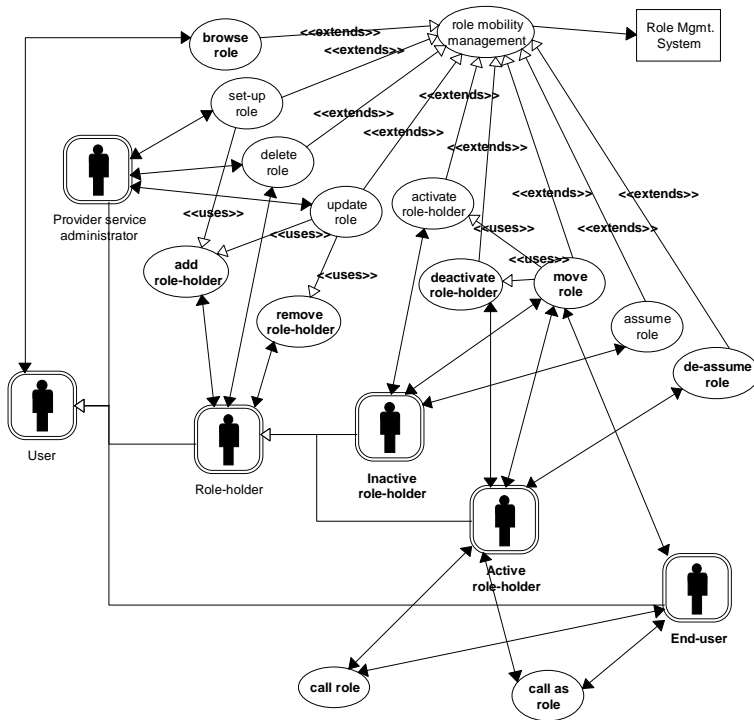


Fig. 2. Role mobility use cases.

The following use cases inherit from the *role mobility management* use case:

- *Browse roles*: describes the exchange were a listing of roles and their attributes is returned for all the roles in a service provider domain.
- *Set-up role*: describes the creation and initialisation of a role. An initial list of role-holders is set-up with the role.
- *Update role*: describes the modification of the profile of a role. For example, modifications to policies for the activation of role-holders. There are two use cases that inherit from this one:
- *Add role-holder*: describes the exchange between the role-management system and a user who is to be registered as a role-holder for a role.

- *Remove role-holder*: describes the exchange between the role-management system and a role-holder that is to cease being a role-holder for a role.
- *Delete role*: describes the deletion of a role from the role management system.
- *Activate role-holder*: describes the exchange between the role management system and an inactive role-holder in order for the later to become active.
- *Deactivate role-holder*: describes the exchange between the role management system and an active role-holder in order for the later to become inactive.
- *Move role*: describes the exchange between the role management system, an active role-holder and the users who are having a session with this role holder, in order for the role to move to another role-holder.
- *Assume role*: describes the exchange between the role management system and an inactive role-holder in order for the latter to become active.
- *De-assume role*: describes the exchange between the role management system and an active role-holder in order for the later to become inactive.

There are two call related use cases:

- The *call role* use case describes the exchange between an user/role-holder, a role-holder, the role management system, and other components of the call-model in order for an user/role-holder to contact a role.
- The *call as role* use case describes the exchange between a role-holder, an user/role-holder, the role management system, and other components of the call-model in order for a role to contact an user/role-holder.

The above use cases set the requirements analysis for the design and deployment of the role management component. The information and computational aspects which derive from this analysis are presented in the following two sections.

3 Information aspects

A role can be 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. It is the responsibility of the SP to make sure that a role ID is unique in the service provider domain. Some could argue that service independent roles, for example the service administrator, could have an ID that is unique for all the services. We believe that this is not a very good idea, because, as explained in the previous section, a role represents a set of tasks and it would be impossible for these tasks to be identical for all services. Apart from that, this would require a special authority to manage the allocation of role IDs.

```

typedef string t_roleId;
typedef string t_userId;
typedef t_userId t_roleHolder
typedef sequence <t_roleHolder> t_roleHolderList;

typedef string t_criterionType;
typedef string t_criterionValue;

enum PolicyAction {
    activate
    deactivate
    call          // also, call_randomly, call_ordered, etc ...
};

struct t_policyElement {
    t_criterionType type,          // e.g. time slot
    t_criterionValue value,      // e.g. <from-time>,<to-time>
    t_PolicyActionList action,    // e.g. activate, call, etc ...
    t_roleHolderList role_holders // e.g. roleholder A, roleholder C
};

typedef sequence <t_policyElement> t_policySpec;

struct role {
    t_roleId role_id,
    t_svcId service_id,          // e.g. the service the role is serving
    t_roleHolderList role_holder_list,
    t_roleHolderList active_role_holder_list,
    t_policySpec activation_policy,
    t_policySpec deactivation_policy,
    t_policySpec call_policy
};

```

Table 1. Role description.

A description of a role includes:

- The role ID.
- The list of the user IDs of the role-holders.
- The user IDs of the active role-holders.
- The policy for the activation/deactivation of a role-holder. For example, the time of day can determine when a role-holder becomes active.
- The policy for deciding which of the active role-holders will answer to a specific call. For example, the user ID of the caller can determine which of the active role-holders will take the call.

Table 1 gives the description of a role in IDL structures.

In a role environment, the call-model should provide for invitations that originate and/or terminate to a role. Since a role-holder is also a user of a service, the invitation should specify whether the role-holder is sending/receiving an invitation as a role or as a simple user. Also, a role-holder may receive an invitation to become active/inactive from the role management system. Table 2 shows these types of invitations in IDL structures.

```

typedef short t_callPartyType;
const t_callPartyType t_user=0;
const t_callPartyType t_role=1;

union t_callParty
    switch (t_callPartyType) {
        case t_user:
            t_userId user_id;
        case t_role:
            t_roleId role_id;
    };

struct t_invitation {
    t_callParty inviter,
    t_callParty invitee
};

```

Table 2. Information to support invitations and role-holder state management.

From this discussion we see that role mobility requires the following enhancements in the information models of the TINA service architecture:

1. An invitation should distinguish between role and user call-parties.
2. A role ID should be used in order to call a role.

4 Computational aspects

A role is a mobile entity, which is delegated for by a *Role Agent* (RA) on the computational level. The RA has all the information regarding that role as shown in Table 1: role ID, list of role-holders, policy for activating/deactivating/calling role-holders. The RA contacts User Agents (UA) for the following use cases: add/remove role-holder (*i_uaRoleHolder*), activate/deactivate role-holder and move role (*i_uaRoleHolder*). A RA is contacted by UAs in the assume/de-assume role use cases (*i_raUsrMgmt*). There is only one RA per role in a service provider domain.

The *Role Registrar* (RR) keeps track of all the RAs in a service provider domain. It manages the life cycle of a RA and it can give a list of all the roles provided by a service provider (*i_rrQuery*). The provider administrator user application interacts with the RR in the following use cases: set-up/update/delete role (*i_rrRoleMgmt*), browse roles (*i_rrQuery*). There is one RR per service provider domain.

The *Role Locator* (RL) is able to provide the location of the RA for a given role ID. It interacts with the UA in the call role use case and the role call use (*i_rlQuery*). The RR also uses the RL in order to register a new RA in the set-up role use case (*i_rlReg*). There is one RL per service provider domain.

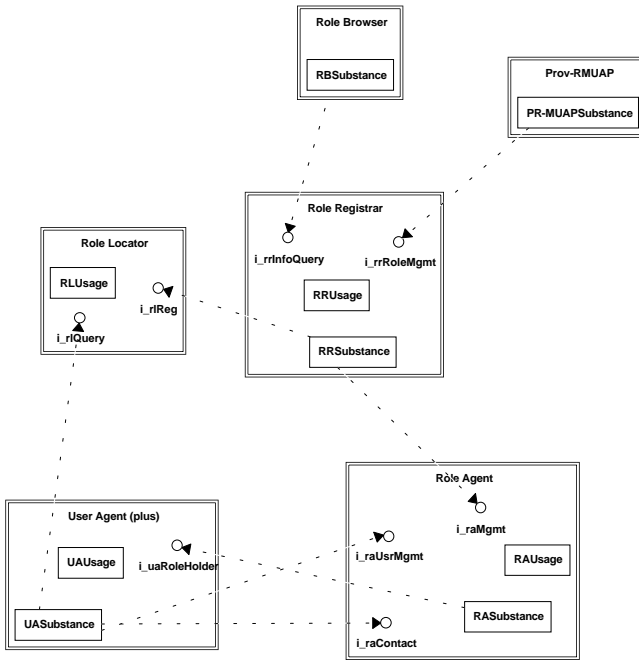


Fig. 3. Computational view of the role management system.

Figure 3 shows a computational model for the role management component, in UML notation. The rectangles denote computational objects; these map to UML categories. The circles denote interfaces. According to the TINA USCM model [TINA-SAA] a computational object (CO) has a usage part which supports the usage of a CO by other COs, a substance part which allows the CO to use other COs, a core part, and a management part. The usage and the substance parts of the COs also appear in the diagram.

The enhancements to the TINA architecture to support role mobility are:

1. User agents should be able to call or be called by role agents (*i_uaRoleHolder*).
2. User agents should be able to interact with the role locator.
3. The user agent of a role-holder should support the assume/de-assume role uses (*i_raUsrMgmt*).
4. The user agent of a role-holder should support the activate/deactivate role-holder use cases (*i_uaRoleHolder*).
5. The user agent of a role-holder should support the add/remove role-holder use cases (*i_uaRoleHolder*).

5 Deploying role mobility

In the European ACTS project Prospect, a subset of the use cases of section 2 is considered. A first assumption is that in most cases it is the end-user of a service that will have to contact a role. Therefore, the “call as role” use case is not addressed. Another assumption is that there will be no need to move a role from one active role-holder to another during a session between an end-user and the role. Therefore, the “move role” use case is not addressed in the Prospect environment.

Prospect is focused on inter-domain service management in the open service market [ISN-97]. Role mobility is considered a very interesting area from the management point of view. The enterprise model in Prospect involves a Tele-education service (TES) provider who employs a number of multi-media services in order to offer courses to remote students. The students who take the courses are the end-users of a customer organisation. The services used by the TES provider (TESP) include a multimedia conferencing service, a hypermedia service, and a web-store service.

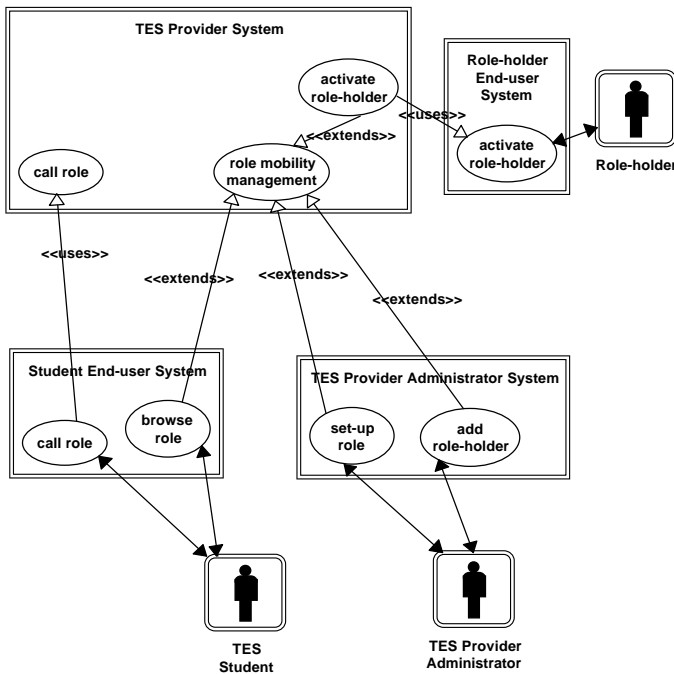


Fig. 4. The Prospect trial use cases in UML notation.

This environment has been demonstrated with a number of project trials [P-D5.1B]. Role mobility will be demonstrated in the last project trial, T2.2. Prospect

has followed and applied much of the TINA service architecture to the system model for service management and service control.

Prospect envisages one TES dependent role, the “tutor” role. After a course takes place, the students may wish to consult a tutor in private. The TES provider can have a number of different “tutor” role-holders active at a time. When a student contacts the “tutor” role, one of the active role-holders will be contacted and the call will be passed on to them.

The policy for the activation of role-holders is determined by the time of day. Each role-holder knows beforehand which hours of which days they may be activated. Also, when there is a big demand for the tutor role, extra role-holders can be.

The following use-cases are planned for the Prospect demonstration:

1. *Set-up role*: The TES provider administrator sets-up the “tutor” role. A number of role-holders is given, together with the role policies for activating and calling role-holders. The activation policy is based on the time-of-day. There is no calling policy; that means that when a call is made, an active role-holder is picked randomly.
2. *Add role-holder*: The role agent (RA) contacts the user agent (UA) of each role-holder and notifies them of their role-holdership.
3. *Activate role-holder*: The role agent (RA) activates a number of role-holders according to the time-of-day. In busy hours, the RA may decide to attempt to activate extra role-holders.
4. *Browse role*: A student browses the roles that are available by the TES provider. They eventually find that there is a “tutor” role.
5. *Call role*: The student makes a call to the tutor role. The UA of the student will receive the call and pass it to the appropriate RA. The RA will pick an active role-holder randomly and it will check if they are ready to take the call. If not, the RA will try another active role-holder or activate a new one. When an active role-holder is eventually available, the call will be passed to the UA of that role-holder.

The above use-cases also appear in Figure 4.

Prospect is developing a full role management component and a number of different user interfaces that interact with this component. The user interface for the TES provider administrator allows for browsing and managing roles. The user interface for the students allows for browsing and contacting roles.

6 Conclusions

The work that is presented in this paper attempts to model and tackle the issue of role mobility in a way similar to those for personal and terminal mobility. We consider this attempt successful. However, it is worth pointing out the similarities and the differences between role mobility and the other kinds of mobility in TINA. Starting with the similarities, we see that a role can be modelled as a

mobile entity like other, more familiar mobile entities: persons and terminals. A role has a location, which is a person. A role can be addressed by using a unique identifier, the role ID. There is a specific group of locations (persons) that can accommodate a role, the role-holders. On the computational level, a role agent (RA), who works in a way similar to that of the user agent (UA) or the terminal agent (TA), can delegate for a role.

The main difference between role mobility and the other kinds of mobility is that a role is a virtual entity which can exist in more than one location at the same time; more than one person can be active role-holders at a given time. This does not cause any significant complications to the design for two reasons. The first is that all the instances of a role are supposed to have the same behaviour. The second reason is that there is only one entity (the role agent) which decides what location a role can appear to have for each caller; the role still appears to exist on one location at a time for each specific session. Another difference is that a role, service independent or not, has a limited scope within the domain of a service provider. Also, a role is an entity that is by nature mobile. Since a role resides on a person, a person cannot be available at all times. When a person becomes unavailable, the role has to move to another person or suspend its operation.

As the number of services and service providers is expected to increase in the future, the demand for better support will increase too. This requires flexibility with identifying roles for each service and managing them in an efficient way. As shown, the role mobility component is generic enough to encompass a wide range of service independent and service dependent roles. Different service providers can reuse it and it can be easily integrated with the TINA architecture, subject to a minimum number of modifications in the information and computational models of TINA.

On the information level, a role ID should be used to identify a role. The invitation to a person should specify whether this person is contacted as a role-holder or a simple user. On the computational level, a user agent should be able to distinguish between a role invitee and a user invitee: a call to a role invitee should be routed to the corresponding role agent.

7 Further work

The implementation of the role management component is currently under development in Prospect; the component is lined-up for demonstration in the project trial in June 1998.

Acknowledgements

Part of this work was done under the auspices of the European ACTS project Prospect (contract number AC052). I would like to thank all the participants in the PCS activity in Prospect for their feedback and their support in this work. The views and ideas expressed in this document do not necessarily reflect those of the Prospect consortium.

References

1. [F.850] Principles of universal personal telecommunication (UPT), ITU-T standard F.850, ITU-T, 1993.
2. [ISN-97] *Inter-domain integration of services and service management*, D. Lewis, T. Tiropanis, A. McEwan, C. Redmond, V. Wade, R. Bracht, Proceedings of the fourth international conference on intelligence in services and networks, IS&N '97, Cernobbio, Italy, May 1997.
3. [P-D5.1B] *ACTS Prospect deliverable D5.1B: Operational plan for trial 2*, The Prospect consortium (AC052), November 1997.
4. [TINA-BM] *TINA Business Model and Reference Points version 4.0*, M. Yates, W. Takita, L. DemouDEM, R. Jansson, H. Mulder, TINA-C, May 1997.
5. [TINA-SA] *Service Architecture version 5.0*, C. Abarca, P. Farley, J. Forsl w, J. C. Garc a, T. Hamada, P. F. Hansen, S. Hogg, H. Kamata, L. Kristiansen, C. A. Licciardi, H. Mulder, E. Utsunomiya, M. Yates, TINA-C, June 1997.
6. [TINA-SAA] *Service Architecture Annex version 5.0*, C. Abarca, P. Farley, J. Forsl w, J. C. Garc a, T. Hamada, P. F. Hansen, S. Hogg, H. Kamata, L. Kristiansen, C. A. Licciardi, H. Mulder, E. Utsunomiya, M. Yates, TINA-C, June 1997.