Personal Mobility Support in Future Service Architectures

P.P.Demestichas, E.C.Tzifa, M.D.Louta, N.I.Liossis, A.K.Kaltabani, M.E.Anagnostou

National Technical University of Athens, Computer Science Division,
9 Heroon Polytechneiou Str, Zographou 15773 Athens, GREECE
Tel: + 30 1 772 25 29, Fax: + 30 1 772 25 34
email: pdemest@cc.ece.ntua.gr

Abstract: Support for personal mobility will be among the key factors for success in the competitive communications market of the future. This paper proposes enhancements to the personal mobility support capabilities of service architectures. The TINA service architecture is used as a reference, even though our approach is applicable to other models as well. Our starting point is a business case that falls into the realm of personal mobility. The aim of the business case is to enable users that are found outside their home domain to access services by choosing the best visited retailer, i.e., the one offering, adequate quality services in the most cost efficient manner. In the sequel the following key issues are addressed. First, the introduction of the additional functionality that is required for supporting the business case, and the realisation through appropriate service components. Second, the integration of the new service components in the standard TINA service architecture. Third, the detailed description of a version of the logic of the new components. In this last respect, we formally state, mathematically formulate and solve problems related to the visited retailer selection.

Keywords: Retailer, TINA, 0-1 linear programming, Mobile Intelligent Agents

1. INTRODUCTION

The ongoing liberalisation and deregulation of telecommunications will introduce new actors in the respective market of the future [1,2,3,4]. In principle, the main role of all the players in such a competitive environment will be to constantly monitor user demand, and in response to create, promote and provide the desired services and service features. Some key factors for success are described in the following. First, the efficiency with which services will be developed. Second, the quality level, in relation with the corresponding cost, of new services. Third, the provision of universal service access to the widest possible set of users, regardless of physical locations, terminals used and the inherent heterogeneity of communication systems. Fourth, the efficiency with which the services will be operated (controlled, maintained, administered, etc.). Fifth, the realisation of the services in a highly flexible way in order to adapt to changing customer service demands instantly.

The challenges outlined above have brought to the foreground several new important research areas. Some of them are the definition of new business models [3,4], the specification of advanced service architectures (SAs) [5,6,7,8,9,10,11], the development of advanced service creation environments (SCEs) [12,13,14,15,16], the elaboration on the personal mobility concept [6,17,18,19], and the introduction of advanced software technologies, e.g. distributed object computing [20,21] and intelligent mobile agents [22,23,24,25].

The aim of this paper is to propose enhancements to the personal mobility support capabilities of legacy service architectures. Personal (as opposed to terminal) mobility may be defined as the ability of telecommunications consumers to use services that are personalised with their preferences and identity ubiquitously, independently of both physical location and specific features of the equipment used. An additional aim is to introduce the enhancements by minimally impacting the existing service architectures. Our reference service architecture will be the one specified by the Telecommunications Information Networking Architecture Consortium (TINA-C) [3,7,8,9,26,27,28]. However, the presented practices may be applied to other models as well.

Our approach in this paper is the following. The starting point is a presentation of assumptions regarding the entities and the roles at the business level in the telecommunications world of the future. Then, we expose the role of some of the presented business level entities, in a target *business case*, i.e., an important service provision scenario that should be supported in an open competitive communication environment. Through the business case presentation, the importance of the personal mobility and service architecture concepts are illustrated. Finally, the use and expansion of legacy service architectures, so as to support the advanced personal mobility features of the business case, is discussed. Expansions refer to the following. First, the identification of the additional functionality required, so as to fully support the target business case, and its subsequent attribution to new service components, so as to minimally impact the existing service architecture. Second, the integration among the new and the already existing service architecture (TINA) components. Third, the detailed presentation of a version of the logic of the new components (however as it will be discussed, other versions of the logic of the new components may be considered and incorporated).

A typical view of the competitive telecommunications world of the future may be the one depicted in Figure 1. Without being exhaustive four main different entities can be identified, namely, the *user*, *service provider*, *retailer*, and *connectivity provider*. The role of the service provider is to design and implement services. The role of the retailer is to promote, sell, customise, deliver and operate these services. Limited by techno-economic or administrative reasons each retailer offers services only inside a *domain*. A user may have subscription contracts for specific services with one or more retailers (*home retailers*). Through a subscription, a user acquires the right to use a service in a retailers' domain (*home domain*). Finally, the role of a connectivity provider is to offer the network connections necessary for supporting the services.

Such highly competitive and open environments should encompass mechanisms that will enable the users to find and associate with the most appropriate retailers, i.e., those offering, at a given period of time, adequate quality services in a cost efficient manner. Hence, the following business case that falls into the realm of personal mobility may be pursued: to enable users to access services outside their home domains through the best visited retailer (*visited retailer selection*).

The realisation of the identified business case requires a significant amount of co-operative computing among the involved business level entities (e.g., user, home and visited retailers). These interactions are described in a high level manner in section 2. Service architectures provide an initial framework wherein such complex computations may be encapsulated. Hence, in section 3 the relevant elements of the TINA framework are revisited. Our focus will be on the access session of the TINA service architecture and on the framework specified for basic personal mobility support [6,17,18].

Section 4 introduces the extensions necessary for the support of the advanced personal mobility concepts imposed by the business case, and explains how they can be integrated with the TINA framework. In this section extensions are specified in terms of the (high level description of the) new functionality needed, the corresponding service components, and software technologies. Section 5 presents in finer detail a version of the logic of the new components that are required for the visited retailer selection. In particular, we define, mathematically formulate and solve an optimisation problem that may be used in the evaluation of the quality of each retailer offer. From that point, the best, visited retailer will be determined by the performance with respect to the objective function of the optimisation problem. Section 6 provides a set of indicative results. Finally, section 7 includes future plans and some concluding remarks.

2. BUSINESS CASE: SELECTION OF THE BEST VISITED RETAILER

It is assumed that a user is roaming out of his home domain and wishes to make use of a specific service. While in a foreign domain, the user accesses a terminal that can support the service and makes a request for the particular service usage through the *terminal software* (TSW). We assume that various *candidate retailers* (CRs) may serve the terminal (Figure 2). Therefore, an interesting option that should be offered to the user is the selection of the best candidate visited retailer (for the specific service and the related user preferences).

It is assumed that a *default retailer* (DR) undertakes the responsibility of interacting with the user, until the candidate retailer selection is completed. The default retailer may be determined either automatically (e.g. by means of terminal mechanisms), or by user intervention. Moreover, it is assumed that the home retailer (HR) maintains the necessary information for user identification, as well as the user profile and service subscription data.

Enabling the service usage through the most appropriate retailer requires that the three general phases depicted in Figure 3 are conducted.

The aim of the first general phase is user authentication. To this end, co-operation between the home retailer and the default retailer is required. This process will not be further addressed in this paper.

The second general phase is the core of the retailer selection. For this reason, the home retailer is invited to apply the relevant mechanisms, so as to select on behalf of the user the best retailer for the service at hand. It may be assumed that the home retailer determines the set of candidate retailers by means of a pertinent service (which may be based on location information, federation agreements, etc.). The basis for the retailer selection comprises the user preferences, requirements and constraints, as they are designated by the personal profile information and the relevant service subscription data, the list of eligible retailers, and the retailer offers (e.g., cost at which the desired quality levels are provided). The selection is service specific and can be a sophisticated process undertaken by an agent. While in principle the agent could be provided by a third party, it is rather obvious that the data used by this agent, are maintained by the service retailer.

In the third phase the result of the selection is available, and hence an association, and consequently a service usage, may start between the user and the selected retailer.

In the next section we revisit some relevant elements of the TINA framework, which will be the basis for hosting the computations required by the business case presented in this section. Section 4 presents the extensions to the TINA framework that are necessary for fully supporting the business case presented in this section. Section 5 describes in more detail a version of the functionality of the new components.

3. ELEMENTS OF THE TINA FRAMEWORK

A fundamental TINA concept is that of the *session*. It was introduced for depicting the temporary relationship among a group of objects that are assigned to collectively fulfil a task for a period of time. Four session types have been defined, namely *access*, *service*, *communication*, and *connectivity* (Figure 4) [6]. The access session is the gateway to any specific service usage. In other words, through an access session a user may request or be invited to services. The service session provides the context, in which service execution (usage), control and management takes place. The communication session provides the means of controlling the communication connections, which are required during the service session. A connectivity session serves as an environment for controlling the actual network connections.

Relevant to this paper is the TINA access session, which is a service independent segment that acts as a gateway to any service specific usage. It comprises capabilities that allow user authentication, basic personal mobility support [6,17,18] and service invocation. The basic structure of the TINA access session is depicted in Figure 5.

Relevant to the access session are the *access session User Application* (as-UAP), the *Provider Agent* (PA), the *Initial Agent* (IA), the *User Agent Home* (UAH) and the *User Agent Visited* (UAV) components. The as-UAP resides in the terminal domain and models the entity through which the user conducts the access session. The PA represents the retailer in the terminal (user) domain. The IA is the initial access point to a domain, and supports capabilities for the user authentication. The UAH and UAV components represent the user in the home and visited retailer domains, respectively. Their role is to intercept and process user requests. The UAH and UAV have subordinate objects (SOs) that maintain user specific information. For brevity, these objects are not shown in full detail in Figure 5. The information held in these objects is the user subscription data (e.g. services, for which there is a subscription), the user profile (e.g., service preferences, conditions for service provision), etc. In principle, the components in the visited retailer domain are assumed to have subset capabilities and contents with respect to their peers in the home domain. Moreover, it is assumed that the UAH is created when the first subscription is made and is deleted when all subscriptions are terminated. A UAH maintains (or may acquire through the subordinate components) the subscription information and personal user information, part of which is used to initialise the UAV.

The TINA access session components invoke the *Service Factory* (SF) when specific service usage requests are granted. The SF is responsible for the creation of the *User Session*

Manager (USM) and *Service Session Manager* (SSM), which are the other basic s*ervice session* related components.

4. INTRODUCTION OF ADVANCED PERSONAL MOBILITY FEATURES

This section introduces the extensions to the TINA access session model that are necessary for fully supporting the functionality required by the business case of section 2. Initially, the new functionality required will be described in a high level manner in sub-section 4.1. This functionality will be attributed to service components in sub-section 4.2. Finally, integration issues among the new components and the TINA service architecture will be provided in sub-section 4.3. A more detailed description of a version of the functionality of the new components is presented in section 5.

~~1.1~~4.1 Requirements : Identification of additional functionality

Μορφοποιημένο: Κουκκίδες και αρίθμηση

The TINA access session offers the framework for user authentication and service invocation. The novel feature that is not supported is the overall task of retailer selection. As a first step, this aspect requires an entity that will act on behalf of the user. Its role will be to capture the user preferences, requirements and constraints regarding the requested service, to deliver them in a suitable form to the appropriate retailer entity, to acquire and evaluate the corresponding retailer offers, and ultimately, to select the most appropriate retailer.

As a second step, the retailer selection requires an entity that will act on behalf of the retailer. Its role will be to collect the user preferences, requirements and constraints and to make a corresponding offer, taking also into account the underlying connectivity providers. The overall object model is depicted in Figure 6. The retailer management mechanisms that are indicated in Figure 6 are standard means through which the retailer-related selection components may be managed.

4.2 Analysis : Definition of new components

In this sub-section we define the components that will undertake the functionality identified in the previous section. New components are defined, in order to comply with the requirement to introduce the functionality by minimally impacting the service architecture.

In general, the behaviour of the new components should possess some generic properties, which are also typical of the software entities that are characterised as (intelligent mobile) agents [22]. The first such property is *autonomy*, which is the ability of the component to act autonomously on behalf of an authority, which in our case is the user or the retailer. Autonomy is justified by the potential complexity associated with the retailer selection problem, which may make user intervention infeasible or undesirable. Complexity may also be tackled by the *intelligence* property. The third desirable property is *social behaviour*, since the new components will have to interact not only among them but also with the existing, standard TINA components. Finally, the *mobility* property, even though not essential, may provide some performance advantages as discussed in [29].

Based on the discussion above the following components are defined and designated with the functionality that was identified in the previous sub-section. First, the *Subscribed User Agent* (SUA), whose role is to select on behalf of the user the most appropriate retailer for the service at hand. Second, the *Retailer Agent* (RA), whose role is that of promoting the services offered by the retailer. More specifically, the SUA obtains the user preferences, requirements and constraints from the user profile and the service subscription data, interacts with the RAs, so as to obtain the corresponding retailer offers, and selects the most appropriate retailer for the desired service. Moreover, it may be extended to record past experience (e.g., retailer selections, service usage characteristics, etc.). The RA promotes the offers of the retailer, interacts with SUAs, and the underlying connectivity provider mechanisms.

As will be shown in the next sub-sections the SUA will be considered as a component that is also attributed with the mobility property. The core of the SUA functionality is not affected from this choice. However, this choice has been made so as to be as general as possible with respect to the SUA capabilities.

4.3 Design : Integration in the service architecture

This sub-section describes the integration of the new components with the TINA service architecture. This is accomplished by providing a formal description of the business case and the interactions among the involved service components.

~~1.1.1~~4.3.1 Detailed description of the business case

Μορφοποιημένο: Κουκκίδες και αρίθμηση

The interactions that take place among the involved service components are described below and depicted in Figure 7.

It is assumed that a user who has a subscription to a service, accesses a terminal, supplies personal identification data (e.g., User_Id, Password) and issues a request for the specific service usage, through the UAP and the PA that reside in the terminal (*Step 1*). The PA forwards this request to the IA of the default retailer (IA_Default) (*Step 2*) which, in the sequel, uses the user's identification data so as to determine the user's home retailer i.e., UAH (*Step 3*). Thereafter, authentication is conducted through the co-operation of the default and home retailer, and in particular of the IA_Default and UAH (*Step 4*). After the successful authentication, the IA_Default creates a UA_Default together with its SOs (*Step 5*) and furthermore invokes the UA_Default in order to initialise itself (*Step 6*). The UA_Default is similar to the UAV introduced in section 3. The outcome of this phase is an access session establishment between the default and home retailers (equivalent to that depicted in Figure 5).

Following, the PA forwards the service usage request in the default retailer domain, and in particular, to the UA_Default (*Step 7*). Eventually, the UAH processes the request and identifies that the SUA-based mechanism should be initiated (*Step 8*). Therefore, the UAH is invited to create the SUA and to initiate it with the end user's service subscription data and the personal profile information (*Step 9*).

The SUA uses a service (e.g., brokerage service based on location information, federation agreements, etc.) in order to retrieve the list of eligible candidate retailers (*Step 10*). Then, SUA replicas are produced and are assigned to interact with a particular RA. Since we have assumed that the SUA is also attributed with the mobility property, the SUAs may be sent to the nodes of the candidate retailers where they will interact with the local RAs (*Step 11*). The overall model of SUA and RA interaction is depicted in Figure 8. A version of the logic that underlies the interactions among the SUA and the RA will be presented in section 5. In particular we will present a scheme for obtaining and evaluating the quality of each retailer offer.

At this point there is a need for an entity that will assess all the offers that have been made by the retailers, and ultimately, select the best retailer. In this respect, we assume that the results of the interactions between each SUA replica and RA are sent to the SUA residing in the default retailer domain (*Step 12*). This "parent" SUA is responsible for selecting (on behalf of the user) the retailer that makes the best offer. This decision is stored in the SUA along with the results of the negotiation and the object reference of the selected retailer (*Step 13*).

Hereafter, the access session with the default retailer will be released (if the default retailer is not selected) and an access and a service session will be established between the selected and home retailer. The SUA residing in the default retailer domain requests from the PA to start a service session in the selected retailer domain (*Step 14*). For that reason, the PA sends a corresponding request to the IA_Selected (*Step 15*). The authentication process for the end-user takes place between IA_Selected and UAH (*Step 16*). Afterwards, the IA_Selected creates a UA_Selected together with its SOs and invokes the UA_Selected to initialise itself (*Step 17*). The IA_Selected completes the establishment of an access session by returning a reply to the PA. The PA requests from the UA_Selected to start a service session providing also the Service_id. The UA_Selected analyses the PA's request and also requests the SF to create the SSM and the USM for the specific service session. For brevity the last few steps are not presented in Figure 7.

4.3.2 Service component interactions in the context of the business case

In this sub-section we describe in more detail the service component interactions that take place in the business scenario. These interactions are also depicted in Figure 9(a)-(c).

1. A user that has a subscription to a service accesses a terminal, and through the as-UAP, supplies personal identification data, and issues a request for using the service. The request is forwarded to the PA. The arguments passed with the request are the service identifier (Service_id), user identifier (User_id) and user password.

2. The PA sends an Access_Session_Request() event to the IA_Default. The request includes the arguments received from the as-UAP.

3. The IA_Default uses a naming service to get an object reference for the end user's UAH. The object reference (UAH_ref) is retrieved.

4. Authentication process for the end-user takes place between the IA_Default and the UAH.

5. After the successful authentication the IA_Default creates a UA_Default (i.e., a UAV in the default domain) together with its subordinate objects. The IA_Default invokes the created UA_Default to initialise itself. The initialisation includes the User_id and object references for the PA (PA_ref) as well as for the user's UAH (UAH_ref).

6. The IA_Default returns a reply to the PA. This reply includes the object reference for the created UA_Default (UA_Default_ref) and the object reference for the UAH (UAH_ref).

7. The PA requests to the UA_Default to start a new service session of the given Service_id. It is assumed that the Service_id indicates that the selection of the best (most appropriate) retailer is desired (i.e., the service identifier serves also as an <optimise service> flag or command).

8. The UA_Default analyses the PA's request and informs the UAH.

9. The UAH identifies that the retailer selection procedure should be initiated (through the service identifier that plays the role of the <optimise service> flag). The SUA is created in the Home Domain and the list of candidate retailers is formed.

10. The list of candidate retailers is exploited, so as to obtain the RA references and the addresses of the candidate retailer nodes to which the SUA replicas should migrate, in order to interact with the RAs.

11. SUA replicas learn the object references obtained in the previous step and migrate to the identified nodes.

12. The SUA and the RA in each candidate Visited Retailer Domain interact. The aim is to obtain a retailer offer for the user preferences, requirements and constraints regarding the specific service. A version of the logic that underlies the interactions between the SUA and the RA will be presented in section 5.

13. The results of the negotiations between each SUA replica and the RA of each candidate Visited Retailer Domain along with the object references of IA_candidate (IA_candidate_ref) are sent to the SUA residing in the Default Retailer Domain.

14. The SUA residing in the Default Retailer Domain makes a decision on the most appropriate retailer. The reference of the selected retailer is denoted as IA_Selected_ref.

15. The SUA residing in the Default Retailer Domain requests the PA to terminate the access session established with the UA_Default.

16. The SUA residing in the Default Retailer Domain requests the PA to start a service session with the selected retailer. This request includes the object reference for the IA in the retailer's domain that was finally selected (IA_Selected_ref).

17. The PA sends an Access Session Request to the IA_Selected. The request includes the arguments received from the SUA, the user identifier (User_id) and the user password.

18. The IA_Selected uses a naming service to get an object reference for the end user's UAH. The object reference (UAH_ref) is retrieved.

19. Authentication process for the end-user takes place between IA_Selected and UAH.

20. The IA_Selected creates a UA_Selected (i.e., a UAV in the selected retailer domain) together with its subordinate objects. The IA_Selected invokes the created UA_Selected to initialise itself. The invocation request includes the end user identity (User_id) and object references for the PA (PA_ref) as well as for the end user's UAH (UAH_ref).

21. The IA_Selected completes the establishment of an access session by returning a reply to the PA. This reply includes the result (i.e., a success or fail) as well as an object reference for the created UA_Selected (UA_Selected_ref) and the object reference for the UAH (UAH_ref).

22. The PA retains the received object reference.

23. The PA requests to the UA_Selected to start a service session. The service identifier that is supplied now (Service_id_opt) corresponds to the same service but indicates that the optimisation mechanisms have been completed.

24. Since there is no <optimise service> flag (command) in the arguments the SF is requested to create the SSM, USM and all the other relevant objects. From this point on a service session exists.

5. RETAILER SELECTION PROBLEM

In this section we describe in more detail a version of the logic that underlies the SUA and RA interactions. As already presented SUA replicas interact with the RA of each candidate retailer (Figure 8). The aims of the SUA-RA interactions are the following. First, to supply to the RA the user preferences, requirements and constraints regarding the specific service. Second, to obtain the corresponding retailer offers. Third, to select the retailer that makes the best offer.

The tasks outlined require a method that will enable the assessment of the quality of the offer of each retailer. In this respect, in this section we provide the following. First, in sub-sections 5.1-5.4, we define and solve a problem that may be used for evaluating the quality of the retailer offer. The overall algorithm for determining the best retailer is presented in sub-section 5.5.

5.1 Evaluating the quality of an offer

As already explained each SUA replica acts on behalf of a user u of a given class (i.e., with a known profile and service subscription data). User u wishes to use a given service s.

A fundamental assumption at this point is that service s is composed of a set of distinct service features (e.g., see Figure 10). Furthermore, let us assume without loss of generality that these service features are offered (supported) by the candidate retailer to this user (according to the service subscription data and the profile). Each service feature has an associated set of possible quality levels. The service subscription data of the user and the retailer policies determine each of these quality level sets. Moreover, there is a cost associated with the provision of a service feature at a given quality level.

An objective function, which models the quality of the retailer offer, consists of the following factors: The benefit, which stems from the provision of a service feature and the cost, at which each retailer offers a service feature at a given quality level.

The constraints of the problem refer to the following. First, each service feature should be assigned to an allowable quality level. Second, there may be an upper bound on the overall price that the user may afford during the service usage. Likewise, the third constraint may be associated with a lower bound on the overall user satisfaction.

As an example consider the model of Figure 10: A given user wishes to access a service. The service consists of four service features, each offered at three quality levels. It may be envisaged that a cost is imposed by the retailer for the provision of a service feature at an associated quality level. The service subscription data and the profile of the user indicate that the user is interested in 3 out of the 4 service features. Moreover, these service features may be offered to the user in only 2 of the 3 allowable quality levels. A benefit (measure of the user satisfaction) will be derived from the provision of a service feature at an associated (allowable) quality level.

Taking into account the aspects outlined above, the most suitable retailer for user u is the one that makes the best offer, i.e., the one that yields the best solution to the following problem.

Problem 1: [Evaluation of the Quality of the Retailer Offer]. Given:

(a) a user u who wants to use a service s,

(b) the profile of user-u and the service subscription data,

(c) the service features that compose service-s (these service features are designated by the service subscription data and may be supported by the retailer),

(d) the quality levels at which each service feature may be offered,

(e) the benefit (user satisfaction) deriving from the provision of a service feature at an allowable quality level,

(f) the cost imposed by the retailer for the provision of a service feature at an associated (allowable) quality level,

(g) the upper bound on the overall price of the service usage that the user may afford,

(h) the lower bound on the benefit (satisfaction) that the user wants to experience during the service usage,

find, the best service configuration pattern, i.e., assignment of each service feature to an allowable quality level that optimises an objective function, which is associated with the user satisfaction and the price of the service usage, subject to the constraints imposed by the upper price bound and the lower user satisfaction bound.

Sub-section 5.2 provides a formal statement of the problem of evaluating a retailer offer. In sub-section 5.3 this problem is formulated as a 0-1 linear programming problem [30,31,32]. In sub-section 5.4 we present a heuristic solution to the problem.

5.2 Formal problem statement

Let u be the user who wants to use a given service- s through the most suitable retailer. The comprised service features of service s will be denoted as $SF(s)$. Among these service features, of interest to the user are those that belong to the service subscription data of the user profile and will be denoted as $SF(u,s)$. It holds that $SF(u,s) \subseteq SF(s)$.

For each service feature i ($i \in SF(s)$) the respective quality levels, at which it may be used, are represented by the set $Q(i)$. The set of quality levels that are in line with the service subscription data of the user profile is denoted by $Q(u,i)$. It holds that $Q(u,i) \subseteq Q(i)$. Assuming that service feature- i is provided at quality level- j, the benefit and cost associated with this assignment will be represented as $b_{SQ}(i,j)$ and $p_{SQ}(i,j)$ ($i \in SF(u,s)$, $j \in Q(u,i)$), respectively.

The objective of our problem is to find an assignment A_{SQ} of service features i ($i \in SF(u,s)$) to quality levels j ($j \in Q(u,i)$) that maximises an objective function $f(A_{SQ})$. Among the terms of this function there is the overall benefit (user satisfaction from the assignment), which is expressed by the function $b(A_{SQ})$, and the cost $p(A_{SQ})$, at which the retailer will provide the assignment.

The constraints of our problem are the following. First, each service feature- i ($i \in SF(u,s)$) should be assigned to only one quality level- j ($j \in Q(u,i)$). Second, the constraint regarding the total cost should be preserved. Therefore, the total cost should not exceed a predefined value $p_{\max}$ representing the maximum price that may be afforded by the user for the service usage. Therefore, the condition $p(A_{SQ}) \leq p_{\max}$ should hold. Third, the overall satisfaction that the user should experience should not be lower than a given value $B_{\min}$, i.e., $b(A_{SQ}) \geq B_{\min}$.

Concentrating again on the example that is illustrated in Figure 10 we may observe the following: (i) $SF(s) = \{sf_1, ..., sf_4\}$; (ii) $SF(u,s) = \{sf_2, sf_3, sf_4\}$; (iii) $Q(u, sf_2) = \{q_{22}, q_{23}\}$, etc.

The overall problem can be formally stated as follows.

Problem 1: [Evaluation of the Quality of the Retailer Offer].

Given a user u, who wants to use service s through the most suitable retailer, the profile and the service subscription data of user u, the set of service features $SF(u,s)$ that service s comprises and at the same time are of interest (relevant) to user u, the set of quality levels $Q(u,i)$ at which each service feature i may be offered and are of interest (relevant) to user u, the benefit $b_{SQ}(i, j)$ and cost $p_{SQ}(i, j)$ associated with the assignment of service feature i ($i \in SF(u,s)$) to quality level j ($j \in Q(u,i)$), the upper bound on the price $p_{\max}$ that the user may afford, and the lower bound $B_{\min}$ on the benefit (user satisfaction) that the user wants to experience during the service usage, find an assignment of service features to quality levels A_{SQ} that optimises an objective function $f(A_{SQ})$ that is related to the overall user satisfaction $b(A_{SQ})$ and price $p(A_{SQ})$ suggested by the assignment, under the constraints $p(A_{SQ}) \le p_{\max}$, $b(A_{SQ}) \ge B_{\min}$, and that each service feature is assigned to exactly one quality level.

5.3 Optimal Formulation

In this section, the problem above is formulated as a 0-1 linear programming problem. In order to describe the assignment A_{SQ} of service features to quality levels, we introduce the decision variables $x_{SQ}(i, j)$ ($i \in SF(u,s)$, $j \in Q(u,i)$) that take the value 1(0) depending on whether the service feature-i is (is not) assigned to quality level-j. The problem of obtaining the most appropriate assignment A_{SQ} may be obtained by reduction to the following optimisation problem.

Problem 1: [Evaluation of the Quality of the Retailer Offer].

Maximise:

$$c_B \cdot \sum_{i\in SF(u,s)} \sum_{j\in Q(u,i)} b_{SQ}(i, j)\cdot x_{SQ}(i, j) - c_P \cdot \sum_{i\in SF(u,s)} \sum_{j\in Q(u,i)} p_{SQ}(i, j)\cdot x_{SQ}(i, j) \quad (1)$$

subject to

$$\sum_{j\in Q(u,i)} x_{SQ}(i, j) = 1 \qquad \forall i \in SF(u,s) \quad (2)$$

$$\sum_{i\in SF(u,s)} \sum_{j\in Q(u,i)} b_{SQ}(i, j)\cdot x_{SQ}(i, j) \ge B_{\min} \quad (3)$$

$$\sum_{i\in SF(u,s)} \sum_{j\in Q(u,i)} p_{SQ}(i, j)\cdot x_{SQ}(i, j) \le p_{\max} \quad (4)$$

Relation (1) expresses the objective of finding the best assignment of service features to quality levels that maximises the cost function, which is associated with the overall benefit (user satisfaction), and the corresponding price. Weights c_B and c_P provide the relative value of the benefit (user satisfaction) related part and the price related part of the objective function. Constraints (2) guarantee that each service feature will be assigned to exactly one quality level. Constraint (3) guarantees that the level of user satisfaction will not be lower than a pre-defined value. In the same manner, constraint (4) guarantees that total cost will not exceed a predefined value.

5.4 Computationally efficient solution

This section discusses computationally efficient solutions for the problem of evaluating the quality of the retailer offer that is addressed in this paper. In general, there may be a significant amount of computations associated with the optimal solution of problem 1. In this respect, the design of computationally efficient algorithms that may provide good (near-optimal) solutions in reasonable time is required. Classical methods in this respect are simulated annealing [33], genetic algorithms [34], taboo search [35], greedy algorithms [31], etc. Hybrid or (user-defined) heuristic techniques may also be devised. Finally, exhaustive search should be conducted in case the solution space is not prohibitively large. In this section we briefly outline the importance of exhaustive search and describe an algorithm that is based on simulated annealing.

5.4.1 Exhaustive search

Exhaustive search should be conducted in case the solution space is not prohibitively large. These algorithms apply a method for scanning all the solution space. The cost of the respective solutions is evaluated and finally, the best solution is maintained. The complexity of the search in our case is $\prod_{i \in SF(u,s)} |Q(u,i)|$, i.e., a function of the service features that are relevant to the user and the quality levels at which these service features may be offered.

In the results section there are examples in which the complexity of exhaustive search is reasonable.

5.4.2 Algorithm based on Simulated Annealing

Such algorithms are required in case the solution space is prohibitively large to be scanned in an exhaustive manner.

During each phase of an algorithm that is based on the simulated annealing paradigm, a new solution is generated by minimally altering the currently best solution (in other words, the new solution is chosen among those that are “neighbouring” to the currently best one). If the new solution improves the objective function value (i.e., the difference between the objective function value of the old and the new solution, Δc, is negative) the new solution becomes the currently best solution. Solutions that decrease the objective function value may also be accepted with probability $e^{-(\Delta c/CT)}$ (Metropolis criterion). This is a mechanism that assists in escaping from local optima. CT is a control parameter, which may be perceived as the physical analogous of the temperature in the physical process. The algorithm ends when either $CT = 0$ (temperature reaches 0) or when a significant number of moves have been made without improving the cost function.

The development of a simulated annealing-based procedure means that the following aspects have to be addressed: configuration space, cost function "neighbourhood" structure and cooling schedule (i.e., manner in which the temperature will be reduced). The configuration space is the set of feasible solutions $x_{SQ}(i, j)$ that satisfy the constraints (2)-(4). The cost function is the one introduced by relation (1).

The neighbourhood structure of a solution is produced by reallocating a service feature-i from its current quality level-j to another randomly chosen (higher or lower) quality level-j'. The cooling schedule may be calculated according to $T' = r \cdot T$, where T is the temperature and r is usually a number that ranges from 0.95 to 0.99.

The simulated annealing-based algorithm may be described as follows.

Basic Simulated Annealing Algorithm.

Step 0. Initialisation. Get an initial solution, IS, and an initial temperature value T. The currently best solution (CBS) is IS, i.e., $NBS = IS$, and the current temperature value (CT) is T, i.e., $CT = T$.

Step 1. If $CT = 0$, or if the stop criterion is satisfied, the procedure ends and a transition to *step 6* is performed.

Step 2. A new solution (NS) that is neighbouring to CBS is found.

Step 3. The difference of the costs of the two solutions, CBS and NS is found, i.e., the quantity $\Delta c = C(CBS) - C(NS)$ is computed.

Step 4. If $\Delta c \leq 0$ then the new solution becomes the currently best solution, i.e., $CBS = NS$. Otherwise, if $\Delta c > 0$, then if $e^{-(\Delta c/CT)} > rand[0,1)$, the new solution becomes the currently best solution, i.e., $CBS = NS$.

Step 5. The cooling schedule is applied, in order to calculate the new current temperature value CT, and a transition to *step 1* is performed.

Step 6. End.

There are various alternatives for realising the stop criterion mentioned in *step 1*. In our version, the algorithm stops when no improvement in the objective function has been achieved after a given number of temperature decreases (in other words consecutive moves or alterations of the currently best solution). Neighbouring solutions (*step 2*) are selected randomly among all the neighbouring ones of the currently best solution, with the same probability for all neighbours.

5.5 Outline of SUA and RA capabilities

In this section we describe the algorithm, on which the SUAs and the RAs base accomplishment of their tasks. The algorithm should be seen as a more detailed description of the tasks in steps 9-14 in sub-sections 4.3.1 and 4.3.2, taking also into account the schemes of this section.

Step 1. The "parent" SUA is created and initialised with the preferences, requirements and constraints of user u regarding service s. These are expressed by the following data. First, the set of the service features $SF(u,s)$ that are of interest (relevant) to the user. Second, for each service feature i ($i \in SF(u,s)$) the corresponding set of allowable quality levels $Q(u,i)$. Third, the values $b_{SQ}(i,j)$ that describe the benefit (user satisfaction) stemming from the provision of service feature-i at quality level-j ($j \in Q(u,i)$). Fourth, the upper limit on the price $p_{\max}$ and the lower limit on the user satisfaction $B_{\min}$ that the user may afford, or wants to experience, respectively, during the service usage.

Step 2. The "parent" SUA obtains the list of candidate retailers, the addresses of the corresponding nodes, to which the SUA replicas should migrate, and the references of the RAs. A SUA replica is created for each candidate retailer.

Step 3. The SUA replicas migrate to the candidate retailer domains and the parent SUA migrates to the default retailer domain.

Step 4. Each SUA obtains the retailer offer for the user preferences, requirements and constraints regarding service-s. These are expressed by the prices $p_{SQ}(i,j)$ associated with the provision of service feature-i at quality level-j.

Step 5. Each SUA replica evaluates the quality of the retailer offer by solving the appropriate instance of problem 1. The result is sent to the parent SUA.

Step 6. The parent SUA in the default retailer domain selects a retailer by comparing the objective function values that each retailer has scored.

6. RESULTS

In this section, some indicative results are provided in order to assess the proposed software framework, which allows the incorporation of advanced personal mobility features in the TINA SA. In general, the scope of our paper is to augment the personal capabilities of legacy SAs. To this end, a business case was defined which is composed of three phases, namely user authentication, retailer selection and association establishment. More specifically, the contribution of this paper lies in the following areas. First, the realisation of a business case which required the introduction of new service components and the integration with the existing SA. Second, the definition, mathematical formulation and optimal as well as computationally efficient solution of the evaluation of the quality of the retailer offer problem that should be solved in the context of the retailer selection phase.

The results of this section aim at the provision of indicative evidence of the overall retailer selection scheme, with respect to a random retailer selection scheme. In the rest of this section two sets of experiments will be used for demonstrating these aspects. The experiments are differentiated from specific assumptions concerning user preferences and not from their focus which is as described above.

In order to test the performance of the proposed framework of this paper we assumed the existence of a terminal that falls into the domains of N candidate retailers. Users access this terminal in order to initiate a service usage. In the context of our experiments, we assume that the users request always the same service, which is video conference. We have chosen a simple and well-known service in order to explain more effectively the proposed scheme.

Video conference comprises two service features, namely audio and video. In the context of our study, four quality levels have been considered for these service features. Specifically, the quality levels that have been defined for audio correspond to 8kbits/sec, 16kbits/sec, 32kbits/sec and 64kbits/sec, respectively. In a similar manner, the defined quality levels for video correspond to 15frames/sec, 20frames/sec, 25frames/sec and 30frames/sec, respectively.

Regarding the different users that access the terminal, we assume the existence of k user classes. In the definition of these user classes we have also assumed that all users in these classes are interested for both service features. However, each user class involves different quality levels for these service features.

Concerning the implementation issues of our experiment, the whole TINA access session has been implemented in Java [36] (see also [4]). The OrbixWeb CORBA compliant platform [37] was used for the inter-component communication. The implementation of the intelligent, mobile agents, i.e., SUA and RA, was based on the use of the Voyager platform [38]. The particular platform was chosen, since it is a robust agent platform that has been successfully used in other applications. Furthermore, it supports all functional and technical requirements identified for implementing the identified business case.

In the first set of experiments, the user profile and data subscription presented in Table 1(a) for each defined user class, assuming that $k = 10$ was adopted. More specifically, in this table we indicate the quality levels, i.e., QA_i and QV_j ($1 \le i, j \le 4$) for audio and video, respectively, which are of interest to each user class. The user satisfaction of providing a service feature at a given quality level, i.e., co-efficients $b_{SQ}(i, j)$, for each user class is taken to be equal for all allowable quality levels. In this respect, the users of each class are equally attracted by all the quality levels at which a service feature, which is of interest to them, may be offered. Hence, the objective of the retailer selection scheme is reduced to the minimisation of the cost at which the service features will be provided.

In the same manner, in Table 1(b) the retailer policies regarding the features (audio and video) of videoconference and the respective quality levels is described, assuming that $N = 5$. More specifically, this table indicates the offered quality levels, for audio and video, for each retailer, as well as the cost (expressed in arbitrary cost values) which is associated with the provision of a service feature at a given quality level. It is noted that in the considered test case, all retailers do not offer all possible quality levels. Furthermore, there is a non uniform increasing rate at the cost, at which each retailer provides a service feature at a quality level.

As previously mentioned, the objective of our experiment is to provide indicative evidence of the overall retailer selection scheme, with respect to a random retailer selection scheme. In this respect, first, in Table 2 we present the outcome of the application of the retailer selection scheme for the user classes and retailer policies described in Table 1. Specifically, for each user class, the derived value of objective function (1) as well as the selected retailer is shown. It is noted that the co-efficients $b_{SQ}(i, j)$ are taken to be equal to 50. Moreover, with reference to Table 1, from the derived results we may conclude that the desired service features are always offered at the lowest allowable quality level. This is justified since (a) the users are equally attracted by the different quality levels and (b) the considered allocation yields minimum cost.

Following, we examine the effectiveness of the overall retailer selection scheme with respect to the random retailer selection scheme. In this respect, in Figure 11 a comparison between the cost of providing the desired service features at allowable quality levels when the retailer selection scheme is applied and the respective cost deriving from a random retailer selection, for each user class, is provided. Again, the user profiles and retailer policies described in Table 1, have been considered. From the obtained results we observe a decrease, which is up to 25%, in the overall cost when the retailer selection scheme is applied. This decrease is due to the selection of the most suitable retailer taking into account the user preferences and the retailer policies.

Regarding retailer specialisation, in Figure 12 we present the percentage of the requests that were handled by the retailers for the different user classes. From the obtained results we observe that a great number of requests was handled by retailer-5 because of its suitability for adequately serving 5 out of the 10 user classes.

In the second set of experiments, we have again adopted the user profile and data subscription shown in Table 1(a). However, in this test case we have assumed that the user satisfaction of providing a service feature at a given quality level, i.e., co-efficients $b_{SQ}(i, j)$, for each user class is not equal for all allowable quality levels. In this respect, in Table 3 for each user class we show the different coefficients, which indicate the level of attraction for all the quality levels at which a service feature, which is of interest to them, may be offered. Furthermore, from Table 3, we may conclude that, in principle, higher quality levels are more attractive to the users. This is expressed by the higher $b_{SQ}(i, j)$ co-efficients. Hence, the objective of the retailer selection scheme is to find the assignment that maximises the value of objective function (1). Regarding the retailer policies, the ones shown in Table 1(b) are also valid in this set of experiments.

Similarly to the first set of experiments, in Table 4 we present the outcome of the application of the retailer selection scheme for the user classes described in Table 3 and retailer policies described in Table 1(b). More specifically, we present, for each user class, the derived value of objective function (1), the selected retailer and the quality levels at which the service features (audio and video) were provided. Comparing to the corresponding results in the first set of experiments (Table 2), we may observe that the value of objective function (1) is increased up to 5%, approximately, for user class k_1. The increase is due to the higher attraction level for a specific quality level. For example, for user class -k_1, QA_2 is equal to 55 while in the first set of experiments the respective factor is taken equal to 50.

Apart from the above observations, we may also stress that for the user classes k_2, k_5 and k_8 different retailers have been selected comparing to the first set of experiments. This is, again, owing to the introduction of different levels of attraction for different quality levels. This introduction entails the selection of a different retailer, which offers a better balance between user satisfaction and derived cost.

In Figure 13 a comparison the retailer selection scheme and a random retailer selection is provided. In this respect, we present the value of objective function (1) in case the retailer selection scheme and the random retailer selection scheme are applied. Again, the user classes described in Table 3 and retailer policies described in Table 1(b) have been used. From the obtained results we observe an increase, which is approximately up to 3%, in the value of the objective function (1) when the retailer selection scheme is applied. The increase is due to the selection of the most suitable retailer for the specific user preferences and the retailer policies.

Regarding retailer specialisation, in Figure 14 we present the percentage of the requests that were handled by the retailers for the different user classes. From the obtained results we may observe that a great number of requests was again handled by retailer-5 because of its suitability for adequately serving 5 out of the 10 user classes. However, comparing to the first set of experiments, a more uniform allocation of the load (handled requests) is observed since our assumptions on user preferences results in more active involvement of the rest of the retailers.

7. CONCLUSIONS AND FUTURE WORKS

This paper proposed enhancements to the personal mobility support capabilities of legacy, standardised service architectures. The TINA service architecture was used as a reference, even though our approach is applicable to other models as well. Our starting point was the elaboration on a business case that falls into the realm of personal mobility. The aim of the business case is to enable users that are found outside their home domain to access services through the best visited retailer, i.e., the one offering, adequate quality services in the most cost efficient manner. The following key issues were addressed. First, the introduction of the additional functionality that is required for supporting the business case, and the realisation through appropriate service components. Second, the integration of the new service components in the standard TINA service architecture. Third, the detailed description of a version of the logic of the new components. In this last respect, problems related to the best-visited retailer selection problem were formally state, mathematically formulated and solved.

Directions for future work include, but are not limited to the following. First, the migration from simulation-based studies that were conducted for this paper, to the realisation of wide scale trials, so as to experiment with the applicability of the framework presented herewith. Second, the experimentation with various versions of the SUA and RA functionality.

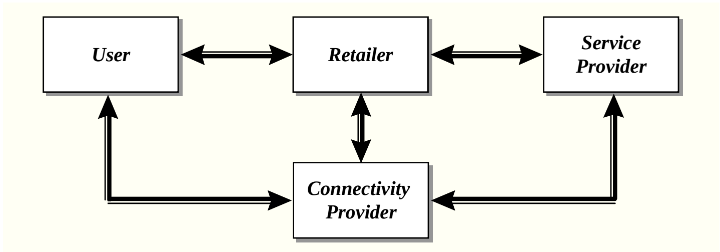


Figure 1. A view of the business level entities in the future competitive telecommunications environment

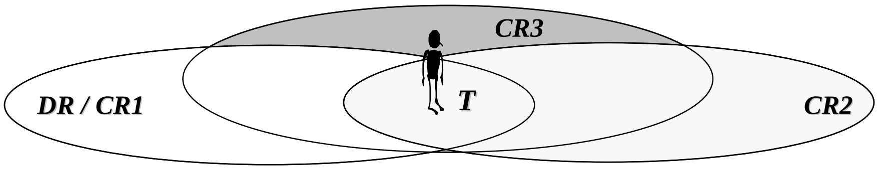


Figure 2. A user accesses a terminal in a foreign domain, i.e., outside his home domain, and wishes to make use of a service; the terminal falls into the domain of various candidate retailers. Candidate retailer 1 (CR1) is also the default retailer (DR)

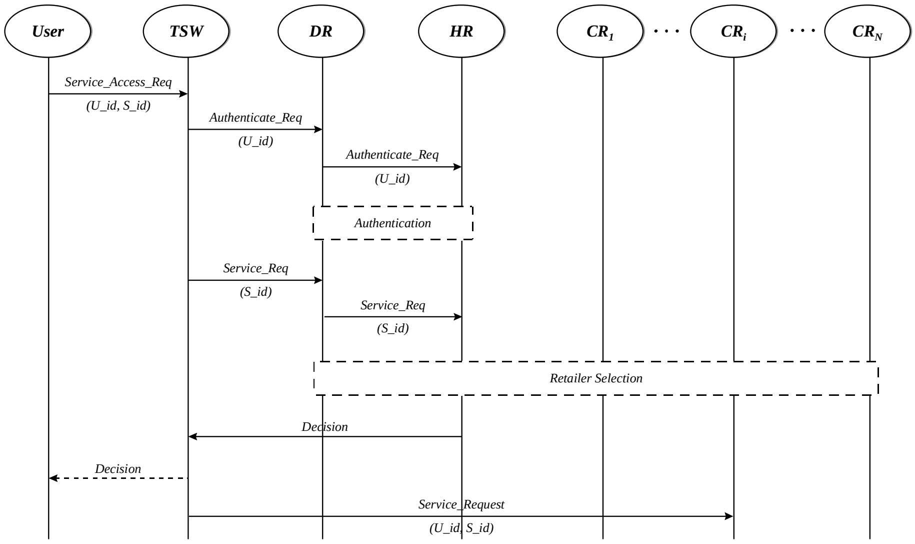


Figure 3. Interactions among the business level entities during the best visited retailer selection business case

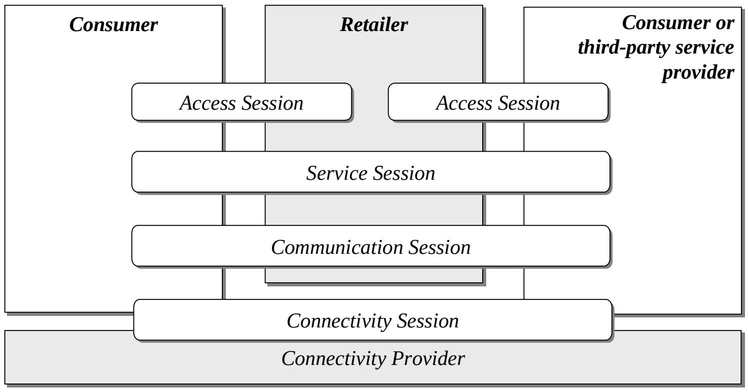


Figure 4. Session concept in TINA and relation with business roles

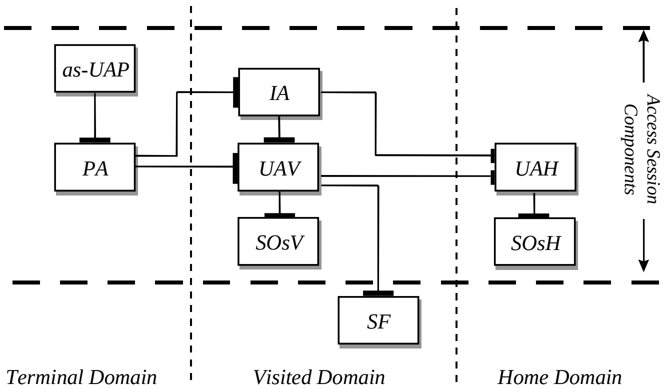


Figure 5. TINA access session attributed with basic personal mobility support capabilities

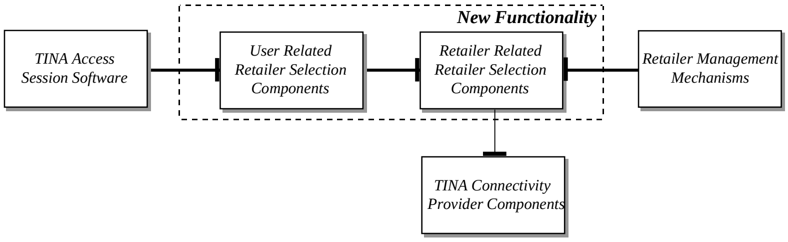


Figure 6. High level representation of the new functionality that should be defined for fully supporting the best visited retailer selection business case

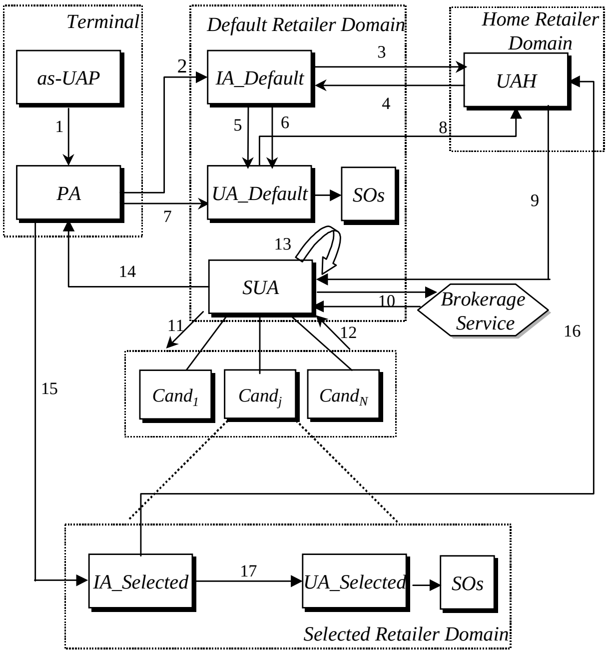


Figure 7. Interactions among service components during the business case

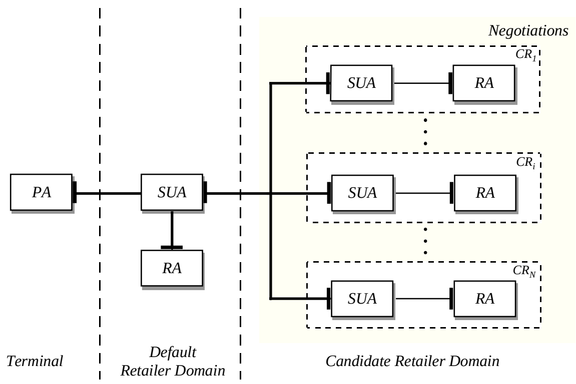
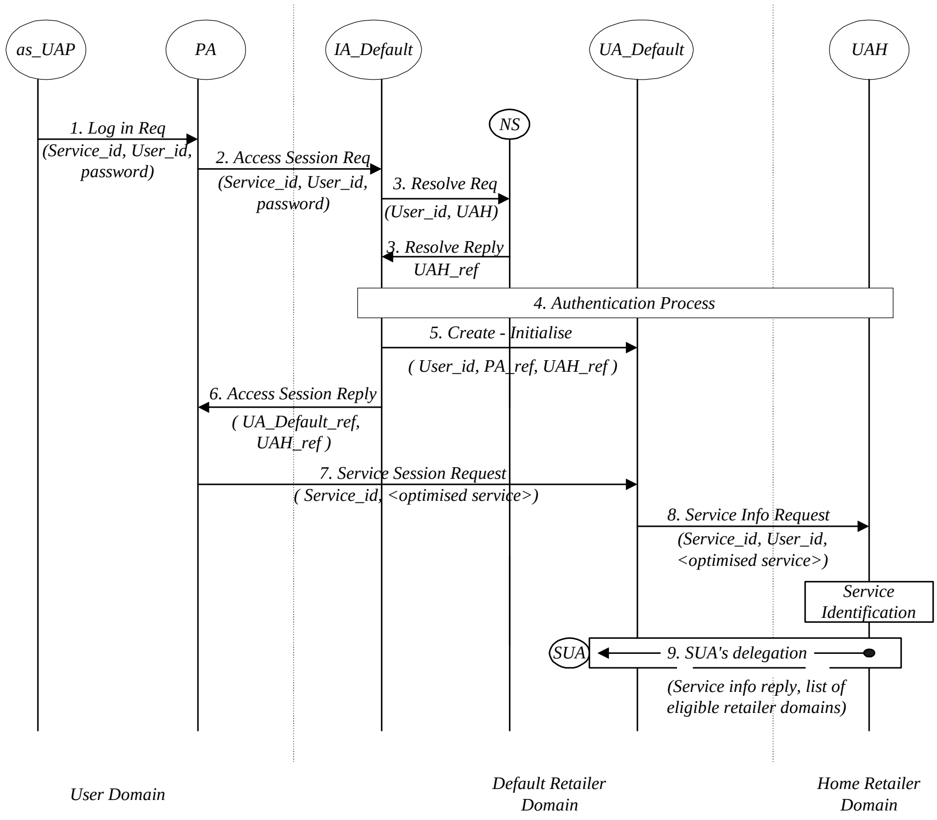
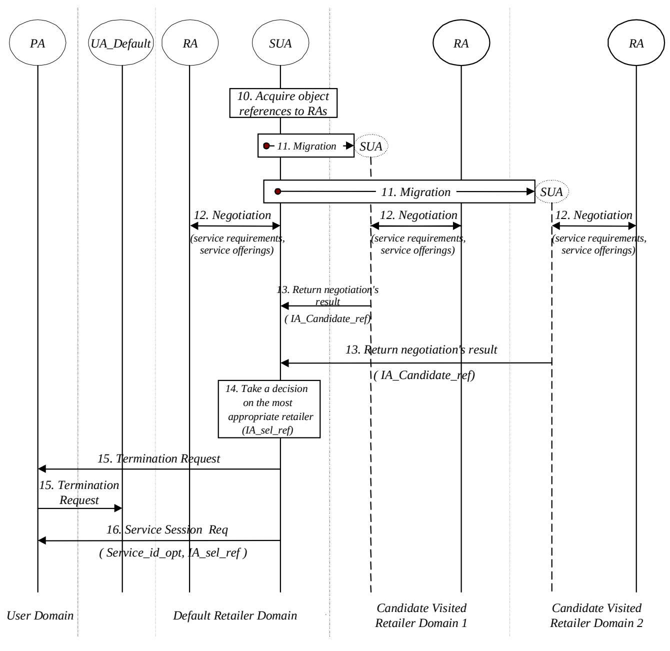


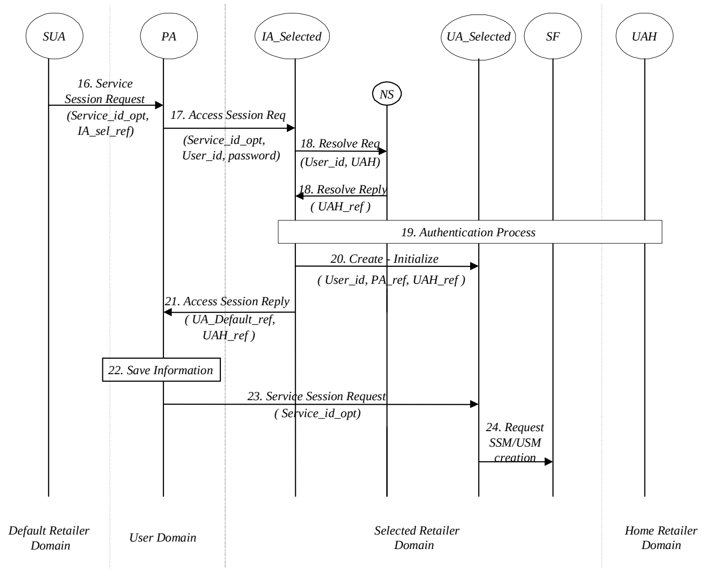
Figure 8. Interactions among the SUAs and the RAs


as_UAP
PA
IA_Default
UA_Default
UAH
NS
1. Log in Req
(Service_id, User_id, password)
2. Access Session Req
(Service_id, User_id, password)
3. Resolve Req
(User_id, UAH)
3. Resolve Reply
UAH_ref
4. Authentication Process
5. Create - Initialise
(User_id, PA_ref, UAH_ref)
6. Access Session Reply
(UA_Default_ref, UAH_ref)
7. Service Session Request
(Service_id, <optimised service>)
8. Service Info Request
(Service_id, User_id, <optimised service>)
Service Identification
SUA
9. SUA's delegation
(Service info reply, list of eligible retailer domains)
User Domain
Default Retailer Domain
Home Retailer Domain

(a)


PA
UA_Default
RA
SUA
RA
RA
10. Acquire object references to RAs
11. Migration
SUA
11. Migration
SUA
12. Negotiation
(service requirements, service offerings)
12. Negotiation
(service requirements, service offerings)
12. Negotiation
(service requirements, service offerings)
13. Return negotiation's result
(IA_Candidate_ref)
13. Return negotiation's result
(IA_Candidate_ref)
14. Take a decision on the most appropriate retailer (IA_sel_ref)
15. Termination Request
15. Termination Request
16. Service Session Req
(Service_id_opt, IA_sel_ref)
User Domain
Default Retailer Domain
Candidate Visited Retailer Domain 1
Candidate Visited Retailer Domain 2

(b)



(c)

Figure 9. Interactions during the business case among the new and the standard TINA components; (a) Steps 1-9; (b) Steps 10–16; and (c) Steps 16–24

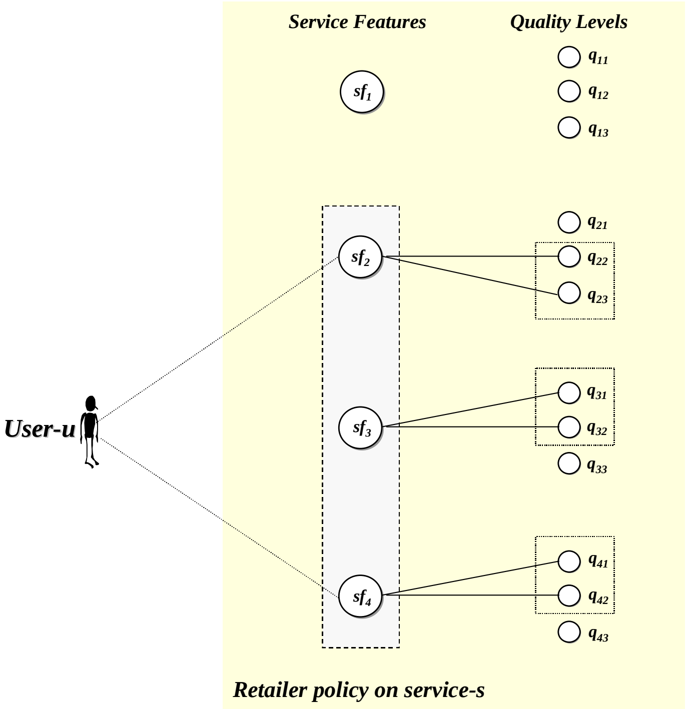


Figure 10. User-u wishes to access service-s. The service is composed of 4 service features, and each service feature may be offered at 3 quality levels, according to the retailer policy. The service subscription data and the profile of the user indicate that the user is interested in 3 of these service features. Moreover, these service features may be offered to the user in only 2 of the 3 allowable quality levels

User Class	***QA*** $_1$	***QA*** $_2$	***QA*** $_3$	***QA*** $_4$	***QV*** $_1$	***QV*** $_2$	***QV*** $_3$	***QV*** $_4$
k_1	✓	✓			✓	✓		
k_2		✓	✓			✓	✓	
k_3	✓				✓	✓		
k_4	✓	✓				✓		
k_5			✓	✓		✓	✓	✓
k_6		✓					✓	✓
k_7				✓	✓	✓		
k_8	✓	✓	✓				✓	
k_9	✓						✓	
k_{10}	✓	✓	✓	✓	✓	✓	✓	✓

(a)

Retailer	***QA*** $_1$	***QA*** $_2$	***QA*** $_3$	***QA*** $_4$	***QV*** $_1$	***QV*** $_2$	***QV*** $_3$	***QV*** $_4$
N_1	**1**	**1.5**	**3**	**5**	**2**	**3**	**5**	**7**
N_2	**0,9**	**1,2**			**2**	**2,8**	**4**	
N_3	**1**	**1,4**	**2,5**		**1,8**	**2,5**		
N_4	**1**	**1,4**	**2,5**		**2**	**2,8**	**4**	
N_5	**0,9**	**1,2**			**1,8**	**2,5**		

(b)

Table 1: First set of experiments. Description of (a) user profile and service data subscription and (b) retailer policy

User Class	*Objective Function Value*	*Retailer Selected*
k_1	97,3	5
k_2	96,3	5
k_3	97,3	5
k_4	96,6	5
k_5	95	3
k_6	94,8	2
k_7	93	1
k_8	95,1	2
k_9	95,1	2
k_{10}	97,3	5

Table 2: First set of experiments. Outcome of the retailer selection scheme application for each user class

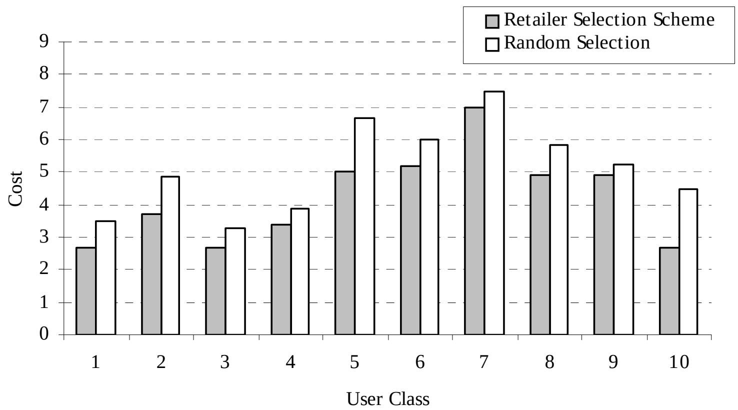


Figure 11: First set of experiments. Comparison of the retailer selection scheme and the random retailer selection scheme

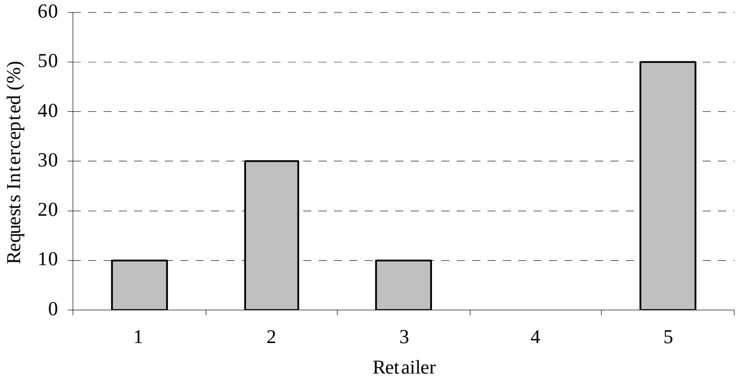


Figure 12: First set of experiments. Specialisation of the retailers with respect to the interception of requests for the various user classes

User Class	***QA$_1$***	***QA$_2$***	***QA$_3$***	***QA$_4$***	***QV$_1$***	***QV$_2$***	***QV$_3$***	***QV$_4$***
k_1	50	55			50	50,5		
k_2		50	51,5			50	50	
k_3	50				50	50,5		
k_4	50	51				50		
k_5			50	51		50	53	53
k_6		50					50	51
k_7				50	50	53		
k_8	50	50,3	53				50	
k_9	50						51	
k_{10}	50	50,3	51,5	50	50	51	51	53

Table 3: Second set of experiments. Description of user profile and service data subscription

User Class	*Objective Function Value*	*Retailer Selected*	*Assigned QA_i*	*Assigned QV_j*
k_1	102	5	QA_2	QV_1
k_2	96,5	**3**	QA_3	QV_2
k_3	97,3	5	QA_1	QV_1
k_4	97,3	5	QA_2	QV_2
k_5	96,5	**4**	QA_3	QV_3
k_6	94,8	2	QA_2	QV_3
k_7	95	1	QA_4	QV_2
k_8	96,5	**4**	QA_3	QV_3
k_9	96,1	2	QA_1	QV_3
k_{10}	97,6	5	QA_2	QV_2

Table 4: Second set of experiments. Outcome of the retailer selection scheme application for each user class

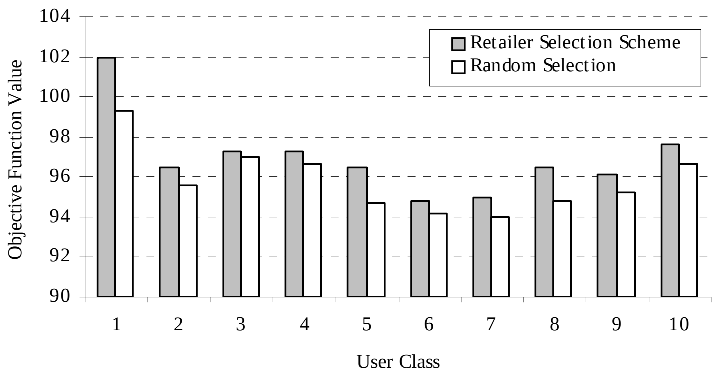


Figure 13: Second set of experiments. Comparison of the retailer selection scheme and the random retailer selection scheme

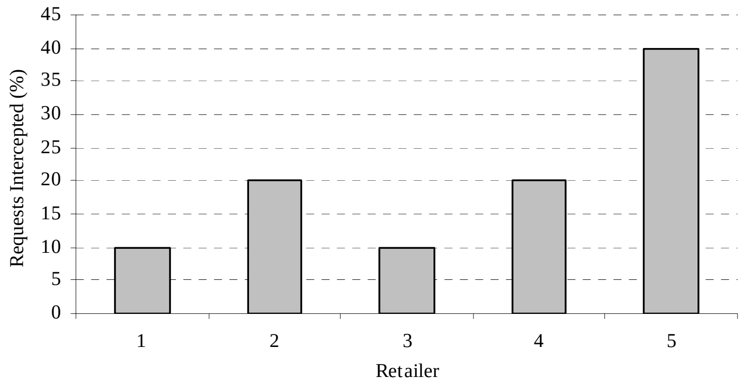


Figure 14: Second set of experiments. Specialisation of the retailers with respect to the interception of requests for the various user classes

8. References

[1] H. Ungerer, N.P. Costello, "*Telecommunications in Europe: Free choice for the user in Europe's 1992 market*", 2nd ed., Office for Official Publications of the European Communities, Brussels, 1990

[2] H. Zuidweg, M. Campolargo, J. Delgado, A. Mullery, (Eds.), "*Paving the way for an open service market; 6th Conf. on Intelligence in Services and Networks (IS&N'99)*", Springer-Verlag, Berlin, 1999

[3] TINA-C Deliverable, "*Definition of service architecture*", Version 5.0, June 1997

[4] ACTS project MONTAGE (AC325), CEC deliverable D25, "*Design of advanced agent-based accounting and charging and personal mobility support services*", April 1999

[5] S. Trigila, A. Mullery, M. Campolargo, J. Hunt, "Service architectures and service creation for integrated broadband communications", *Computer Communications*, Vol. 18, No. 11, 1995

[6] S. Trigila, K. Raatikainen, B. Wind, P. Reynolds, "Mobility in long-term service architectures and distributed platforms", *IEEE Personal Commun*., Vol. 5 No. 4, Aug. 1998

[7] T. Magedanz, "TINA - Architectural basis for future telecommunications services", *Computer Communications*, Vol. 20, No. 4, 1997

[8] W.J. Barr, T. Boyd, Y. Inoue, "The TINA initiative", *IEEE Commun. Mag.*, Vol. 31, No. 3, March 1993

[9] F. Dupuy, G. Nilsson, Y. Inoue, "The TINA Consortium: Toward networking telecommunications information services", *IEEE Commun. Mag*., Vol. 33, No. 3, Nov. 1995

[10] A. Lazar, "Programming telecommunication networks", *IEEE Network*, Vol. 11, No. 5, Sept./Oct. 1997

[11] A. Lazar, K.-S. Lim, F. Marconcini, "Realising a foundation for programmability of ATM networks with the binding architecture", *IEEE J. Selec. Areas Commun.*, Vol. 14, No. 7, Sept. 1996

[12] M. Tag, "Service creation environment engineering", In Proc. *Interworking'96 Conf.*, Nara, Japan, 1996

[13] ACTS project SCREEN (AC227), CEC Deliverable D28, "*SCREEN engineering practices for component based service creation*", Dec. 1998

[14] ACTS project SCREEN (AC227), CEC Deliverable D38, "*Advanced service creation environment*", Aug. 1998

[15] P.P. Demestichas, F. Westerhuis, M. Guyot, C. Loftus, E. Sherratt, M. Anagnostou, "Service creation environment for open distributed service architectures", In Proc. *IEEE International Conference on Intelligence in Networks 1998* (*ICIN'98*), Bordeaux, France, May 1998

[16] E.C. Tzifa, P.P. Demestichas, M.A. Lambrou, A. Kaltabani, S. Kotrotsos, M.E. Anagnostou, "Advanced service creation environment: Practices, implementation and application", *Technologies for the Information Society: Development and Opportunities*, J.-Y.Roger, B.Stanford-Smith, P.T.Kidd, (Eds.), IOS press, Amsterdam, 1998

[17] B. Wind, M. Samarotto, P. Nicosia, M. Lambrou, E. Tzifa, "Enhancing the TINA architectural framework for personal mobility support", In Proc. *5th Conf. On Intelligence in Services and Networks 1998* (*IS&N'98*), Antwerb, Belgium, May 1998
[18] ACTS project DOLMEN (AC036), CEC deliverable ASD4, "*Open Service Architecture for Mobile and Fixed Network Environment (OSAM)*", Final release, Aug. 1998
[19] ITU-T Recommendation F.581, "*Universal Personal Telecommunications (UPT)* - *Service Description Service Set 1*", Geneva, Switzerland, 1992
[20] S. Vinoski, "CORBA: Integrating diverse applications within distributed heterogeneous environments", *IEEE Commun. Mag*., Vol.35, No. 2, Feb. 1997
[21] Object Management Group (OMG), "*The Common Object Request Broker: Architecture and specification*", Version 2.0 (Updated Revision 2), July 1996
[22] "Mobile software agents for telecommunications", Feature topic in the *IEEE Commun. Mag.*, Vol. 36, No. 7, July 1998
[23] P. Morreale, "Agents on the move", *IEEE Spectrum*, Vol. 35, No. 4, April 1998
[24] "Active and programmable networks", Special issue in the *IEEE Network*, May/June 1998
[25] "Active networks", Feature topic in the *IEEE Computer*, April 1999
[26] TINA-C Deliverable, "*TINA business model and reference points*", Version 4.0, May 1997
[27] TINA-C Deliverable, "*Service component specification*", Version 1.0, Jan. 1998
[28] TINA-C Deliverable, "*Network resource architecture*", Version 3.0, Feb. 1997
[29] A. Kind, M.E. Anagnostou, M.D. Louta, J. Nicklisch, J. Tous, "Towards the Seamless Integration of Mobile Agents into Service Creation Practice", in Proc. *6th Conf. on Intelligence in Services and Networks* (*ISN'99*), April 1999
[30] H.M. Salkin, "*Integer programming*", Addison-Wesley, Reading, Massachusetts, 1975.
[31] C. Papadimitriou, K. Steiglitz, "*Combinatorial optimization: Algorithms and complexity*", Prentice Hall, Inc., 1982
[32] M.R. Carrey, D.S. Johnson, "*Computers and intractability: A guide to the theory of NP-completeness*", W.H. Freeman, San Fransisco, 1979.
[33] E. Aarts, J. Korts, "*Simulated annealing and the Boltzmann machines*", J. Wiley & Sons, New York, 1989
[34] Z. Michalewicz, "*Generic algorithms + Data structures = Evolution programs*", Springer-Verlag, Berlin, 1995
[35] F. Glover, E. Taillard, D. de Werra, "A User's Guide to Taboo Search", *Annals of Operations Research*, Vol. 41, pp. 3-28, 1993
[36] J. Gosling, B. Joy, G. Steele, "*The Java Language Specification*", Addison-Wesley, 1996
[37] IONA Technologies, "*OrbixWeb Programmer's Guide*", Nov. 1997
[38] ObjectSpace Voyager, "*Core Technology User Guide*", Version 2.0, Dec. 1997