“Scenario”-adaptivity for e-service management in heterogeneous networks

A. Iera, A. Molinaro, A. Pudano, and D. Ursino
Dept. D.I.M.E.T. – Universit`a “Mediterranea” di Reggio Calabria
Via Graziella, Localit`a Feo di Vito, 89060 Reggio Calabria, Italy.
E-mail: {iera, antonella.molinaro, alfredo.pudano, ursino}@ing.unirc.it

Abstract— **A “Situation&Location” aware system has the main task of handling the set of distributed information and resources available within heterogeneous networks to provide a user with seamless and optimized network access to the desired service from any location. Situation&Location and its whole set of variables can be a powerful means to trade-off among user requirements, application constraints, and underlying network conditions. We propose a Multi Agent System able to capture the concept of Situation&Location and perform the task highlighted above. We follow a general-purpose approach with the aim of guaranteeing the highest versatility is possible. The main strength of our proposed system is its “adaptability” that can be achieved without the use of either new terminals or novel technologies. The only requirement is the introduction of a middleware functionality into already existing terminals and into some network servers.**

Index Terms — Context-awareness, Intelligent Agents, user modeling, QoS adaptivity, heterogeneous wireless systems

I. INTRODUCTION

In the last few years research and development activities of telecommunications and computer science communities have centered on the provision of customized services, also called *e-services*. These services can be regarded to as a collection of network-resident software programs that collaborate for supporting users in both accessing and selecting data and services to satisfy their needs in every-day life. It is then clear that the *ubiquitous* access to e-services is crucial.

An ubiquitous system [2] has to allow users to access the desired service *anywhere* and *anytime*. This concept has been somehow considered in current third-generation (3G) networks with the Virtual Home Environment (VHE) concept. VHE provides service personalization and “is a capability whereby a user is offered the same service experience in a visited network as in his home system” [3]. “The VHE ensures that users are consistently presented with the same personalized features, whatever terminal, wherever the user may be located” [4].

The VHE deployment in next-generation (4G) networks is complicated by two main factors: the fast trend towards the interworking of more and more heterogeneous network segments [1], and the growing request from networks customers for high communication reliability and quality of service (QoS) capability. Nonetheless, this is a relevant feature that novel 4G platforms have to carefully take into account to be competitive on the future telecommunications market.

Given the heterogeneity of network segments, appropriate interface extensions have to be defined in order to provide service personalization not only in cellular networks, but also over satellite access segments, wireless local (WLAN) and wide (WAN) area networks, as well as in pico-cell networks [1]. Furthermore, there is the need of adapting the system behavior to the featuring variables (e.g., context and environment) that characterize the various application scenarios. As an example, let us consider two typical 4G scenarios:

- a mobile user travelling across heterogeneous geographical locations (each one with its own characteristics) crosses a variety of contexts and environments;

- a user stands still in an exact location while the environment around him/her completely changes along with its relationships and requirements (e.g., due to a terrorist attack or a natural disaster).

In both cases, there is a strong need for continuous information exchange among all involved actors to the aim of matching user exigencies while facing the specific network constraints (i.e., type of network, resource availability, current network load, etc.). We define “situation” as the whole set of information that needs to be collected/handled/modified in order to define a logical representation of the scenario where the user operates. This set of information continuously evolves and reacts to various circumstances; thereby, Situation is a concept evolving trough different states.

In emerging 4G networks the Situation concept has to be exploited in a wider extent. It has to drive users/network in taking decisions to dynamically react to network condition changes and face users’ needs in a precise context by overcoming the limits arising from the external environments. The resulting Situation-adaptive 4G environment we are envisaging has to be characterized by suitable mechanisms for assuring users the capability to cooperate and access data and services in a way independent of both the geographical position and the type of devices used to connect; and for guaranteeing access to all information the user needs to achieve his/her objectives, even if it resides in different systems and while respecting his/her privacy.

Our main purpose is to create a *Situation-aware* overall telecommunications system that takes into account available *resources* (in terms of human, device, and network resources), *user's location, application typology* and *security* requirements, while always aiming towards user satisfaction.

It is clear that a traditional approach to the problem is inadequate. A unique centralized and very large system taking complete control of users and resources is unsatisfactory, rather it seems mandatory to exploit a certain number of autonomous smaller systems for guaranteeing the best global performance. This motivated our design of a software agent-based system platform that we call a Multi-Agent System (MAS). MAS is a middleware-based model that exploits a particular client-server paradigm powered by Intelligent Agents, whose peculiarities are:

- It is highly *subjective.* It handles quite a rich and detailed *user-profile* definition to record all relevant information about the user characteristics and behavior in particular Situations and Locations.
- It is *adaptive* with regard to users' coordinates, device, and application, and is able to dynamically react to them. The reaction is related to the device the user is exploiting in a given location and to the service he/she wants to access under certain resources availability, which contribute to define the overall Situation.
- It is *dynamic* and *flexible,* since it is provided with suitable mechanisms for reacting to the variations of the user's Situation and Location.
- It is *generic,* since it is capable of operating on a large variety of application contexts.
- It is *XML-based.* XML is a powerful and flexible language that well suits to our purposes; indeed, it is ideal for data interchange and interactions handling within the system. This allows achieving good performances even when resources are scarce.

This paper is organized as follows. In section 2 a general overview of the proposed Situation&Location-aware system is given; it can be applied to any application context. In sections 3 and 4 the agent-based MAS platform architecture and the main agents characteristics are respectively described. Conclusive remarks are reported in section 5.

II. THE PROPOSED SITUATION&LOCATION-AWARE SYSTEM

The focus of our research is to facilitate flexible, dynamic, personalized (user point of view), and optimized (network point of view) access to the network services by relating it to the user's Situation&Location. Moreover, we define access rules for handling either "normal" and "emergency" situations. In normal situations, parameters like the network status and the QoS requisites of the user application/terminal are basilar to drive the choice of the services and the resources to assign to the user. Among the system's tasks: the selection/reselection of the network where the traffic has to be routed as well as the dynamic negotiation/renegotiation of the communication sessions' parameters (the latter aspect implies cross-layer procedures). The heterogeneity and redundancy of the networks under analysis can also be exploited to handle emergency situations (e.g., natural disasters, terroristic attacks, black-outs, medical emergency, etc.). This can be done by compensating the temporary unavailability of one network coverage through the utilization of alternative - still functioning - wireless segments. In such a case, the handling of those communications services characterized by stringent security constraints (banking transactions, highly reserved communications, etc.) is interesting. In this case the network could switch the communication to a more secure segment.

To summarize, our target system has to optimally handle users with different terminal devices (PCs, notebooks, PDAs, mobile phones, etc.) accessing different "Applications" available at different "Locations" under a large variety of "Situations" (i.e. "Scenarios"). Variations in the Location may imply variations in the Situation. For example, a mobile user, under a 3G network coverage area, enters a WiFi hot spot. This could push him/her to change the access network, because the new choice best fits his/her profile, e.g. higher bandwidth is available. In this case, by changing the Location the Situation also changes.

On the other hand, it is also possible to have a change in the Situation without any modification of the Location. For example, in a multi-system coverage area, following an unexpected event (e.g., high congestion, network failures, terrorism attack, natural calamity, and so on), only a subset of the network segments still keeps on functioning. The emergency, for example, could force an inter-system handoff for groups of calls. Resources will be re-allocated in the still-functioning networks by privileging those services which are considered unavoidable (e.g., ambulance support, civil protection, fire-man support, tele-assistance, tele-medicine, etc.).

Now it clearly emerges the necessity for a clear definition of the "Situation" and "Location" concepts to refer to when operating within a future 4G scenario.

According to our view, by the term "Location" we mean a more general concept than the geographic position; this will be clearer in the next subsection. Obviously Location includes Position, i.e. information on the geographical coordinates and the velocity of the user. According to [6], position awareness is very important for future mobile applications and services. In our approach, we assume that position information can be suitably acquired by the user/terminal using one of the techniques currently available in communications networks (for example, satellite navigation systems like GPS, even augmented exploiting terrestrial infrastructure like 3G or WLAN [24], [5]). Whatever the technique used for discovering user position, it is fundamental the system to be able of *tracking* the user, foreseeing his/her movements, and consequently guaranteeing service continuity when either he/she changes the access network (vertical handover) or the access point within the same network (horizontal handover).

By the term "Situation" we mean the composite information that is collected to define a logical representation of the context where the user operates. In this sense, Location is one of the variables characterizing Situation; in other words, it is fundamental for locating the specific Situation within the global area of the system. We can say that Location contributes to define the geographical determination of the more complex concept of Situation. Of course, Situation is a concept that can evolve trough different states; therefore an accurate real-time information management mechanism is necessary to update the description of the particular Situation the user is currently living. This will be the topic of next sections.

A. A global system of resources

For better understanding, we preliminarily specify what we intend with the term "resources" in our system. They can be:

- *Human Resources:* these comprise all human beings distributed in the system area. Each user has his/her own profile storing all information necessary to define the complex reality we are considering. Each human resource, according to his/her own profile, can support other users living in similar Situations. On the other hand, it's important pointing out that in such an adaptive system an individual user may represent a resource and become fundamental to other users: acting as a sort of "connection provider" (ad hoc networking), "content provider" (e.g., music sharing), "service provider" (peer-to-peer), or even a "retailer".
- *Device Resources:* these comprise all the devices (either general-purpose or special-purpose devices) everywhere distributed in the system area (in the user's Personal Area, as well as in a Local, Metropolitan or Wide Area). Each device is characterized by a *device profile,* storing some features, such its maximum data rate, generated traffic type, and so on.
- *Network Resources:* these comprise information about network access points, along with their available bandwidth, access cost, coverage area, and so on. They must be managed in such a way as to optimize their use in a particular Situation.

B. A generic model of application context

We briefly define the reference scenario for our research:

- Different generic Service Providers (SPs) offer network services to users roaming in a given geographical area. Resources (of the human, device, and network type) are dynamically distributed in the whole area, and their distribution continuously changes.
- Users have roaming agreements with some SPs, and can access services from both wireless and wired devices. They need to use a given amount of resources for accessing the services and to maintain service continuity while moving through any location.
- Access to services is characterized by the particular Situation and Location currently associated with the user.

As a first assumption, the whole geographical region under the system control is split into smaller *Locations:* each Location is characterized by different amounts of available resources. Location is hence a more complex concept than the trivial geographical position; it also includes information about the (wireless and wired) network access points available in the area.

Each user travelling from a Location to the other can always be localized through his/her Position. And in that Position in the given Location, he/she can exploit available resources. This exploitation is determined by the specific variables defining the Situation the user is currently living.

As mentioned above, a Situation is the set of distributed information that is collected and dynamically, autonomously, and proactively handled to create a logical representation of the current user's working environment. Evidently the various resources and the different kinds of users, devices, and applications determine the whole set of variables representing the Situation.

We can imagine a user who, standing in a particular Location, connects to a given network. The system retrieves the user Position, finds his/her Location, and determines the bearer services and the resources which are available in the current Situation and are capable of providing a given set of services with acceptable QoS. Afterward, the system presents these services to the user by means of a graphical interface whose configuration is strictly dependent on the user's profile: services are indeed shown on the basis of their importance to the user as declared in his/her profile.

It is worth pointing out that each user can access services offered by different Providers; as a consequence, our system, on the basis of the user profile, can determine the best Providers for the user living in a specific Situation. Each SP provides a particular set of services: the user chooses the preferred Provider, and logs in for authentication before achieving the desired service.

The system continues to support the user until he/she decides to disconnect. With regard to this event, the system can also support a mobile user by proactively determining the adjacent Locations where he/she will continue using the service. The most suitable Locations can be suggested to the user; this way he/she can always know where the resources and the bearers, which guarantee the exploitation of a particular service, are available. In this paper, we prefer following a general-purpose approach to give high versatility to this early study, hence we specify neither what service nor the user typology we are arguing about.

III. THE GENERAL MULTI AGENT SYSTEM (MAS) ARCHITECTURE

The approach we follow to define our system is based on the exploitation of software-based Intelligent Agents working together in a Multi Agent System (MAS).

The provision of novel services in an open, global and mobile environment has significantly stimulated research work on new programming paradigms to enhance the flexibility and the adaptivity of the traditional Client/Server model. Mobile Agents represent a paradigm based on the principle of code mobility, and enhance the traditional client-server paradigm by performing changes along two orthogonal axes:

- The location where the know-how of the service is located.
- The actor that provides the computational resources.

In other words, if component A is a mobile agent, it has the know-how capabilities and a processor, but it lacks the resources where to perform its operations. It has the ability to migrate autonomously to a different computing node where the required resources are available. The computation associated with the interaction takes place on component B that has a processor and the required resources. As a consequence of this reasoning a mobile agent is not bound to the network host where it begins execution. The ability to travel allows a mobile agent to move to a destination agent. Moreover, the agent may be interested to exploit the services offered by the destination agent system.

In this work we focus on the Multi-Agent System (MAS) programming paradigm, which provides location-aware computing entities (mobile agents) with the possibility of migrating with code and execution state from a network host to another one while in execution [20]. The properties of mobility and adaptivity make the MAS technology high flexible because it can exploit information content from anywhere by accessing distributed resources and performing distributed operations in a completely asynchronous way with respect to both commanding users and originating hosts [20], [22], [15]. While first research activities on mobile agents aimed at the definition of models, design principles and methodologies, as the focus has moved to the implementation of MAS and MAS-based complex services, one of the main objectives of state-of-the-art MAS platforms is becoming the identification and provision of a common middleware[1] of basic facilities. This is fundamental for constructing an efficient system supporting service management. In addition, the intelligent agents approach, associating agents to user, device and services, allows a deep adaptivity of behaviour to the huge heterogeneity of application contexts.

The adoption of the Multi-Agent technology in the distributed system area is encouraged by many researchers. The deriving benefits are the followings [16]:

- Asynchronous/autonomous task execution.
- Reduction of network traffic and client processing power.
- Flexibility and extensibility.
- Decentralized control and management.
- Increased robustness.
- Adaptivity.

From all previous reasoning, it is possible to conclude that mobile agents provide flexibility in dynamically (re-) distributing intelligence inside a distributed network environment, in particular to reduce network load and to optimize custom service performance. The MAS benefits listed above allow to overcome various problems and inefficiencies typical of traditional Client-Server architectures, especially when dealing with intrinsically mobile application scenarios, such as users that modify their terminal for accessing the Internet and portable devices roaming from one hosting network to another one.

In addition, Quality of Service (QoS) requirements impose on-line adaptive operations that can significantly make profit from both agent migration and resource management.

The possible drawback of the Multi-Agent technology is represented by the security risks introduced by the necessity to host the execution of new computing entities carrying their own code. Furthermore, an agent may be attacked, modified or deleted by a hostile agent platform on a malicious network host. Therefore, many efforts are required in such a field, related to authorization and authentication. Another typical concern related to mobile agents is the question if agent migration is always advantageous, if compared with message passing. However, agent migration can be preferred to message passing because more and more evolved e-Services have the necessity to manage big information flows.

To our aim, MAS is able to react to the changes in the Situation for attaining total adaptivity to user, device and application. Adaptive applications in which we are interested need to meet three main constraints represented by *user preferences, device capabilities,* and *application requirements,* in relation to the Situation where user is involved in. This implies common representation of preferences, capabilities, and requirements as well as matching rules [7].

In our middleware platform we decided to use XML (eXtensible Markup Language) to represent the complex reality under observation. This choice has the following consequences:

- The agent ontologies are stored as XML documents; as a consequence, they are light and easy to be exchanged so they are suitable to reside on different devices and software platforms. This feature is crucial to guarantee the management of the agent ontologies in devices characterized by limited bandwidth and processor clock (where traditional, generally resource-expensive, information source managers cannot operate). In spite of this simplicity, the information representation rules embodied in XML are powerful enough to allow a sophisticated information management - this is fundamental for our system to operate in different scenarios.
- The communication language exploited by the various agents is ACML [21], obtained by combining XML and KQML [18], the XML encoding of FIPA Agent Communication Language [19]. The choice of this language has different kinds of advantages, in particular it becomes very simple to integrate agents with a large variety of technologies (e.g. *Secure Socket Layer, SSL,* for handling both the authentication of agent identities and the encryption of Agent CL messages).
- The extraction of information from the various data structures is carried out by means of XQuery [23]. This is becoming the standard query language for the XML environment. Since XQuery is based on the XML framework, it can handle a large variety of data. It has capabilities typical of database query languages as well as features of typical document management systems. Finally, it is provided with various high level constructs for simplifying querying over the Web.
- The manipulation of agent ontologies can be performed by means of the Document Object Model (DOM) [17]. This is a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of XML documents. DOM makes it possible for programmers to write applications properly working on all browsers and servers as well as on a large variety of hardware and software platforms.

[1] A *middleware* is a software that mediates between an application program and a network. It manages the interaction between disparate applications across the heterogeneous computing platforms.

More specifically, our MAS consists of three agent typologies:

- User-Device -Application Agent (UDA),
- Situation Management Agent (SMA),
- Resources Management Agent (RMA).

UDA is a client-side agent, strictly linked to a *user* equipped with a particular *device* and executing a particular *application* from any location. Each user is assigned to his/her own UDA agent, therefore we can say that each UDA is a sort of *personal* agent.

SMA and RMA are server-side agents. SMA (or even SMAs) is exploited for managing the whole set of resources and profile information for properly handling a Situation (defined in terms of user, application, device and network conditions) and reacting to it. RMA handles resource distribution and availability for the whole area of interest.

According to [14], a *unifying* platform should target all types of access networks, varying from picocells and Wireless LANs to satellites. Nowadays, various *middleware* solutions are available over these networks. Our convincement is that, with the network heterogeneity, the huge number of Providers and bearer services, and the increasing diversity of devices [1] - terminals, network components, and application servers - a common middleware platform can provide global-area management independently of any implementation model. We assume that SMA agents are part of this middleware layer, which is common to all underlaying networks. Indeed SMA, by "following" the user during his/her movements across heterogeneous networks (accessing services from different Providers through different bearer-services) should be independent of technologies and should always be able to "understand" what is happening around the user.

As for RMA agents, we should take some precautions in order to avoid heavy computational load. A feasible solution consists in splitting the global area into sub-regions, i.e. smaller regions constituted by a certain number of Locations. In each sub-region there is an RMA in charge of handling the resources (humans, devices and networks) situated therein. This way, our global system is constituted by different parallel-running computing agents (i.e., the RMAs) that distribute tasks among them. This is a distributed approach and, as a consequence, offers high reliability: in case of faults or RMA crash, there is always some other RMA that can give support. On the other hand, like every distributed solution, it can be affected by problems related to data consistence and interaction synchronization. According to these considerations, a feasible approach could be the adoption of a hybrid paradigm: an RMA-Coordinator could be foreseen for a certain number of sub-regions. Both RMAs and RMA coordinators need to be suitably chosen according to the underlying network segments across which our mobile agent migrate.

A. The high-level behaviour

In the following, we illustrate the general high-level behaviour of our system; in this description, each information flow is labelled by a letter that is also reported in Figure 1 to help the reader to better understand the information exchange among agents.

Before going on with the system behavior descriptions it is important to highlight that, in our platform, we suppose that each user signs a sort of subscription by means of a given form to fill. This kind of registration is the most suitable way for getting first knowledge of the user profile that, afterwards, is continuously updated. Hence, our approach has the advantages of both collaborative and content-based user modelling techniques: this *hybrid* approach, which will be better clarified in the following, is the best to our aims [6], [8], [9], [10], [11], [12].

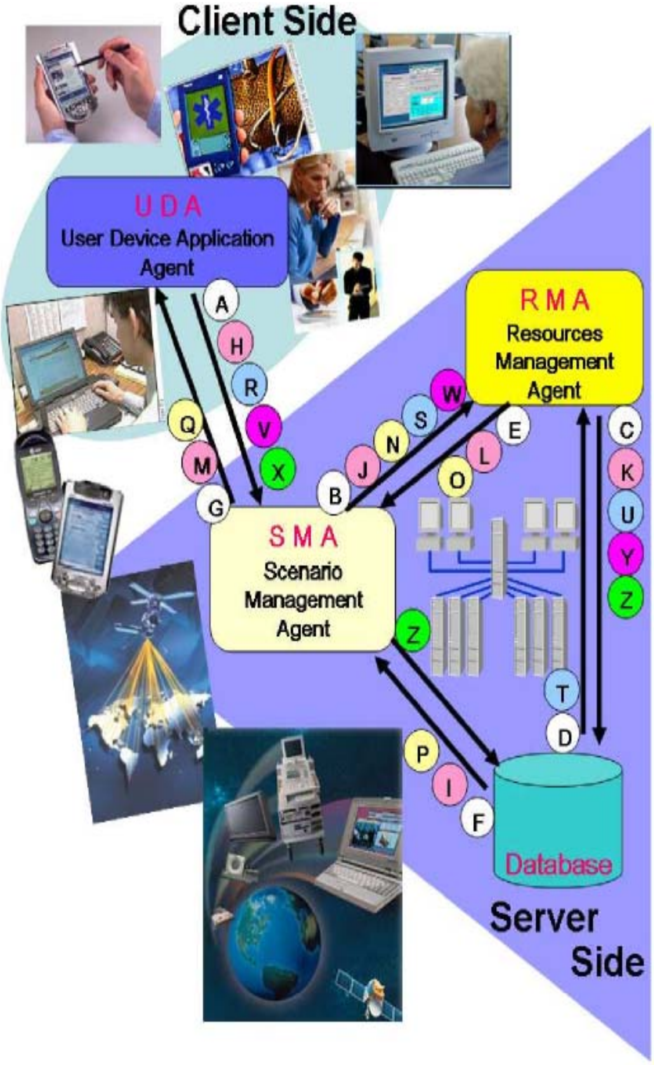
Client Side
UDA
User Device Application Agent
RMA
Resources Management Agent
SMA
Scenario Management Agent
Database
Server Side
A
H
R
V
X
Q
M
G
B
J
N
S
W
O
L
E
C
K
U
Y
Z
Z
T
D
P
I
F

Fig. 1. MAS Global Schema and General Behaviour.

1) A user U_j connects to the network by means of a specific device D_i. At this moment, he/she is not using any specific application yet.
2) U_j can be mobile or fixed: in both cases when accessing the network, he/she is in a particular *Position* within a specific *Location* of the global area of interest.
3) When U_j accesses the system, a corresponding UDA agent, $UDA_{ijStarting}$, is activated. It sends (label **A** in Fig.1) to a SMA agent, $SMA_{ijStarting}$, the identifiers of U_j and D_i along with the user Position. This information will be sent again by UDA to SMA each time some parameters change, for example when U_j changes his/her position or device terminal.
4) $SMA_{ijStarting}$ determines the *Location Area* where U_j

Position lies; then it sends to RMA **(B)** a request to receive the whole resource set (i.e. human, device, and network resources) available therein. RMA also provides the status of the network resources in terms of congestion level, maximum delay, available bandwidth as well as other qualitative network parameters. All this information makes $SMA_{ijStarting}$ capable of determining the QoS level achievable by an Application from a specific Location. Finally, RMA adds the human resource U_j to the resources associated to $Location_x$ *(C-D)* (note that the profile of U_j has been already recorded and stored both in the server database and in the user device).

5) After having received the resources available at $Location_x$, **(E),** $SMA_{ijStarting}$ matches the profile of D_i with the requirements of each service supplied by each Service Provider (for example, *Maximum Bandwidth, Video Audio and Text Enable*, and so on). Available services are summarized in a *Service Provider Global List* **(F)** where, for each service/application, the set of resources, required for achieving it, is specified. Some of these services will be marked as *Not Available* owing to the limits of D_i; this means that U_j cannot access them unless he/she changes D_i with another device satisfying the service requirements. All other services will be compared with resources available at $Location_x$.

6) From the result of the matching between resources and requirements $SMA_{ijStarting}$ finds the *Available Service List* for $Location_x$; this list contains all services that can be accessed in that particular Situation. For each available service $SMA_{ijStarting}$ determines also the QoS level reachable from the Location where the user is located.

7) $SMA_{ijStarting}$ compares the *Available Service List* to the characteristics of U_j stored in the user profile; this way, $SMA_{ijStarting}$ determines how close available services are to the preferences of U_j. $SMA_{ijStarting}$ sends to $UDA_{ijStarting}$ the *Available Service List* **(G)** sorted on the basis of U_j profile.

8) $UDA_{ijStarting}$ shows to U_j the whole set of *Available Services* sorted according to both his/her profile and the available network resources; they are also classified on the basis of the relevant Service Provider.

   At the end of the list, $UDA_{ijStarting}$ adds also the remaining Services of the *Service Provider Global List;* these are:

   a) the services marked as *Not Available* in the current Situation due to resource limits;
   b) the services that is impossible to achieve owing to device constraints.

   Each service is shown to the user along with a QoS level indicator specifying the QoS achievable in the current Situation.

9) U_j chooses the Provider and logs in for identification. After this, he/she chooses the target service. If the selected service is not available owing to device constraints, $UDA_{ijStarting}$ sends to the user suitable message specifying that the chosen service cannot be accessed by means of D_i.

10) If U_j chooses a service Sv_k **(H)** that can be accessed by means of D_i two cases might occur:

   a) Sv_k is available in the current Situation. In this circumstance, SMA_{ijk} **(I)** retrieves the set of resources required by Sv_k and stored in the *Service Provider Global List.* Then SMA_{ijk} sends to RMA **(J)** the request of "Resource Access" for that particular Situation: RMA reserves the resources **(K)** characterized by the best trade-off between service requirements and resource availability. When RMA ends its task, **(L)** SMA_{ijk} sends to UDA_{ijk} a confirmation message **(M).** UDA_{ijk} lets Sv_k start, following all indications received from SMA_{ijk}.

   During the whole access of U_j to Sv_k, UDA_{ijk} monitors the various activities and tries to support U_j; as an example UDA_{ijk} can switch the link type to the best available link according to the updated Situation information. As a further example, a service might become available only if a link different from the best one is exploited: in this case UDA_{ijk} suggests it to U_j and points out the difference in terms of costs and other useful information.

   b) Sv_k is not among the available services for the current Situation. In this circumstance our system has to take user mobility into consideration: if U_j is mobile, then the system suggests a possible Location where the user can access Sv_k. On the contrary, if U_j is fixed, then he/she receives the message "Service Not Available". After this message UDA_{ijk} asks to U_j whether he/she wants to know the Locations where Sv_k can be accessed or if he/she wants to choose another service. In the former case, step (11) is performed, otherwise it is necessary to return to step (9).

11) SMA_{ijk} communicates to RMA the resource requirements of Sv_k **(N)** and requires the whole set of Locations adjacent to $Location_x$ where they are available. Observe that this request is dual w.r.t. that considered in point (4). Thanks to its proactivity, SMA_{ijk} is capable of supporting U_j while he/she exploits Sv_k even if the Situation changes.

12) RMA sends to SMA_{ijk} the list of Locations **(O),** and SMA_{ijk} sorts them on the basis of available QoS. This QoS has to match the QoS level preferred by U_j in the past and indicated in his/her profile. SMA_{ijk} proposes the different Locations to UDA_{ijk} **(Q)** that suggests them to U_j. The resources necessary for achieving a given QoS level are determined by applying the procedure referred in [13]. According to [13], QoS can be perfectly modelled on the basis of some parameters, such as minimum bandwidth, maximum delay, maximum congestion level, preferred link type, etc. Therefore, according to the chosen QoS level, the resources needed by U_j can be determined.

13) Of course, UDA_{ijk} stores each choice made by U_j in his/her profile, which is, therefore, continuously updated. When U_j changes his/her Location, UDA_{ijk} sends to SMA_{ijk} information on the new Location **(R);** SMA_{ijk} asks RMA **(S)** to acquire the new resources necessary for Sv_k **(T);** in addition, RMA releases the resources

previously acquired **(U).** This procedure continues until U_j accesses Sv_k: this implies that our system repeats steps (11) to (13) until U_j ends his/her session.

14) When U_j decides to close the application, UDA_{ijk} communicates this to SMA_{ijk} **(V)** that asks RMA to release the resources previously occupied **(W-Y);** after this, the whole process starts again from point (4) .

15) When U_j wants to disconnect from the system, $UDA_{ijEnding}$ sends the related command and the updated profile to $SMA_{ijEnding}$ **(X)**. After this, $SMA_{ijEnding}$ stops supporting U_j and updates his/her stored profile in the Server Database with the new information. **(Z)** Finally, RMA detracts the "human resource" U_j from the global area resources and turns his/her status into *inactive.*

It is worth pointing out that we can generalize our paradigm to foresee that *human* support can be required by a user in an *emergency* Situation. In this type of Situation, indeed, resource requirements are humans[2]. The system finds each required human resource, and sends him/her an *emergency message* specifying the Position of the user who required the support.

IV. AGENT DESCRIPTION

In this section we briefly describe the main features characterizing each agent acting in the envisaged system.

A. UDA: User-Device -Application Agent

A User-Device-Application Agent, $UDA_{ijk,}$ is associated with a user U_j exploiting a device D_i for executing a particular application A_k. UDA_{ijk} supports U_j from the moment he/she connects to the system by means of D_i until he/she decides to disconnect.

UDA Ontology

The ontology of UDA_{ijk} consists of the tuple

$$< DP_i, UP_j,\ AP_k,\ SInfo >,$$

where:

- DP_i, the D_i Device Profile, stores some characteristics of D_i, such as the maximum data rate, the medium type (e.g., video, audio, etc.) it can handle, the interface link type (wired, wireless, satellite, etc.), and so on.
- UP_j, the U_j User Profile, stores some characteristics of U_j, such as the service he/she wants to achieve, his/her background, his/her Position, and other useful information.
- AP_k, the A_k Application Profile, stores some characteristics of A_k, such as its required security level and priority, its interaction styles (realtime or store & forward), the number of involved telecommunication devices, and so on.
- *SInfo* stores various information about the current Situation for user U_j, such as the available services (determined by SMA according to some variables characterizing the current Situation, such as the available resources and bearer services), the preferred link type, and the suggested Location. These parameters are exchanged between SMA and UDA, handled by the latter one to make the user aware of the Situation and allow him/her to take the most advantageous decisions.

Table I illustrates the parameters characterizing the ontology of UDA_{ijk}.

B. SMA: Situation Management Agent

The Situation Management Agent *(SMA)* has to manage the whole information collected from the distributed agents working in the system. This information is dynamically, autonomously, and pro-actively handled in order to create the logical representation of the environment and the particular scenario where the user is located and operates: namely, the *Situation.*

SMA Ontology

The SMA ontology consists of the tuple

$$< UDAInfo, SPL,\ \ SPGL > ,$$

where:

- *UDAInfo* is the information sent by UDA_{ijk} to SMA necessary for defining a Situation that involves U_j while he/she exploits a device D_i to access application A_k.
- *SPL* is the *Situation Profiles List.* This is a list of *Situation Profile* objects that are created by SMA. This list includes both the current user Situation (related to the current user Location) and the Situations in the nearest Locations where the User-Device-Application requirements can be fulfilled.
- *SPGL* is the *Service Provider Global List.* In this list there is the whole set of *Services* provided by each Service Provider. Each service is defined in terms of human, device, and network resource requirements that a user needs in order to access that service.

Table II describes the SMA ontology.

C. RMA: Resources Management Agent

The Resource Management Agent *(RMA)* acts as a "mediator" between SMA and specific resource management modules of the underlying networks. It is responsible for the reservation, allocation, and de-allocation of resources, according to SMA indications as well as to resource availability.

[2] Remember that each user can be considered as a human resource to other users with characteristics determined by his/her profile

DP_i		**Profile of the Device D $_i$**
	DId_i	*Identifier* of Di
	$BMax_i$	*Maximum Bandwidth* that Di needs for accessing or downloading data from the network
	VE_i	*Video Enabled;* it is set to 1 if Di supports video traffic,0 otherwise
	AE_i	*Audio Enabled;* it is analogous to VE_i but relevant to audio
	TE_i	*Text Enabled;* it is analogous to VE_i but relevant to text
	$Mobile_i$	*Mobile Enabled;* it is set to 1 if Di is a mobile device
	$UsedLinkType_i$	It is the link type used by Di according to a pre-established set of typologies with different characteristics
	$AvailableLinkType_i$	It is a list of link types detected by Di in the current Situation
UP_j		**Profile of the User U $_j$**
	UId_j	*Identifier* of Uj
	$RegProviderList_j$	*Registered Provider List;* it is the list of Service Providers which Uj is registered to
	$DesiredService_j$	*Desired Service* of Uj *;* it is the service Uj currently wants to access
	$DesiredDest_j$	*Desired Destination* of Uj *;* it is the next Location where Uj will go (optional)
	$Position_j$	It indicates where Uj is currently located
	$MaxTime_j$	It is the *Maximum Time* Uj wants to spend for enjoying a service (optional)
	$SpecificProfData_j$	It lists parameters of the Uj profile that are specific for every application context. They are updated at every access.
AP_k		**Profile of the Application A $_k$**
	$AppId_k$	*Identifier* of Ak
	$SecurityLevel_k$	*Security Level* required by Ak
	$Priority_k$	*Priority Level* of Ak. It is important for resources distribution
	$T\ ype_k$	*Application Type* relevant to Ak. It indicates how many devices are involved in a communication session for Ak (Multicast, Unicast,...)
	$InvolvedUser_k$	*Involved Users List;* it is a list of users involved in a session relevant to Ak
	$InteractionStyle_k$	*Interaction Style* of Ak *;* it can be Realtime or Store & Forward
	$ActiveInd_k$	*Activity Indicator;* it is set to 1 if Ak is active, it is set to 0 otherwise
SInfo		**Situation Information**
	AvailableService	It is the list of services available in a given Situation
	PrfLinkType	*Preferred Link Type;* it is determined on the basis of the Situation characteristics
	SuggestedLoc	*Suggested Location List;* it is the list of Locations where Uj could move to continue accessing application Ak. Locations are ordered on the basis of the Uj preferences stored in his/her profile and related to his/her previous choices

TABLE I. THE ONTOLOGY OF UDA_{ijk}

UDAInfo		**Information about User, Device and Application**
	UId_j	*Identifier* of *Uj*
	$DesiredService_j$	It is the Service *Uj* wants to receive
	$DesiredDest_j$	*Desired Destination* of *Uj;* it is the next Location where *Uj* will go (optional)
	$Position_j$	*Position* where *Uj* is currently located
	DP_i	the whole *Profile of Device Di* (see Table I)
	AP_k	the whole *Profile ofApplication Ak*(see Table I)
SPL	*SPL*	**Situation Profile List** *List of Situation.* Every *Situation* is a well defined complex object
SPGL	*SPGL*	**Service Provider Global List** the *list of Services* furnished by each Service Provider. Each *Service* is a well defined complex object.

TABLE II. THE ONTOLOGY OF SMA

Up to now we illustrated RMA by considering it as the single agent responsible for resource management: actually, RMA is just one of the agents in charge of resource management. Indeed, in our work, three further agents cooperate with RMA. They are in charge of network resources and they operate according to the behaviour illustrated in [13]:

- *Resource Agent (RA)* reserves, allocates, and deallocates the network resources according to user preferences and profile as well as network availability;
- *Fault Agent (FA)* monitors the network to detect the (possible) presence of faults;
- *Negotiator Agent (NA)* negotiates with RMA and SMA, anytime an increase (or reduction) in the network resource availability occurs. Their aim is to find a new resource distribution able to maximize user satisfaction. Negotiation is necessary any time the current resource allocation is incompatible with the network status. *NA* tries to make the network resources distribution compatible with the new network status, and to maximize user satisfaction according to the information retrieved by SMA and RMA. After the decision on the new resource distribution, *NA* computes the relevant values of high-level QoS parameters and notifies RMA of them. Then RMA notifies SMA of the new resource distribution, and SMA changes Situation parameters change accordingly. In any case, *NA* provides both *RA* and RMA with all changes, concerning high-level QoS parameters and bandwidth, following the negotiation phase. Consequently, *RA* can perform the above-mentioned activities.

RMA Ontology

The ontology of *RMA* consists of the tuple

<GlobalArea(N,A), HRL, DRL, NRInfo, SituationInfo>,

where:

- *GlobalArea(N,A)* is the graph representing the global system area. *N* is the set of nodes, *A* is the set of arcs. The geographical area managed by our system is divided into smaller regions called *Locations:* each Location is a node, and the adjacency relationships between two Location areas is represented by an arc connecting them.
- *HRL* is the *Human Resources List.* In this list the whole set of human resources that are active in the global system area is included. As already mentioned, each user can be considered as a human resource; his/her profile gives the needed information about his/her competencies. Each human resource stored in this list consists of

  < HLocation, UserProfileInfo, Status >.
- *DRL* is the *Device Resources List.* In this list there is the whole set of instruments/tools that are located in the different Locations of the global area. These can be general-purpose and specialist devices. As with *HRL,* also each *DRL* element consists of

  < DLocation, DeviceProfileInfo, Status >.
- *NRInfoList* is the *Network Resource Information List.* The most important network parameters are stored in each element of this list. The list works as an information intermediary between SMA and the agents assigned to the network management. SMA stores in this field information for achieving the required resources that *RA* has to allocate; on the other hand, also *RA* stores here the resources assigned to a specific user.
- *SituationInfoList* is a list of various *SituationInfo* complex elements. Each object of this list stores information about a specific Situation, and is useful to allow a simple Situation management.

Table III describes the RMA ontology.

V. CONCLUSIONS

In this paper a platform based on the exploitation of a Multi Agent System is proposed with the aim of providing a user with seamless and optimized network access to the desired service based on its current “Situation&Location”. High versatility is guaranteed by the general-purpose approach we followed in our research. In fact, a key feature of the proposed system is the presence of a middleware platform exploiting user profile-, resource-, and situation-management agents whose ontologies are specified in XML languages. The main strength of the resulting system is, therefore, its high degree of “adaptability” that can be achieved without the use of either new terminals or novel technologies.

REFERENCES

[1] T. Zahariadis, “Trends in the Path to 4G”,IEE Commun. Eng., Feb. 2003, pp. 12-15.

[2] T. Itao et al., “Context-Aware Construction of Ubiquitous Services”, IEICE Trans. Commun., vol.E84-B, no.12, Dec.2001, pp. 3181-88.

[3] ITU-T Draft Rec. Q.1711, “Report of the Meeting Held in Geneva, Switzerland, 5-22 May 1998, Part II”, May 1998.

[4] 3GPP TS 22.121, “The Virtual Home Environment (Release 4)”, v. 4.0.0, Oct. 2000.

[5] 3GPP TS 23.005, “Stage 2 Functional Specification of Location Services in UTRAN (Release 99)”, v. 3.3.0, Oct. 2000, http://www.3gpp.org

[6] S. Arbanowski, et al., “I-centric Communications: Personalization, Ambient Awareness, and Adaptability for Future Mobile Services”, IEEE Communications Magazine, vol. 42, no. 9, pp. 63-69, September 2004.

[7] H. Van Kranenburg et al., “Ubiquitous Attentiveness: Enabling Context-Aware Mobile Applications and Services”, 1st Euro. Symp. Ambient Intelligence, Veldhoven, The Netherlands, Nov. 2003.

[8] M. Balabanovic and Y. Shoam. Fab, ”Content-based, collaborative recommendations”, Communication of ACM, 40(3):66-72, 1997.

[9] C. Basu, H. Hirsh, and W. Cohen, “Recommendation as Classification: Using Social and Content-Based Information in Recommendation”, Proc. of Fifteenth National Conference on Artificial Intelligence (AAAI’98), pp. 714-720, Madison, Wisconsin, USA, 1998. AAAI Press / The MIT Press.

[9] C. Basu, H. Hirsh, and W. Cohen. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In Proc. of Fifteenth National Conference on Artificial Intelligence (AAAI’98), pages 714-720, Madison, Wisconsin, USA, 1998. AAAI Press / The MIT Press.

ONTOLOGY FIELD	SINGLE ELEMENT	DESCRIPTION
GlobalArea(N,A)	*GlobalArea(N,A)*	**Global Area Graph** the *Graph* representing the global system area
HRL	***HR*** *HLocation* *UserProfileInfo* *Status*	**Human Resources** *Location* where the human resource is located *User Profile Information* *Status* of the human resource. It is set to 0 if the resource is *Occupied,* it is set to 1 otherwise
DRL	***DR*** *DLocation* *DeviceProfileInfo* *Status*	**Device Resources** *Location* where the device is located *Device Profile Information* *Status* of the device resource. It is set to 0 if the resource is *Occupied,* it is set to 1 otherwise
NRInfoList	***NRInfo*** *NetId* *NLocation* *OverallInfo* *NType* *UserConnected*	**Network Resources Intermediary Info** *Identifier* of the Network Resource *Location* where the access network is located *Global Information* about the network status *Network Interface.* It can be either wireless or wired *Connected Users List.* List of users connected to this network
SituationInfoList	***SituationInfo*** *UserID* *UserNetData* *Loc* *AppProfile* *UsedNetId* *RequiredNet* *AssignedNet*	**Situation Information** *Identifier* of the user involved in the Situation *User Profile data* relative to the network usage *Location* of the Situation *Application Profile* *Identifier of the Network* exploited by User *Network Resources required* by User *Network Resources Assigned* to User

TABLE III. THE ONTOLOGY OF *RMA*

[10] A. Kobsa, “Generic user modeling systems”, User Modeling and User-Adapted Interaction, 11:49-63, 2001.

[11] M. Pazzani, “A framework for collaborative, content-based and demographic filtering”, Artificial Intelligence Review, 13(5):393-408, 1999.

[12] J. Delgado, N. Ishii, and T. Ura, “Content-based collaborative information filtering: Actively learning to classify and recommend documents”. In Proc. of Int. Workshop on Cooperative Information Agents (CIA’98), pp 206-215, Paris, France, 1998. Lecture Notes in Computer Science, Springer-Verlag.

[13] G. Araniti, P. D. Meo, A. Iera, and D. Ursino, “Adaptively Controlling the QoS of Multimedia Wireless Applications Through “User Profiling” Techniques”. Journal of Selected Areas in Communications, 21(10):1546-1556, 2003.

[14] S. Karlich et al., “A Self-Adaptive Service Provisioning Framework for 3G+/4G Mobile Applications”. IEEE Wireless Communications, 11(5):48-56, October 2004.

[15] P. Bellavista, “Mobile Agent Models and Technologies for Distributed Coordinated Applications in Global Systems,” Phd Thesis, University of Bologna, 2000.

[16] D. Chess, C. G. Harrison, and A. Kershenbaum, “Mobile Agents: Are They a Good Idea?”, 1995

[17] Document Object Model, [http://www.w3.org/DOM/].

[18] T.W. Finin, R. Fritzon, D. McKay, and R. McEntire, “KQML as an agent communication language”, Proc. of the ACM International Conference on Information and Knowledge Management (CIKM’94), Gaithersburg, Maryland, USA, 1994, pp. 456-463, ACM Press.

[19] Foundation for Intelligent Physical Agents Specifications (FIPA), [http://www.fipa.org].

[20] A. Fuggetta, G. P. Picco, and G. Vigna, “Understanding Code Mobility”, IEEE Transactions on Software Engineering, vol. 24, n. 5. pp 342-361, 1998.

[21] B. Grosof and Y. Labrou, “An Approach to using XML and a Rule-based Content Language with an Agent Communication Language”, Proc. of the IJCAI-99 Workshop on Agent Communication Language, Stockolm, Sweden, 1999, pp. 96-117.

[22] D. Kotz and F. Mattern, “Agent Systems, Mobile Agents, and Applications”, 2nd Int. Symp. Agent Systems and Applications and 4th Int. Symp. Mobile Agents (ASA/MA 2000), Switzerland,2000

[23] World Wide Web Consortium - XML Query, [http://www. w3. org/XML/Query].

[24] G. Heinrichs, P. Mulassano, and F. Dovis, “A Hybrid Positioning Algorithm for Cellular Radio Networks by Using a Common RAKE Receiver Architecture”, Proceedings of the PIMRC Conference, Barcelona, Spain, 2004

Antonio Iera graduated in Computer Engineering at the University of Calabria, Italy, in 1991 and received a Master Diploma in Information Technology from CEFRIEL/Politecnico di Milano, Italy, in 1992 and a Ph.D. degree from the University of Calabria in 1996. From 1997 to 2000 he has been with the University "Mediterranea" of Reggio Calabria, Italy, as Assistant Professor and from 2001 to 2004 as Associate Professor. Currently, he is Full Professor of Telecommunications at the same University. His research interests include: broadband wireless networks, heterogeneous wireless systems, advanced systems for Personal Communications, broadband satellite systems.

Antonella Molinaro graduated in Computer Engineering at the University of Calabria, Italy, in 1991. She received a Master's degree in Information Technology from CEFRIEL/Politecnico di Milano, Italy, in 1992, and a Ph.D. degree in multimedia technologies and communication systems from the University of Calabria in 1996. Since 1998 she has been an Assistant Professor, first at the University of Messina and then at the University of Calabria. She is currently an Associate Professor at the University "Mediterranea" of Reggio Calabria, Italy. Her research interests include wireless and satellite systems; inter-working of heterogeneous networks; wireless and mobile Internet technologies and applications.

Alfredo Pudano received the Laurea Degree in Electronic Engineering from the University Mediterranea of Reggio Calabria in July 2004 by discussing the thesis *"A Scenario-adaptive, multi-agent system for handling e-services in heterogeneous telecommunications networks "*. From September 2004 to December 2004 he collaborated with the Computer Science and Telecommunications Research Groups at the same university. From January 2005 he is a PhD Student at the University Mediterranea of Reggio Calabria. His research interests include user modeling, device adaptivity, telecommunications networks, domotics, power consumption saving and comfort in indoor environments by means of Artificial Intelligence solutions.

Domenico Ursino received the Laurea Degree in Computer Engineering from the University of Calabria in July 1995. From September 1995 to October 2000 he was a member of the Knowledge Engineering group at DEIS. He received his PhD in System Engineering and Computer Science from the University of Calabria in January 2000. From October 2000 to December 2004 he was an Assistant Professor in Computer Science at the University Mediterranea of Reggio Calabria. Currently he is an Associate Professor at the same University. His research interests include user modeling, intelligent agents, e-services, knowledge extraction and representation, scheme integration and abstraction, semi-structured data and XML, Cooperative Information Systems.