VPIN: An Event Based Knowledge Inference for a User Centric Information System

Netzahualcoyotl Ornelas [2], Noëmie Simoni[1], Chunyang Yin[1], Antoine Boutignon[3]
[1]Institut TELECOM / TELECOM ParisTech
46 rue Barrault, 75634 Paris Cedex 13 France
{simoni, yin}@telecom-paristech.fr
[2]L2TI - Institut Galilee - University Paris 13
99 Avenue J-B Clément, 93430 Villetaneuse France
netza@univ-paris13.fr
[3]Société Française du Radiotéléphone (SFR)/DRI
40-42, quai du Point du Jour, 92659 Boulogne Billancourt Cedex, France
antoine.boutignon@sfr.com

Abstract

In the Next Generation Network, the telecommunication environment becomes more and more heterogeneous and mobile (concerning user, terminal, network, service). User must be the central point of the consideration due to this ambient and dynamic network environment. The future information systems should be user oriented and have to perform well with ever changing of the telecom environment due to mobility and usage. Unfortunately, the existing solutions are traditionally inspired by the methodologies which are object-oriented, application-related, which leads a semantic gap between the user new requirements and the real environment.

In this paper, to reduce the mentioned semantic gap, we propose to adopt a vision of User Centric and a common information model as the departure point. We propose a persistent VPIN (Virtual Private Information Network) as a knowledge base applying this information model. Therefore, VPIN is able to take into account all the heterogeneous elements in the NGN context and be independent of any application. To mange the dynamicity of the ambient environment, we propose that the VPIN acts as an inference through the events handing. To insure that the events can be handled automatically, we propose alternative ways to manage and update the VPIN.

Keywords: User-centric approach, information model, QoS, State, inference

1 Introduction

Next Generation Networks offer the telecommunication consumers with numerous business benefits as well as the opportunity to work faster and smarter. In order to take advantage of these benefits and opportunities, we should take into account the technologies evolution and the application domain enlargement. The technologies evolution brings us the heterogeneity. Nowadays, although the core network rests the same, the access to the backbone is differed from wired to wireless ways. Especially for the wireless access, we have now not only different ways (WiFi, Bluetooth, IrDA) to construct a local network, but also the ways (GPRS, UMTS, Wimax) to have a very large coverage. The heterogeneity exists equally in the terminal types as PC, laptop, PDA, Mobile phone etc. As a result, the services are thus also heterogeneous due to different supports and providers.

As we mentioned, NGN has also brought us an enlarged application domain, which enables the users to move during the utilization of the network and the services. According to the different actors in the NGN, we resume that there are generally four types of mobility:

First, we can consider the user mobility, which is the capacity of a user to change of terminal. For example when the user has different types of devices, he desires to use his PDA to check his e-mails and after, as he moves to another room, he uses his personal computer to continue checking his e-mails. Besides this mobility, there is the terminal mobility. It refers to the capacity of a terminal to change of access networks. For example, when a user is having a conversation using his cell phone and the user is driving in his car, in that case, the cell phone cross several access networks, and there is not an interruption in the communication. An additional type of mobility is the network mobility. It permits a network equipment to move of place, allowing the communication with different network equipments that are in constant movement too. An example is a vehicular network. It means, that any type of vehicle that contains the same router equipment, allows communicating with the different vehicles located nearby and that are constantly in movement. And in last, the service mobility, which is the capacity of a service to change of component when a component that provides a service stop working. We can imagine a server of streaming that diffuses a Radio station, therefore, if that server stops working, there could be another server that retakes the same diffusion in real time, as a result, the service continue to be delivered.

At the apparition of personal computer systems, there was an obligation for the user to have a minimum technical knowledge of the system installed in order to use it and take advantage of it. Nevertheless, it's not an easy task for user to control every kind of Operating System in the market (Windows, Linux, and Mac, etc.), as a result, the utilization of the system by the user was minimum compared to the capacity offered by the personal computer.

However, the utilization of systems resources started to be insufficient to fulfil the user requirements. All capabilities in the resources were integrated in the system; consequently, the access feasibility to these resources can be improved through a network. For example, when stocking a big volume of information, it's better to do it through a disk available in the network whose capacities are bigger than in the user system. In addition, the access to this information can be done by any computer even of different user location. This optimizes the use of this resource without the necessity to change the computer hard disk when this is already almost at 100 percent of utilization.

However, this new way to access resources through a network, implies end user to learn some additional, and sometimes, complicated process to achieve the use of a service. Therefore, these technical skills that must be learned by end users are not easy to accomplish as we think; especially when network applications evolve quickly and a constant apprenticeship is needed.

Although the access to resources by the network is a viable solution, end users felt an important limitation to adapt the use of their services, according to the technical specifications, and configuration of the network. In fact, a limitation in the network could be, when end user wishes to watch a TV program with his personal computer. Most of the time, the session establishment is achieved, but sometimes quality in the reception is not good enough as the user wished. More over, if the user wants to move to another place and his computer is connected by an Ethernet cable, he is obligated to disconnect the cable, and to reconnect via other access network such as the Wi-Fi for example. Besides, it could happen that the reception of the TV program is not possible due to some interference in the Wi-Fi access network. Therefore, a cut of his service is introduced. Otherwise, some user skills are required to achieve this procedure, and re-establish the reception of the TV program.

Hence, telecommunications operators and service providers realized the difficult encountered by the end users. The complexities of those user problems are not only when accessing his services, but also when trying to adapt them according their preferences. Therefore, from now, operators and service providers, consider the user as the main point of interest in the offer of new services. It means, they prefer to better know what the end users desire to access their services than the follow the technological tendencies.

In fact, that's the end-users who desire access to all types of services; for example, the user own services (e.g. check e-mail) or services that are offered according to his location (e.g. receive announces of reduction in his PDA when he is in a supermarket). In addition, it is always the user who desires access to these services with any kind of device available around him at any time. Beside these requirements, the user is usually confronted to some technical barriers that are obstacles to access his services.

It is for this reason, in order to fulfil those user needs, a new approach in the technological conception should be considered. Instead of the design vision of System Centric or Application Centric, we will take the vision of User Centric as the centre of the global architecture. This could change the way the technological evolution was carried out until now. In this new approach, user needs could make the creation of new technologies and a new vision for service delivery, trying to respect, an adaptability of those services according to user location and his preferences.

Anyway, user preferences are more difficult to be achieved as we thought. They can change in a short period of time, according to the user context and they needs. One of these user preferences could be the cost. For example; when a user wants to hire the cheapest Internet service connection at home, he will check the offers and the packets of the different services providers. The selected operator will be the one that offer a major of services with the cheapest price in the market. Another type of user preferences might be technical. Based on the cost example, we can imagine that after the selection of the Internet service provider, the user wishes to choose the type of connection (e.g.

Ethernet, Wi-Fi), besides, maybe the user desires to use another device instead of his personal computer (e.g. PDA). Or even, when user accesses his services, in order to be in communication with his family, he prefers employ a Videoconference program installed in his computer, instead of using the e-mail.

At last, we can consider the usage preferences. They consist in the different configurations that are supported in a component. That is, once the user decided to watch the TV on his computer, some adaptations can be done in the quality of the image (pixels, image in colour or black and white) or the quality of the sound (e.g. sound stereo, etc.).

All of these user preferences could help to accomplish some user needs; but not all of them, as we need to take into account that the user is in constant movement. It's important to recall that these different types of mobility can produce a user-session interruption.

In order to avoid user-session interruptions, (due to the different types of mobility recently explained), we believe that it would be useful to obtain a collection of necessary information according to a specific user, i.e. a user Information System.

The expected Information System could allow a user having a global vision of all available resources no matter which providers support them or where they are placed. This could make user possible to check their capacities as well as what can be offered by a resource.

Actually, this user side information system is expected to be a knowledge system which contains necessary and sufficient information for a dynamic and sophisticate environment, as the behavior related information, user preferences etc to enable the decision to be taken according to the contained information.

This Information System should also be an application-independent knowledge inference managing the user side applying rules' information and maintaining itself by responding automatically and alternatively the dynamic mobile telecom world.

Unfortunately, the existing solutions are traditionally inspired by the methodologies which are object-oriented, application-related, which leads a semantic gap between the user new requirements and the real environment.

To have a clear idea of which kind of User Information System we should conceive, some questions need to be answered, as:

a) How to define the "relevant and decisional information" for the user information system? b) How to structure this information in the best possible way to let the Information System be application-independent? c) How to manage the information most efficiently? What are the automatic processes in order to maintain the information system dynamic? d) When a change occurs in the IS, how to update the whole IS automatically in order to keep in touch with the real time world?

To answer the first question, we consider that the relevant information permits to have a global vision of the behavior of all the components, such as terminals, networks and services. This information is present at any place and any time. We believe that the decisional information implies that we must assure the provision of this relevant information at the right time and the right place; that is not only at session establishment but also during the session itself. This could introduce a dynamic session management requirement.

Responding the second question that relates the structure information, we consider that it must be provided and structured according to the real world's vision, which means the different types of mobility and heterogeneity aspects, while respecting the user's requirements (preferences, QoS), in order to make the adequate decisions.

And to answer the last two questions, we think that the automatic process starts when an event is produced, this event could be a change of a component's behavior. This means that an action must be taken in order to fulfill the self-management and offer services transparently (without interruptions).

In order to establish and maintain a user Information System where the heterogeneity (terminal, network, and service) and mobility (user, terminal, network and service) are omnipresent, we consider necessary to have the necessary, sufficient and decisional information (knowledge base), as well as the actions to take when an event is produced (change of the QoS in a component), in order to make the correct decision autonomously. The fact of taking correct decision autonomously makes a transparency for the end user when using his services.

This transparency must be valid no matter the user location, the utilization of terminals and networks that are to his disposition.

Hence, we propose a VPIN (Virtual Private Information Network) [1] based on an information model. The Information Model represents the real world by an informational structure, which is semantic of the new environment. VPIN provides a complete information image of the components of the user's system (including QoS information). The persistent information in the VPIN enables also the separation of the user system and the user's different applications, which helps to manage user sessions in ambient networks.

This paper is organized as follows. We present in Section 2 the related work, which allows us to take into account the recent solutions proposed until now. In

Section 3, we introduce the background of our work to explain the architectural model derived form Meta-model NLN (Node, Link, and Network), as well as our QoS based information model, on which we base our proposition. Afterwards, in Section 4, we propose organizationally the VPIN, which possesses the adequate information for the user. In section 5, the functional dimension is detailed by the identification of the events [2][3][4] the inference functional specifications. In order to examine the feasibility of our concept, in Section 6, we describe an experimentation to illustrate our proposition. And finally, Section 7 presents the conclusions and perspectives for future work.

2 Related work

Before presenting our proposals concerning the management information, we analyze the content of some related structural proposals, which concern different *information models*.

Currently, in Information Technologies environments, some efforts have been done in standards works to consider the *informational models* as the representation of information about entities to be managed. These works are more interested in the representation of information related with the different entities to be managed but they don't have the same abstraction level, or the same coverage. We find the GNIM for the Networks elements, the CIM for the enterprise elements, the WBEM for allowing a uniform access through the web, the DEN-ng for the integration of the context constraints, and finally, the SID, which the aim is to cover all the information system from the strategy, the services, to billing.

The GNIM (Generic Network Information Model) [5] is a recommendation (M.3100) proposed by the ITU-T [6] through the TMN (Telecommunications Management Network). It specifies a generic network management information model for the management of telecommunications networks. It consists of a set of abstract or common managed object classes and their properties that may be specialized to support the management of various technologies, architectures and services. This model can be seen as a set of reusable managed object classes that may be adapted to support the management of various telecommunications networks. As is a generic model, it allows being independent of a technology as ATM (Asynchronous Transfer Mode), Frame Relay, SDH, PDH, PSTN, etc.

The GNIM defines four levels of management. These levels are the Business, Service, Network and Network Element (NE) management levels. Note that the service and network management level aspects are weak.

The CIM (Common Information Model) [7] is proposed by the DMTF (Distributed Management Task Force) [8], which is an open standard that defines how to manage elements and their relationships in an IT environment. The elements are represented as a common set of objects, which allow consistent management independent of their manufacturer or provider.

CIM is an information model that presents a conceptual view of a managed environment. A first goal of CIM is to unify and extend the existing management standards such as SNMP, DMI, CMIP, etc. by using and object-oriented design. The second goal of CIM is enabling to model all the different aspects in a managed environment. These aspects are represented in the "Common Models" created to address system, device, network, user and application aspects. The CIM is comprised of a specification and a schema. The CIM Specification defines the details for integration with other management models, while the CIM Schema provides the actual model descriptions. The CIM Schema captures the notions, which are applicable to all common areas of management, independent of implementations.

CIM itself is structured into three distinct layers:

- A "Core model" that captures the different notions, which are applicable to all areas of management.
- A "Common model" to capture the notions that are common to particular management area (systems, applications, networks and devices), but independent of a particular technology or implementation. The Core and Common models together are referred as the CIM schema.
- The "Extension schemas" to represent the technology-specific extensions of the Common model, for example in operating systems (UNIX or Microsoft Windows).

Thus, CIM is a conceptual model that is not bound to a particular implementation. This allows it to be used to exchange management information in a variety of ways.

One of fundamental aspect in CIM is the ability to exchange information between management applications. The current mechanism for exchanging management information is the Management Object Format (MOF) [9]. MOF defines the meta-schema (a formal definition of the model) to be used to represent the syntaxes and semantics aspects of the model. The meta-schema defines the basic object-oriented concepts: classes, relationships, properties, methods, operations, inheritance, associations, objects, cardinality and polymorphism. A CIM-capable system must be able to import and export properly formed MOF constructs.

The Web-Based Enterprise Management (WBEM) [10] proposed by the DMTF (Distributed Management Task Force) is a set of management and Internet standard technologies developed to unify the management of distributed computing environments. WBEM provides the ability for the industry to deliver a well-integrated set of standard-based management tools, facilitating the exchange of data across otherwise disparate technologies and platforms. It defines a standard common model (i.e., description) and protocols (i.e., interface) for monitoring and controlling resources from diverse sources.

An important part of WBEM is the Common Information Model (CIM), a standard for defining device and application characteristics so that system and network administrators and management programs are able to control devices and applications from different manufacturers or sources in the same way. WBEM standards provide a Web-based approach for exchanging CIM data across different technologies and platforms. CIM data is encoded using Extensible Markup Language (XML) [11] and usually transmitted between WBEM servers and clients using the Internet's Hypertext Transfer Protocol (HTTP).

The WBEM is designed to be extensible, allowing new applications, devices, and operating systems to be specified in the future. Open-source implementations of WBEM are available from several vendors, including OpenPegasus, OpenWBEM, and WBEMsource. WBEM is said to be particularly appropriate for storage networking, grid computing, utility computing, and Web services.

The Directory Enabled Network (DEN) [12] initiative is designed to provide the building blocks for more intelligent management by mapping concepts from CIM (such as systems, services and policies) to a directory, and integrating this information with other WBEM elements in the management infrastructure. This utilizes existing user and enterprise-wide data already present in a company's directory, empowers end-to-end services, and supports distributed network-wide service creation, provisioning and management.

The use of CIM in defining a directory schema enables consistent schema for, and a common understanding of, directory information. Common schema and semantics are especially important when defining and decomposing platform-neutral, high-level policies.

The DMTF DEN Special Interest working group is focused on communicating the benefits of DEN as a key component of the DMTF's management standards. It is working at two levels, as the first one is to use a directory FIRST to "direct" management clients to relevant services, and to hold a subset of management data; the second level is to specify the directory schema (LDAP mappings) for DMTF's CIM Version 2.5 and later releases. Specific modeling and mapping efforts are addressed in the DMTF's LDAP, Network, Policy, and User and Security Working Groups.

The DEN-ng is designed to provide a rich and extensible classification of managed entities. It overcomes UML limitations by linking to ontology's, as well as specifies and design autonomic architectures and generates code dynamically for managing autonomic systems. The DEN-ng models the aspects of a ManagedEntity; other models represent a ManagedEntity as an atomic object.

The DEN-ng object-oriented information model provides a cohesive, comprehensive and extensible means to categorize and represent things of interest in a managed environment, including users, policies, processes, routers, services, and anything else that needs to be represented in a common way to facilitate its representation and management.

The DEN-ng information model defines the static and dynamic characteristics and behavior of these managed entities as independent of any specific type of repository, software usage, or access protocol. Note that the explicit use of dynamic models differentiates it from other current management efforts.

The DEN-ng uses dynamic models to represent the life cycle of managed elements. Many different stakeholders are required to work together to build a product. However, they all have different perspectives on how the product works.

This means that one concept might mean different things to different people. For example, when a business analyst looks at an SLA, that person thinks of contractual obligations and different options for realizing revenue.

The SID (Shared Information Data) [13] is suggested by the NGOSS (New Generation Operations Systems and Software) [14], which provides a "common language" for software providers and integrators to describe information management. SID is used to run business processes and enable reporting; companies can create rules that describe how data must be created and used in SID.

The SID is an object model, which uses Unified Modeling Language (UML) [15] to define the entities and the relationships between them, as well as the attributes and processes (termed methods) which make up the entity or object. SID provides the NGOSS "glue" in giving a representation of different views: Business view, System view, Implementation view and Deployment view. These views are necessary to ensure that business requirements can drive system design and implementation. The SID focuses on modelling network elements and services covering a business, system and implementation viewpoints.

Furthermore, some open issues exist today and concerning the implementation of services with this model. The service activation methods are not described and there is no knowledge of the network itself. Here the notion of Knowledge represent only a Database or Repository of sharable information about model, but does not give us the opportune and needed information to assure a correctly running of service in real time. This includes tools and guidance for service providers, suppliers and systems integrators, with the definition of a Business Process, Systems and Software integration “maps”, and the development of an architecture and Knowledge Base or Repository of documents, models and reference code to support developers, integrators and users. Their principal goal is to provide a rapid development of flexible, low cost of ownership solutions to meet the business needs of the Internet enabled economy.

The NGOSS methodology is addressed by a Business Process Model-based Viewpoint where SID represents the key to Interoperability based on the shared of Business Process and Experience Driven. SID is considered such as the NGOSS Meta-Model for Shared Info & Data.

A consideration to be taken is to search how to have a better knowledge of a service to manage it correctly. For years, it has been gotten information through databases that storage the information required by the system, that is the case of the SNMP [16] protocol. It has been one of the first works where there is a database for monitoring the network.

Another solution is the CMDB (Configuration Management Data Base) [17]. It proposes structured databases that conform to the ITIL (Information Technology Infrastructure Library) [18] norm. It contains a repository of information for components in an Information System. This approach allows different configuration management processes to share data, but requires a lot of resources to create and maintain the integration of all the information.

For years, the human has processed information with the help of external tools (graphics, word processor, etc.); nevertheless, with the increase of volumes of information and computer technology evolution it becomes more difficult to manage and storage it. A new solution has been deployed named Data mining [19]. The goal of the Data mining approaches is to get knowledge from simple data patterns collection. The term of Data mining is often related to two processes, the knowledge discovery and the prediction. This solution is employed by enterprises desiring to know if their business model works correctly, or to find hidden patterns from data. However, this solution doesn’t allow taking decisions in real time, because at the beginning there is a static collection of information and according to rules determined, we can thus obtain the knowledge desired.

All the cited works present the good efforts to provide the conceptual and informational views about the entities to be managed. Therefore, some open issues are still present concerning the methods and mechanisms to implement these models and also to deal with the deployment and dynamic maintain of services deployed. The models cited study more the functional aspects but don’t include behavior and non-functional aspects, which are actually necessary to treat the end-to-end service QoS associated to a service cycle-life. In fact, what we want to have as information becomes important when is used to take decisions; the problem is to establish a knowledge base working as an inference. This knowledge base should automatically update and find the relevant decisional information to take decisions dynamically.

3 Background

In order to have a better comprehension of this paper’s driving concepts, we present the previous works, which are finalized by the Telecom ParisTech INFRES-Lab. The lab is engaged in the dynamic service management framework design [20] [21] [22].

To be specific, the framework concerns first, the architectural models (§3.1) which give out a global and full-scale image of the existing heterogeneous environment; second, the informational structures (§3.2), which provide the basic generic structures to ease the integration of the different information.

3.1 Architectural Models

The focus point of our concepts is how to model the telecommunication world in order to have a generic representation. Our modeling is conceived through *Abstraction*, which is defined by universal encyclopedia to structure the data according to simplification, generalization, selection and schematization. Therefore, the proposed Meta model (Figure 1) of NLN (Node-Link-Network) includes the following three abstracted elements:

The *Node* is defined as an entity, an element that is responsible for a specific process.

The *Link* is the representation of the interaction between two nodes. It can be considered as a virtual communication channel between two ends. It designates any component offering its transfer capacities in order to provide the nodes a support of interconnection.

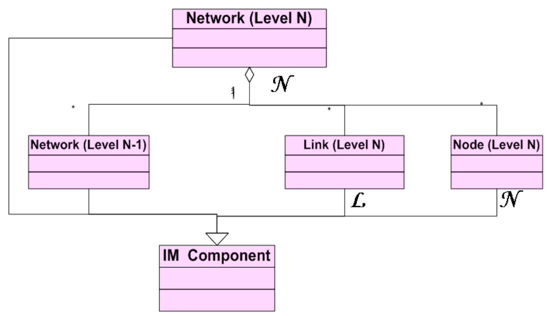

Figure 1: Meta Model: NLN

The *Network* is a set of nodes and links offering a global service in a transparent way. It allows the nodes and the links cooperate in order to offer a certain service. It is defined through four visibility levels. (User, Terminal, Network, and Service). [23][24].

By applying the NLN Meta-model, a visibility level can be represented as a network of components, while a link responds to a communication between two visibility levels. The Meta-model enables the modeling to represent the real world. As we can see in Figure 2, each visibility level represents a network (horizontal), which could be an *equipment network* (Figure 2-1), a *provider network* (Figure 2-2), or a *service network* (Figure 2-3). In every network, there is a behavior vision for each component. This helps when a component can't fulfill any more its SLA or the user's required QoS: a counterpart that respects those requirements can be found to replace the current component. With the help of these networks at different levels, the user's preferences and his required QoS can be satisfied (Figure 2-4).

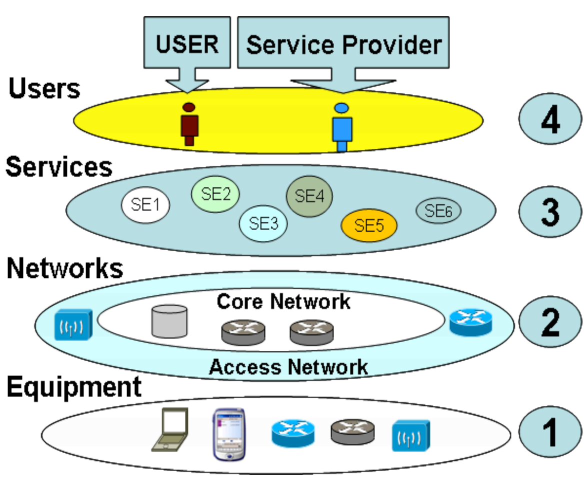

Figure 2: Modeling vision

Besides the horizontal view of this modeling vision, it is possible to have a vertical network composition, which allows the establishment of a session according to the user preferences.

For example, at the equipment level (Figure 2-1) and in the network of the equipment level (Figure 2-2), we can apply the user preferences using the VPIN to choose equipments and the access network in order to offer the required service, providing that the components fulfill the user's required QoS.

Facing the heterogeneous environment, the importance of this Meta-model is the decomposition of the whole telecommunication world into several abstraction levels according to the different service levels, which permits management system to be complete and thorough. Therefore, applying the Meta model at each decision level, we can thus obtain an architecture model (Figure 3), which covers the ambient context. The horizontal and vertical relationships (Figure 3 - 1) are guaranteed thanks to the composition and recursiveness features provided by our model. In fact, each level of visibility is modeled in the same way. At the service level (Figure 3 - 2), service components are managed as service overlay according to their types (SCU, SCA, SCN). Together with the virtual links, the service components are linked by the service logic to form a VPSN (Virtual Private Service Network) [20], which is essential for handling the overall service session. The user selected network from the VPSN represents a complete use centric service.

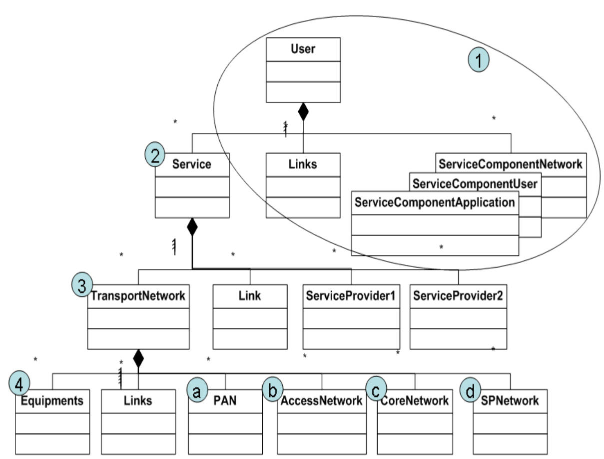

Figure 3: Architecture Model

The VPSN stays always on a transport network (Figure 3 - 3) according to the user's geographical location. As the services are always supported by the service providers, the latter is represented also in the model in order to have all the potential service suppliers at anywhere user arrives.

In fact, the VPSN is supposed to support the four types of mobility in the NGN (that we previously pointed out) within a single E2E connectivity:

- The user mobility, i.e. the capacity of a target user to move from one terminal to another, is supported by the PAN (Personal Access Network) node (Figure 3 - a).
- The terminal mobility, i.e. the capacity of a given terminal to move from one access network to another, is supported by the AccessNetwork node (Figure 3 -b).
- The network mobility, i.e. the fact that the access network itself is moving, it can be supported jointly by AccessNetwork node (Figure 3-b) and CoreNetwork node (Figure 3 -c), and finally,
- The service mobility, i.e. the possibility to replace a service component by another (for example, a nearby service component which is more suitable than the current remote one). This type of mobility is supported by the mode of SPNetwork (Service Provider Network) (Figure 3 -d).

Note that all these four nodes (Figure 3 - a to d) have self-management capabilities to preserve the offered service. The links between these nodes insure interactions between them to assure the service delivery to be continuous throughout the user demanded global service. The E2E service can thus be dynamically maintained by linking certain equipments in the equipment level (Figure 3 - 4), where the routers as the hosts are the equipment nodes, the cables are equipment links and the whole is the equipment network.

3.2 Informational Structures

For efficient representation of the real world, we must have a uniform information structure containing the relevant and synthetic information to make the right decisions at the right place at the right time. It means having both the description information of all resources, but also knowledge of the behavioral aspects, i.e., whatever is on the QoS. The informational model we defined is generic and abstract to describe any ambient resource. In order to have a dynamic management, the "Real Time Profile" (Figure 4) is instantiated in real time and it refers to an element that may belong to any management level.

The component of "Real Time Profile" will have in an instant "*t"*, one of four states, these states are: *Una*vailable, *Available, Activable* and *Activated* (Figure 4-1). The state *Unavailable* means that the resource is temporary or permanently inaccessible. The state *Available* means that the resource is or can be accessible. The state *Activable* takes into account certain logical conditions, which are necessary to assure the activation of a component. For example, concerning the authentication of a user, when he introduces a correct login/password, the resource becomes *Activable*. The state *Activated* means that the resource is being used.

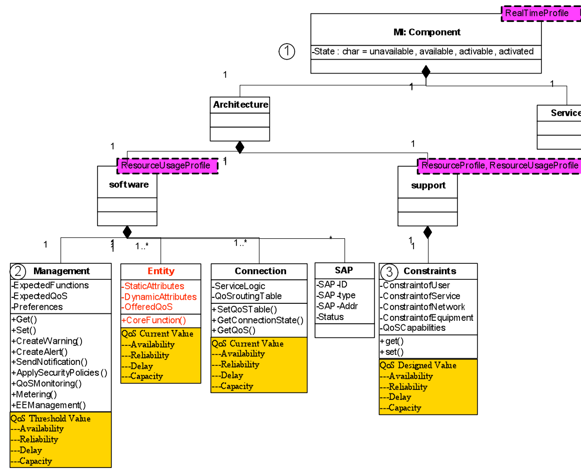

Figure 4: Real Time Profile

We have self-management in each resource by extracting information from the resource profile and resource usage profile [20]. Concerning the component behavior, it is necessary to have a homogeneous expression of its QoS to evaluate the end-to-end behavior. The behavior of each component is obtained by measurable QoS parameters that can be categorized according to four criteria: *Availability, Reliability, Delay*, and *Capacity*. The self-management is done according to this QoS model.

The management class (Figure 4-2) contains *QoS threshold values* (it indicates the limit of a node's normal operation or a link's normal interaction realization, in normal condition of the service exploitation and usage) concerning a required service for QoS self-management. The entity class has the *QoS current values* (it indicates the current status of the node's treatments and link's interactions. This kind of value is to monitor, during the exploitation, by the provisioning in order to have a real time image of the service behavior.), which provide information for the constraints class (Figure 4-3) which includes the *QoS conception values* (it's decided at the phase of service conception. It introduces the maximum possibilities of the node's treatments and the link's interactions.). These classes verify that the service can be offered. This QoS information helps to support the management treatment and the decision-making process of the component.

In order to integrate the personalization, we have also proposed *User profile*. It includes several sub profiles: *"General Information Profile"* which indicates all the information about his resources; *"Location Profile"* which describes each resource according to different location and *"Agenda Profile"* which describes each resource according to diverse activity. The user preferences are applied when sorting and filtering the information for each kind of profile.

4. The VPIN

The VPIN (Virtual Private Information Network) has been proposed [1] to have the whole information concerning the user information system in real time, taking into account his location and activity. The fact of getting the complete information allows having knowledge of the component that conforms the user system information. This collect of information is known as a knowledge base, which allows taking decisions according to the user needs when accessing a service. In order to have the whole user system information, the VPIN is organized in four visibility levels that will be detailed (§4.1). These visibility levels permit to get the information of a resource and its behavior (QoS) at an instant "t". At every level, a network of those resources is formed, which allows having knowledge of the behavior of the resources (QoS). The goal of forming a network at each visibility level is to have the knowledge of the resources available to be used when the user wants to access to a service and establish a session. Therefore, if there is a degradation of the QoS in a component, it could be replaced by another one that fulfills the expected QoS. The change of a component can be fulfilled with the help of the Real Time Profile (§3.2), which is present at every resource and at every visibility level. It's for this reason that the Real Time Profile has been chosen for the proposition. In order to see the information contained in the Real Time Profile, there will be an example of the relevant information (§4.2) that is used in a resource.

4.1 Organization of the VPIN

The VPIN allows organizing the information in a structured way. This organization enables get the pertinent information that is needed in right time and the right place when needed. This organization follows the visibility levels in order to have the whole vision of the user information system. This is applied at every visibility level (User, Terminal, Network, and Service) based on the Meta-Model NLN explained before (§3.1). Applying the Meta-Model NLN to the network level, we can say that the node is a router and the link is the way they are connected, in this case it will be through the routing protocols, and the set of these nodes and links form the network at this visibility level.

The organization of this VPIN (**Figure 5**) offers a global vision of all the components that constitute the user centric information system. The benefit of this organization is that, the modeling structure is the same image of the real world, which allows having pertinent information in a real time. This organization is user centric, it means that it has been conceived to fulfill the user needs, breaking down technical barriers that avoid a user to access his services.

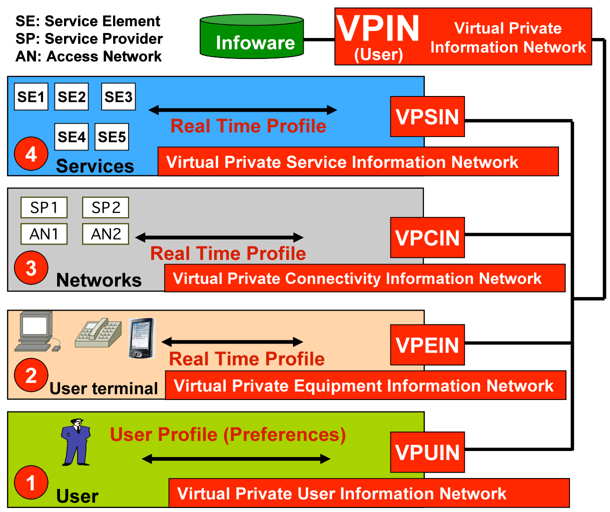

Figure 5: VPIN organization

The organization of this VPIN is done in four levels. At the first level, there is the VPUIN "Virtual Private User Information Network" (Figure 5-1). In this level, there is all the information about the user profile where are included the user preferences. These preferences can change according the user location and activity.

In the second level, there is the VPEIN "Virtual Private Equipment Information Network" (***Figure 5***-2). It is composed of all the equipment information belonging to a user. The equipment information referred is not only that which belongs to the user, but the equipments that are available and can be used by the user according to his location.

As third level, there is the VPCIN "Virtual Private Connectivity Information Network" (***Figure 5***-3). It concerns all the access networks information from which user can be connected. The particularity in this level consists that it can be found the access networks from operators as well as service providers.

In the fourth level, there is the VPSIN "Virtual Private Service Information Network" (***Figure 5***-4). It provides the service composition information related to a user. An example of this level could be, when a user is located at the airport and desires to buy a ticket with his PDA. There is a server that broadcast this information, but if this server stops working, in the VPSIN, there is going to be another server that takes over the task. Hence the user should have always the necessary information about this service.

4.2 Relevant information

As it has been showed (§4.1) the organization of the VPIN allows having the necessary information at every visibility level. But, instead of keeping the allocation-related information in the VPIN, we have decided to have all the information necessary, sufficient and persistent in our knowledge base.

It's within the *Real Time Profile* (described in section 3.2) that there is the QoS information, which is present in every component and in the visibility levels as VPEIN, VPCIN and the VPSIN.

The *Real Time Profile* is capable of providing all the information of a component that conform the user information system. When this information is collected, it allows making the adequate decisions when a user-session is established. The users preferences are considered to take the decisions. They are not only considered at the time of the establishment of the session, but also during the entire session. This makes possible the dynamic session management in real time. The session components are terminals, access networks, core network and services that the user desires to employ according to the QoS component and preferences. In these components, we have to take into account what information needs to be included in each visibility level for the dynamism of the user-session. According to the Real Time Profile structure (Figure 4), we display the relevant information for a PDA (Figure 6) and the service component "E-mail" (Figure 7).

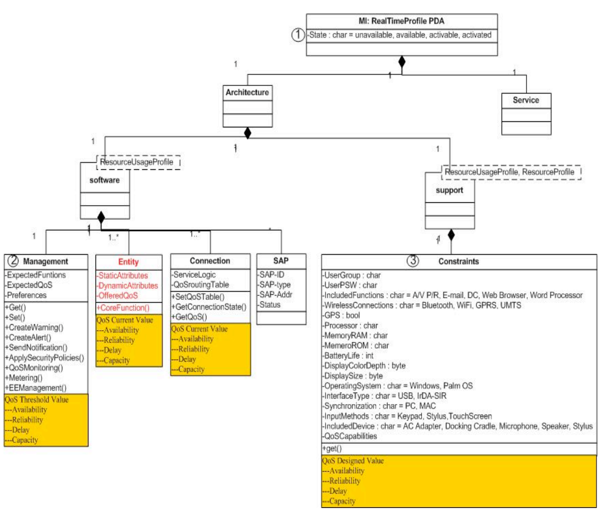

Figure 6: Relevant Information in PDA

These examples can demonstrate that even if there are different components, the information structure is the same. In these components also includes the QoS information, which allows them to know the behavior of the component.

In this Real Time Profile we focus in the third part (Figure 6-3) from which we have the constraint class of the PDA. As we can see, the representation of the information of the PDA follows the same structure given by the Meta-Model NLN (§3.1) and the information that is contained in the constraint class follows the same structure about the visibility levels. It means, there is information that concerns the user, terminal, network and services in the respective order.

The "E-mail" service component (Figure 7) is represented as information model in order to identify the relevant information. In this component, there are displayed the constraints of this service component concerning the user (login, password), equipment (10 MB to be used), the network (protocols POP3, SMTP, IMAP) and the service (text to vocal converter).

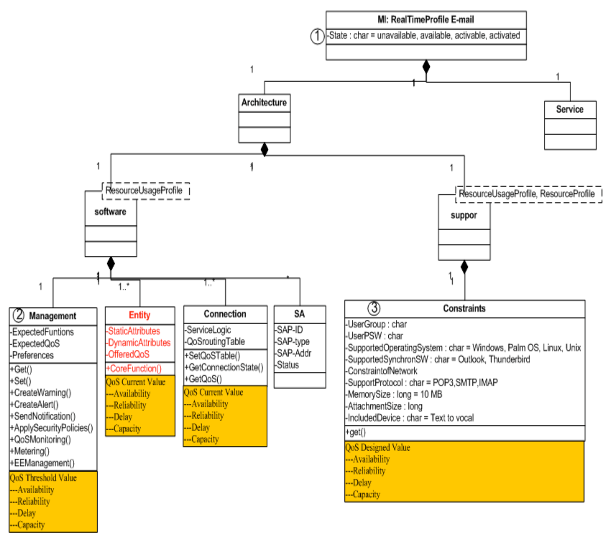

Figure 7: Relevant Information in service component "E-mail"

On of the advantages of having the same structure in a component is that, it allows a homogeneous correlation of the information that is stored in the component. In that way, the information of that component can be demanded at any moment with the security that the given information is completely trusted.

5. VPIN: Functional dimension proposition

After a briefly explanation about the organization of the VPIN (§4), the next aspect to consider is the functional issues in the knowledge base. That is, how to identify the factors that makes changing the information that is stored in the knowledge base. This knowledge base is the representation of changes that comes from a real environment. Another issue to treat is, what information will be modified due to this change in the knowledge base. It is through the information change that an event is triggered.

In this section, we propose the functional dimension, in which we will verify how to identify the inferential events (§5.1), and how this knowledge base going to react when a modification of the information (inference) occurs (§5.2).

5.1 Inferential Events

The VPIN contains relevant information, including QoS information about the component. The collection of this information makes the knowledge base to be decisional. When a change is done in the knowledge base, there is the need to check the architecture to verify the impact that is affected in the knowledge base, therefore, according to the structure, the inference will be done. Nevertheless, this information needs to be updated when the QoS information of a component changes. With the purpose to always have the component adapted to the user needs, we need to know how to react, internally, when there is a change of QoS in a component. That is why the knowledge base can be decisional, due to the management QoS of each component.

In order to detect the causes that make produce a inference in the knowledge base, we identified two kinds of events, the event that is triggered when the QoS in a component changes (*QoS change event*), and the other type of event is produced by the change of state of the component (*state change event*). These two kinds of events are integrated in an agent that is integrated in the Real Time Profile; we called this agent "EMA", which means Event Monitoring Agent.

QoS change event

We have identified an event that makes having changes in the knowledge base. This event is triggered due a change of QoS in a component. We have called this event as *"QoS change event"*.

An example of this king of event could be, when a terminal (PDA) that is used in a session arrives to the maximum capacity in the RAM memory; therefore, the terminal cannot be used anymore. Thus, due to the change of QoS in the component, the QoS change event launches the trigger (Figure 8).

Therefore, an action must be taken; in this case, it will be a change of terminal, for example, changing to the PC terminal.

In that way, the terminal that has failed won't participate anymore to the user-session, due to the internal problem.

It is with the help of the EMA's agent (Event Monitoring Agent) that the behavior of a component can be monitored

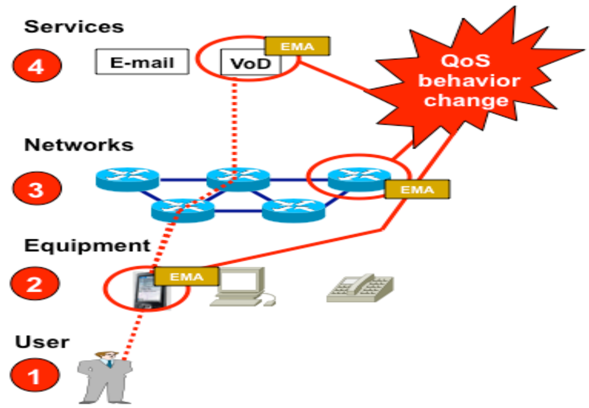

Figure 8: QoS change event detected by EMA

The main focus of the *QoS change event* is the change of QoS, but that is not the only cause in a component that an event has to be triggered.

The other kind of event works through the change of state in a component. We called the second kind of event as *"state change event"*.

State change event

We identify that the "*state change event"* can be caused by a change of state in the component. That is, unavailable, available, activable and activated.

It means that if a component changes the state from unavailable into available, that is a change where the QoS of the component doesn't intervene at all. It's only the state of the component that changes.

The EMA's agent can be used for the *QoS change event* as well as the *state change event*. The difference is that when the QoS of a component changes, it is through the "*QoS change event*" that an action will be taken. In the other hand, if the EMA's agent identifies that the component changes of state, in that case, it's through the *"state change event"* that an action will be taken.

As already explained (§1), the different types of mobility; that is mobility of the user, terminal, network and service. The *state change event* can be produced, due to the different types of mobility. That is, when a terminal moves to another place, maybe it will not be connected to the same access point, nevertheless, it doesn't mean that the terminal had a change of QoS, the terminal has the same capacities, but as it has been moved, a change of the state has been produced.

As showed in (Figure 9), we present an example where the user mobility is invocated. As at every component the EMA's agent is present, when the user moves, the event is triggered when the user tries to login in the other terminal. Therefore, there is a notification that the PDA is no more used for access the service. Therefore, the terminal to use will be the PC. This change of terminal makes an inference at the network and service level. The change is that as the terminal equipment is different (from PDA to PC), a new routing path must be built in order to continue to access the service.

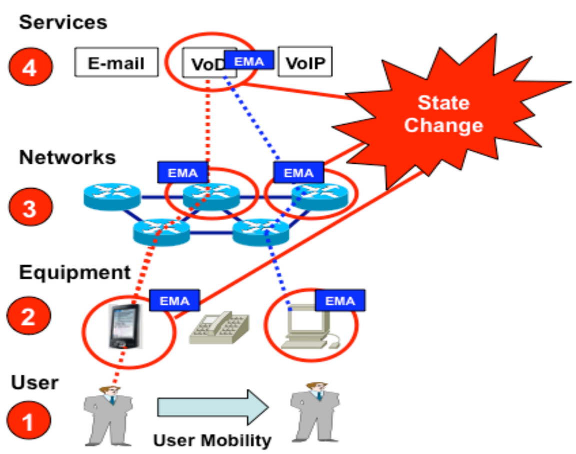

Figure 9: State change event detected by EMA

The inference at the service level will be, the change of the address from the router that receives the traffic. Therefore, another router will send the traffic to the Video over Demand (VoD) service.

Once identified the factors that makes the inference in the knowledge base, we show how this is achieved through the informational model.

As showed in (Figure 10), in the Real Time Profile the agent is placed at two points, when a change of QoS in the component is done, and when the state of the component changes.

The *QoS change event* is triggered through the management class (Figure 10-1) is produced. In the Management class, it exists the "QoS Monitoring" operation; the function of this operation is to monitor the component in a real time. There is also the "Send Notification" operation; the goal of this operation is to send a notification to which it concerns when the event is triggered.

Considering the *state change event*, as mentioned before, this event is triggered when there is a change of the state in a component. The agent is located at the beginning of the component, where the sate of the component is treated (Figure 10-2).

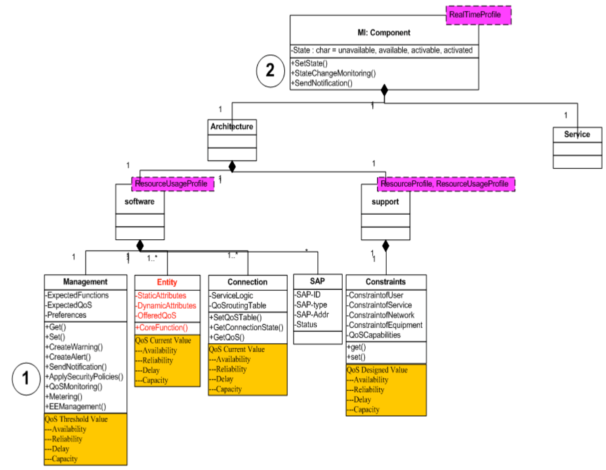

Figure 10: QoS change and state change events managed by the agent

The change of state can be caused due a mobility type. Thus, the operations that are at the beginning of the component are three, the first operation is "Set State", the second operation is "State Change Monitoring", and the third operation is "Send Notification". These three operations allow monitoring the component in real time. The "Set State" operation is used to set the state of the component. The "State Change Monitoring" verifies and updates the sate of the component, and the "Send Notification" operation send the notification to which it concerns.

The agent that is in charge of the events (by component and by state) is directly integrated in the Real Time Profile (Figure 10) in the management class and the state class respectively.

5.2 Inferential management

According to the events identified before (§5.1), it can be detected the different causes of information change in the knowledge base.

Considering these information changes (inference) in the knowledge base, we have to be able how to treat them, and what changes in the knowledge base are done. It means, when the behavior or the state of a component have been changed, the information of that component is registered in the knowledge base. Thus, according to this change, there is an update of the information that has a relation with that component.

First, we focus in the events that are produced due a change of the behavior in a component. As showed in Figure 11, there is the representation of the Real Time Profile as a component (right side) and the Real Time Profile when it represents a network (left side). The network Real Time Profile represents the components at the same visibility level.

As an example, we can say that the component is a terminal (PDA), and that the whole terminals available, form the network of terminals (PDA, PC, Telephone, etc.). Thus, starting with the component behavior, there is the management class (Figure 11-1), which monitor the component internally. When a behavior change is done in the component, and if the management class can't treat this change, there is a notification that is sent through the service class (Figure 11-2) to the Real Time Profile Network. Note that the service class is the interface by which the notifications are sent or received from a component. When the notification is received through the service class (Figure 11-3) of the Real Time Profile Network, it is forwarded to the management class (Figure 11-4), which allows treating the event notification that has been sent from a component. After, the management class notifies to the Entity class (Figure 11-5), and it makes an update about the QoS information that is contained in this Real Time Profile.

Once showed how the inference is done when a QoS behavior in a component changes, now, we show how the inference occurs when there is a state change in a component.

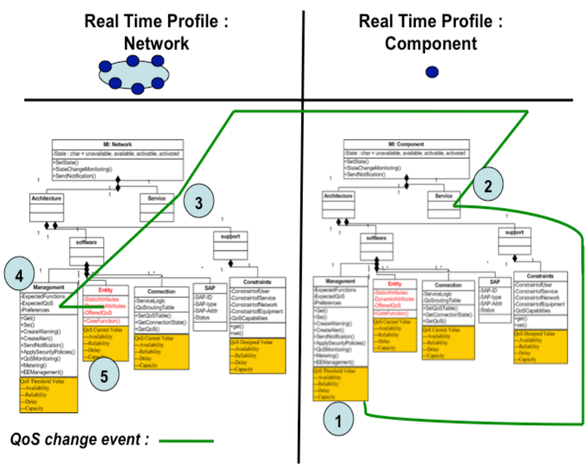

Figure 11: QoS change event inference

Following the same architecture of the Real Time Profile (Figure 11), which can represent a component, or a network, we show in Figure 12 how the inference is treated when the state of a component changes.

When the state in a component changes, the QoS management could not be notified about this change. For example if a component as a PC is being used, and if there is a cut of electricity, the components will be unavailable, even if the QoS of the terminal was working properly. Therefore, it is demonstrated how the inference in that case will be treated.

When a component changes of state, that could be due to the mobility of the terminal, network, service, and for the user mobility, the change of state is produced when the user tries to access to a service through another terminal.

Note that the operations that indicate, set and update the state of a component are at the beginning of the component.

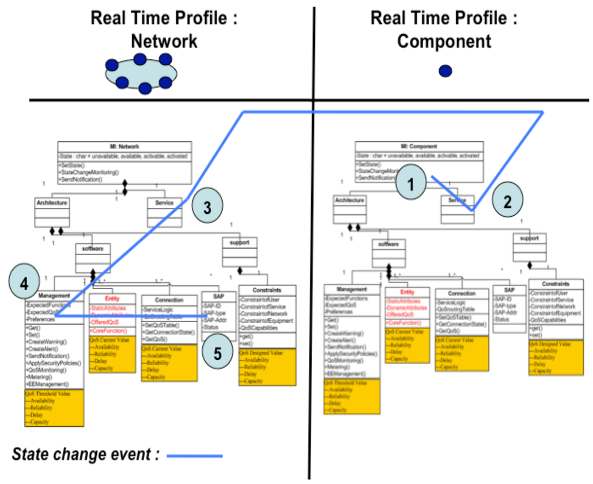

Figure 12: State change event inference

First, when a change of state in a component is done (i.e. available to unavailable), there is a notification sent (Figure 12-1) from the component class to the network Real Time Profile. This notification is sent through the service class (Figure 12-2) of the Real Time Profile component to the service class (Figure 12-3) of the Real Time Profile Network that contains the set of elements that conform the network of the components. Once the service class of the Real Time Profile Network receives this notification, it will be forwarded to the management class (Figure 12-4). With the help of the QoS management that is integrated, the notification can be treated. Consequently, the management class will notify the SAP class (Figure 12-5). The SAP class has the whole list of address of the components that conform the network of components. Therefore, when the notification is received to this class, the address of the component that became unavailable will be erased from the list. This inference can produce an internal change of the Real Time Profile Network, due to the unavailability of the component, a recalculation of the QoS need to be done to have the real information of the components that are available.

It is with the help of the event manager agent (EMA), which is present at every component, that a control of the changes done in the knowledge base can be achieved. Thus, in this way, the QoS information that is stored is always the real image of the components that are available.

6. Experimentation

We describe a scenario with the purpose of presenting the feasibility of implementing the concepts proposed about the events management.

We depart from a scenario where is presented a real world vision (§6.1), and how this scenario function in a platform which the software to do it is Oracle Database (§6.2).

6.1 Real vision scenario

The real-world vision of the scenario (Figure 13) is presented to separate the components that take part of a session throughout different levels.

The goal of the scenario is to ensure an end-to-end user-service session in a seamless and continuous way despite the user's mobility, with the help of the relevant information. In Figure 13-1, the user has to his disposition a Personal Computer (state Activable), a telephone (state Activable) and a PDA (state Activable). At instant “t”, the user checks his e-mail on his PC; the choice of using the PC follows the user preference, which the intention to use the Personal Computer (state becomes Activated) is due to the resolution in the screen. Therefore, in that moment, the user’s VPIN is set according to the user location, which in this case is the office. The current session is supposed to be supported by the service composition including SE1 (authorization), SE2 (e-mail), SE3 (SMS) and SE6 (E-mail client). At the end of the day, the user has to go home and the transport used is his car.

As the user mobility is produced, there’s a terminal exchange based on the user’s preferences. According to user's preferences, he desires to continue checking mails in his PDA. At this instant, the states of the equipments are Personal Computer unavailable, telephone unavailable and PDA activated. The user consults his PDA (SE4), which implies an adaptation to continue checking his e-mail. Once this change has been done, it could be dangerous to the driver while he reading his e-mails (Figure 13-2), therefore, according to his preferences previously configured, the service delivery changes. From now, the e-mail text is translated into vocal SMS. This implies a new component in the terminal (SE5), allowing user to listen to his e-mails while driving. In order to make the adaptation of the service component “text to vocal”, verification should be done to check whether this new component could be supported by the PDA. For the user, all the adaptations according to his service were transparently done with the help of the information that is contained in the VPIN. In this case, the VPIN acts as an orchestrator in choosing adequate service components according to user preferences to ensure the end-to-end service of the user.

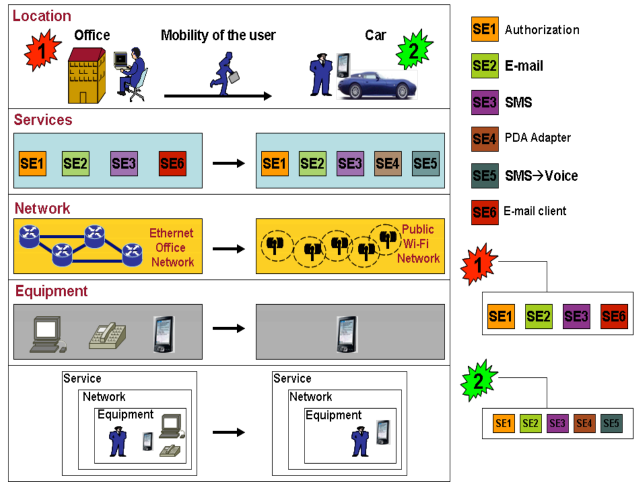

Figure 13: Scenario

This scenario demonstrates a real case of how the services that are employed by a user. To achieve this scenario, the information that is contained in the VPIN has to be the same that reflect the components that are used by the user, thus, the user will access services in a continuous way without interruptions.

6.2 Functional process

In order to demonstrate the proposed concepts about the inferential aspects, we have implemented a structured relational database on the Oracle 11g platform. This implementation demonstrates the first works to implement the *knowledge base*.

Based in the description of the scenario (§6.1), the test to demonstrate is how the inference is taken into account in the knowledge base due to the user mobility. The user mobility implies a change of the terminal for access the service.

The structure proposed is based on the Real Time Profile (§3.2). This structure allows having the knowledge of the whole user-session when the user accesses a service, no matter the user location.

The realization of this implementation has been developed in three steps.

The first one is to create a database structure that reflects the Real Time Profile. As second step, we create the triggers that launch the inference when information is added or updated. At the moment of the creation of the trigger, the process is launched, thus the trigger is ready to be executed when the condition is accomplished. The third step is to feed this database with real information, and as the triggers are already launched, when the information is added or updated, the trigger will be executed.

As the firs step, a creation of the structure in a database has been done (Figure 14). This structure allows having the structure where the information of the component will be stored.

```
"Table MI :"
- Create Table MI (
 MI_id NUMBER (10),
 State VARCHAR (12),
 Archi_id NUMBER (10)
REFERENCES Architecture (Archi_id),
Service_id NUMBER (10)
REFERENCES Service (Service_id),
 PRIMARY KEY (MI_id));

"Table Architecture :"

- Create Table Architecture (
 Archi_id NUMBER (10),
 Soft_id NUMBER (10)
REFERENCES Software (Soft_id),
 Support_id NUMBER (10)
REFERENCES Support (Support_id),

"Table Service:"
- Create Table Service (
 Service_id NUMBER (10),
 PRIMARY KEY (Service_id));

"Table Software:"
- Create table Software
 (Soft_id NUMBER (10),
 Management XMLTYPE,
 Entity XMLTYPE,
 Connection XMLTYPE, SAP XMLTYPE);

"Table Support:"
- Create table Support (Constraints XMLTYPE,
 Support_id NUMBER (10));
```

Figure 14: Oracle's database structure

We propose a series of events allowing the Database to react achieving the inference as described before (§5). These events help to adapt the information according to an event occurred in the user-session. This will avoid the service interruption when user is accessing the service. The tools used for this implementation is Oracle Data Base and SQL Developper for Oracle (**Figure 15**). Theses graphical software tools allow simplifying database development tasks.

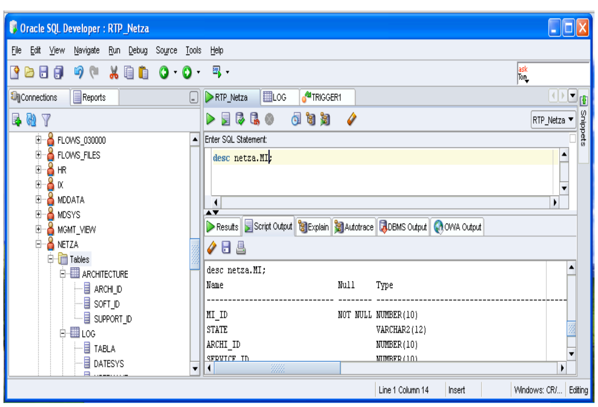

Figure 15: Oracle SQL Developer

In oracle databases, there is the possibility to create process that allows launching actions in a database. The Triggers are programmed internally in the database, thus a space is entirely reserved to store the Triggers. The triggers make possible to modify necessary information when a change in the database has been done. In our scenario, the change of the state is going to produce an action. This permit the inference in our knowledge base through the events previously defined (§5). For creating a trigger, it is necessary a CLI (Command Line Interface) in order to indicate by a command what table to monitor as well as the action to be taken when a change is done in the table.

In the **Figure 16**, there is illustrated an example of the creation of a Trigger in Oracle.

```
CREATE OR REPLACE TRIGGER Netza.event
AFTER INSERT OR UPDATE ON NETZA.ARCHITECTURE
BEGIN
 INSERT INTO LOG (Tabla, Datesys, username, action)
VALUES ('NETZA.Architecture', sysdate, sys_context('USERENV','CURRENT_USER'),
 'Change done in Table Architecture') ;
END;
```

Figure 16: Creation of Trigger in Oracle

The trigger showed in the Figure 16 describes the creation o a trigger. The name of the trigger is "netza.event", and the factor that is going to launch an action is when there is an insertion or update of information in the table "Netza.Architecture". When this condition is accomplished, the action to be taken is to insert a data in a table called "LOG". The content of the Log's table can be defined previously or generated dynamically with the help of the declaration of variables. In this case, in the log table, we make insert the name of the table (netza.architecture), the data of the modification, and the user that has done the modification; as well a message saying that there was a change in the table.

The third step to do is, the insertion of information in the data base structure. With the insertion or update of information, this modification will imply the react.

We have configured some triggers to react in the database, hence if the resource information changes because of behavior, a reaction can be taken place according to the event. The triggers configured, can represent the user mobility, i.e., when the user moves to another place, it's at the moment of the login in another terminal that the event is triggered and the inference occurs.

7. Conclusion and perspectives

In this paper, we analyze initially the user new requirements for accessing services in the Next Generation Network's context. These new requirements are confronted with the increasing development of a heterogeneous environment in equipments, networks and services, as well as the mobility (equipment, network and service). This new paradigm leads, as result, the impossibility of dynamic service adaptation according to user preferences. The latter inspired us to regard the user as the central of the whole system and reconsider the Information System in a User Centric way.

We regard the different solutions proposed by different standardization organisms and point out the lacks concerning the use requirements. We perceived that the pertinent information related to the context NGN (heterogeneity, mobility, user-centric) are not taken into account. Therefore, based on models from Telecom ParisTech, the QoS agent and the state agent, we have introduced the inference events.

Hence, we have proposed our knowledge base (VPIN), which is an information model based capable of dynamically managing the information. Architecturally, it is different from the existing proposals thanks to its application-independent characteristics due to basing on a modeling of the real telecom world.

Functionally, it provides the decisional information as QoS information for a certain user session, and takes into account the user's preferences as an important filter of the information. With the help of this knowledge base, we can make the necessary decisions by the reaction to the events in order to always conform to the user desired service with his QoS. We studied, in the first step, the events activated by the change of QoS and those activated by the change of state. The different agents proposed allow the management of events that can occur during the user-session. Two detailed examples are illustrated to detail how the inferential management works.

The implementation has permitted us to examine the feasibility of the event driven aspect, i.e. how to manage the studied event in the Oracle environment, with a created VPIN in the database. What remains as the future work is the fully development of the event agents which can include all the real time events, and test them in a real time platform in order to evaluate the performance of our proposals.

Acknowledgments

The authors would like to thank Gladys Diaz, Ken Chen and Xiaofei Xiong for their contribution in this work.

8. References

[1] Netzahualcoyotl Ornelas, Noëmie Simoni, Ken Chen, Antoine Boutignon, VPIN: User-Session Knowledge Base for Self-Management of Ambient Networks, UBICOMM 2008

[2] Jürgen Dunkel, Alberto Fernández, Rubén Ortiz and Sascha Ossowski, Event-Driven Architecture for Decision Support in Traffic Management Systems, ITSC 2008

[3] Zakir Laliwala, Sanjay Chaudhary, Event-driven Service-Oriented Architecture, ICSSSM 2008

[4] A. Kumar Harikumar, R. Lee, C. Chiang, H. Yang, An Event Driven Architecture for Application Integratino using Web Services, IEEE IRI 2005

[5] ITU-T, "Recommendation M.3100 TMN-Generic Network Information Model," 1996.

[6] ITU-T, "Telecommunication Standardization Sector http://www.itu.int/ITU-T/

[7] Common Information Model (CIM): http://www.dmtf.org/standards/cim

[8] Distributed Management Task Force (DMTF): http://www.dmtf.org/

[9] Management Object Format (MOF): http://www.dmtf.org/education/mof/

[10] Web-Based Enterprise Management (WBEM): http://www.dmtf.org/standards/wbem/

[11] W3C, "Extensible Markup Language (XML) 1.0"; http://www.w3.org/TR/2006/REC-xml-20060816.

[12] Directory Enable Network (DEN): http://www.dmtf.org/standards/wbem/den

[13]Shared Information Data/Modell (SID): http://www.tmforum.org/browse.aspx?catID=2008

[14] New Generation Operations Systems and Software, NGOSS:http://www.tmforum.org/BestPracticesStandards/SolutionFrameworks/1468/Home.html

[15] Unified Modeling Language (UML) http://www.uml.org/

[16] Simple Network Management Protocol (SNMP): http://www.ietf.org/rfc/rfc1157.txt

[17] Atherton M. "Deploying CMDB Technology Pragmatism and realism will deliver the benefits, Freeform Dynamics Ltd.

[18] Information Technology Infrastructure Library (ITIL) www.itil-officialsite.com

[19] Data Mining: K. Cios, W. Pedrycz, R. Swiniarski, L. Kurgan, Data Mining: A Knowledge Discovery Approach, Springer, ISBN: 978-0-387-33333-5, 2007.

[20] Z. Benahmed Daho, N. Simoni, Towards Dynamic Virtual Private Service Networks: Design and Self-Management, in IEEE/IFIP NOMS 2006

[21] H. Huynh, E. Lavinal, N. Simoni, A Dynamic QoS Management for Next Generation of Services, ICAS 2007

[22] Noëmie Simoni, Chunyang Yin, Ghislain Du Chéné, Service continuity management through an E2E dynamic session in NGN, NOMS 2008

[23] Noëmie Simoni, Chunyang Yin, Ghislain Du Chéné, "An Intelligent user centric middleware in NGN: Infosphere and AmbientGrid" COMSWARE 08

[24] Noëmie Simoni, Simon Znaty, Gestion de Réseau et de service: Similitude des concepts, Spécificité des solutions. ISBN:2225829802

Vitae

Netzahualcoyotl Ornelas was born in Guadalajara (Mexico) on 1980. He graduated as Engineer in Computer Science from University Guadalajara Lamar in 2004 (Mexico). He received the MSc. in Networks from the Pierre et Marie Curie University (Paris 6) in 2006 (France). At present, he is preparing his PhD. in the Department of Networks at the University Paris 13 since 2007 (France). He is founder member of the IPv6 Task Force Mexico. He participated in research projects with University of Guadalajara, RENATER, SFR and Telecom ParisTech about IPv6, Multicast, ENUM, QoS, and NGN architectures. He collaborates in the UBIS project with University Paris 13, Telecom ParisTech and SFR. His current research area of interest is the self-management of networks and services in a NGN context.

Professor Noëmie Simoni is a chairman of Architecture and Engineering of Networks and Services (AIRS) research group at the Department of Computer Science and Network, Telecom ParisTech. Her research interests include QoS management and the end to end modeling of the entire telecommunication system. Through academic projects and industrial contracts, her expertise covers wide range of management topics. Her main works is focused on information modeling, management process re-engineering, workflow integration and convergence of control and management planes. She published over 70 technical papers and is co-author of two books, one on Network and Service management and another one on NGS (New Generation of Services). She is chair and program chair of the French-speaking conference on Networks and Services Management (GRES) which is held every two years and sponsored by Telecommunication Operators.

Doctor Chunyang YIN was born in China, in 1980. She is graduated from Tongji University, China, in Computer Science in 2002. She received the MSc. Degree in Computer Science from ENST, Paris, France in 2005 and the MSc. Degree in Network Security from Tongji University, China, in 2006. She received her PhD. degree in Computer Science and Network from TELECOM ParisTech in 2008. She's currently working as Post Doc. in the Department of Computer Science and Networks, TELECOM ParisTech. As a member of AIRS research group since September, 2004, she has participated projects cooperated with SFR as IA/PO, VTC2E and Seamless Userware. Her main research interests lie in the area of the service continuity in an NGN context.

Antoine Boutignon was born in Lille (France) on 1952. He graduated from ESME Sudria in 1977 (Engineer in Mechanics and Electronics). From 1977 to 1982, he worked at Steria to develop a real time system for EADS. In 1982, he joined the CS group to participate in the development of Datagram / X25 software and systems for telecommunication equipment. From 1990 to 1997, he worked as a study engineer in the Thales Communication R&D and Architecture division in the field of ATM Network equipment and services. He joined SFR in 1997 to participate in architecture studies on Data Networks. He is attached to the SFR R&D and Standards department as manager in the area of IP transport technologies and Mobile Core Network technologies. He has been involved in several research programs with Institut Telecom, Paris 13 and INRIA about IPv6, QoS, IMS, Service and Architecture. He is involved in several collaborative projects European as well as French. He published about 15 papers.