CONTEXT MANAGEMENT IN VIRTUAL HOME ENVIRONMENT SYSTEMS

Stavros Xynogalas, Ioanna Roussaki, Maria Chantzara and Miltiades Anagnostou

Department of Electrical and Computer Engineering, Computer Science Division,

National Technical University of Athens (NTUA)

9 Heroon Polytechneiou Str, 157-73 Zographou, Athens, GREECE

Telephone: (+301) 772 2422, Fax: (+301) 772 2434

staxy@telecom.ntua.gr

The emerging 3rd generation services are provided through various types of networks and terminals. Roaming users rely on new kinds of wireless networks and terminals to offer mobility and make services available everywhere. The users' requirement for continuous access to personalized services from any place, transparently and independently of the underlying network technology and the terminal or point of access, was satisfied by the first VHE prototypes. The challenge in service engineering is now to make these services smarter by adding context management capabilities to them. In this paper we propose an enhanced framework for efficient context management in a VHE environment.

1 Introduction

The emerging 3rd generation networks enable the deployment of very attractive multimedia services that take advantage of the high rates of data transfer. These services will be provided through various types of networks and many different kinds of terminals. Wireless networks and terminals are especially important, as they offer mobility and make services available to roaming users. As expected, one of the first end-users' requirements was to access subscribed and personalized services continuously from any place, transparently and independently of the underlying network technology and the terminal or point of access. This requirement has been addressed by the Virtual Home Environment (VHE) concept, introduced in the late nineties. Several standardization bodies, such as 3GPP [1], UMTS Forum [2], GSM MoU [3], ETSI [4], ITU-T [5], and TTC [6] are working on the specification of the VHE, while several research projects ([7] – [14]) have also elaborated on the validation of the VHE concept. IST project VESPER [15], funded by the European Community, is one of them and our work in this project has acted as a motivation for this article.

Ubiquitous service provision is an especially useful and attractive feature and now that the feasibility of the concept has been testified, service engineers seek ways to combine it with the latest features, the most significant of which is by far context-awareness. Implementations of the Virtual Home Environment are context-sensitive systems, which acquire information about the current network and terminal, check with user profiles and act accordingly, performing all necessary adaptations automatically. The profile mechanism includes interface-specific and service-specific user preferences, which could be perceived as context. Profiles could be edited manually, or created automatically, based on statistical observation of the user's behaviour, for example by remembering the most recent choice of background color or font size. VHE functionality could be used as a starting point for creating a system with full context-awareness.

We examined whether and how it would be possible to handle context efficiently within a VHE system. In this paper we present our analysis of context information and the architectural changes we consider to be necessary, in order to incorporate uniform context manipulation in VHE systems. A VHE-enabling architecture and especially the service personalization mechanisms are described in Section 2. In Section 3 we analyze context data and metadata, providing real service examples and an illustrative usage scenario. Section 4 refers to the necessary architectural changes, which will lead to the creation of an appropriate framework for efficient manipulation of context within the VHE. Finally, in Section 5, conclusions are drawn and future plans are exposed, while our current research work is placed into perspective.

2 VHE architecture and personalization

Within the VHE, users are consistently presented with the same personalized services wherever they go, regardless of the serving network and the terminal. The services capabilities are of course limited by the capabilities of each network and terminal, while they also depend on the users' subscription(s). VESPER defined and demonstrated a VHE architecture providing roaming users with service portability and session mobility across a multi-provider, heterogeneous network and system infrastructure. Based on scalable state-of-the-art technologies, an architectural framework was defined and specified, enabling the realization of a VHE in service provision and use. Existing definitions of VHE and its requirements were analyzed and new elements were added wherever found necessary. The main capabilities, such as portability of personal service environment across networks and terminals, service-to-terminal adaptation, service continuity and service-to-network adaptation, were studied and prescribed within a wide range of transport technologies. During the lifetime of the project a preliminary prototype was implemented, which demonstrated a subset of the VHE architectural constituents and provided a "proof of concept". This prototype was used to validate the proposed VHE architecture, both theoretically and practically, through experimentation.

Finally, migration paths were analyzed, in order to find the best ways for current systems to interwork with, or evolve to, advanced systems meeting the VHE requirements. The VHE-enabling architecture and especially the service personalization mechanisms are described in this Section.

2.1 A VHE enabling architecture

Within the VESPER project, a VHE enabling architecture was designed and tested, taking into consideration the advantages and disadvantages of former relative work and efforts. A high level view of this VHE architecture is presented in *Figure 1*.

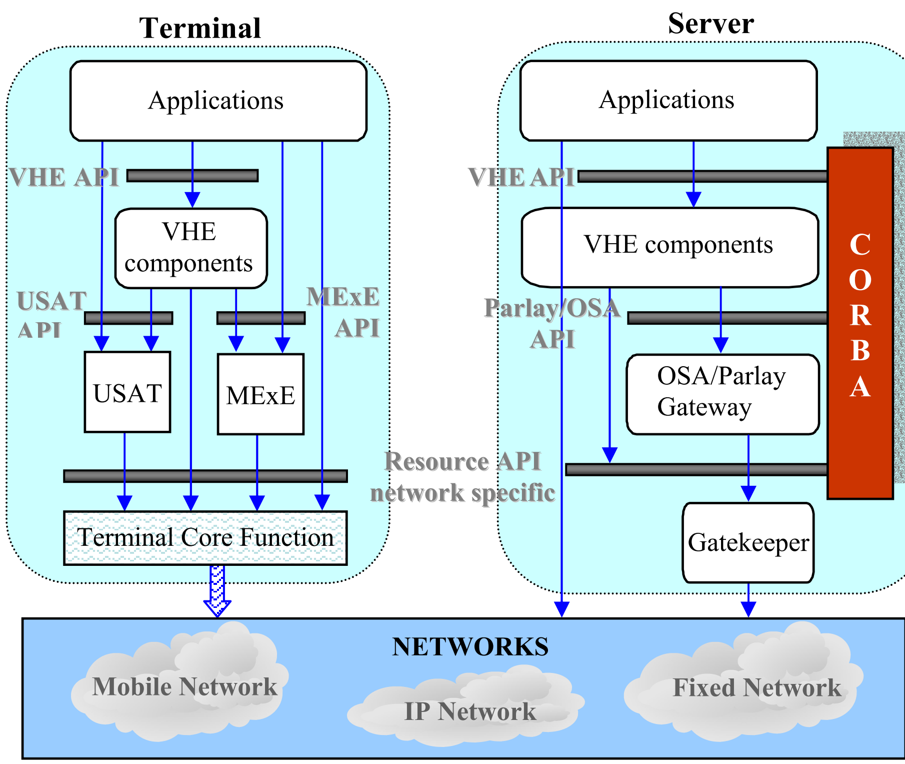

Figure 1. *VHE Architecture*

The connectivity path between the server and the terminal is depicted, as well as the different access networks. The most critical part of it is a set of components realizing the VHE concept, accessible by applications via the VHE API. The VHE components are located both at the terminal and server sides and they enable the provision of VHE featured services. On the server side the VHE components take advantage of the underlying network functionality via the standardized open OSA/Parlay APIs [16] that allow service provisioning independently of the underlying network technology (i.e. UMTS, IP, PSTN, ISDN, GSM, etc). On the user's terminal the VHE components may make use of the core terminal functionality via the UMTS SIM Application Toolkit (USAT) [17] or the MExE APIs [18]. The described architecture addresses different types of networks: IP Networks (wired or wireless), Mobile Networks (GSM, GPRS, EDGE, UMTS,…) and Fixed Networks (PSTN, ISDN, xDSL,…). The internal software structure of the VHE server is illustrated in *Figure 2*. The server API categories are depicted, as well as the VHE components that materialize the VHE concept.

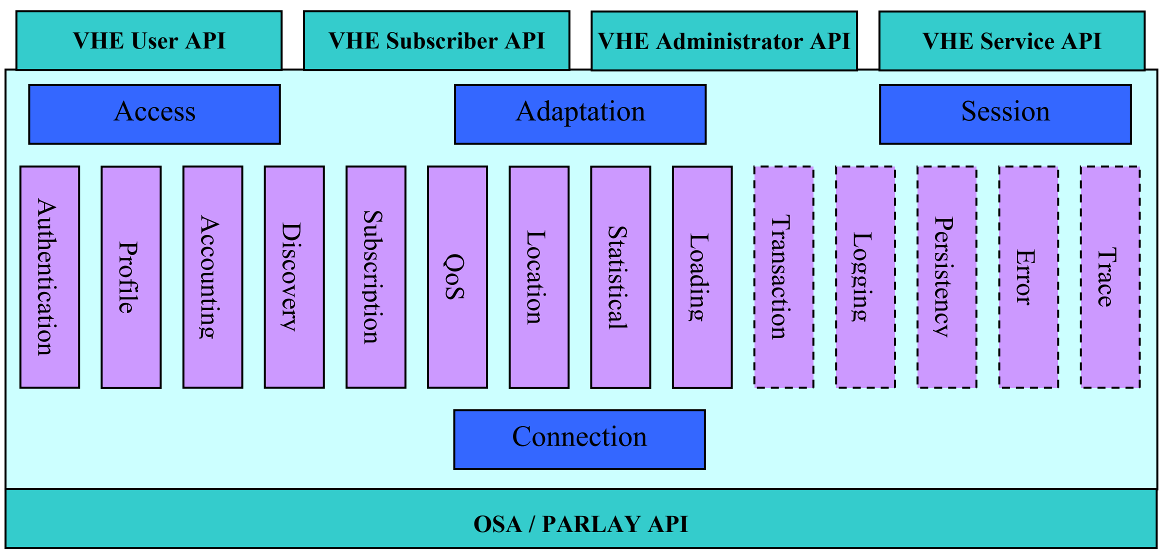

Figure 2. *VHE Server Structure*

The VHE components that are briefly described below are accessible from users, subscribers, administrators and the services through the VESPER VHE APIs. The *Access Component* provides the necessary functionality for a secure VHE session creation, within which the end-user can make use of a personalized service. The *Adaptation*

Component is responsible for service adaptation to network, terminal and user interface preferences, so that services are designed independently from network and terminal type. This component manages the input/output interactions with the end-user. The *Session Component* provides the functionality that enables an end-user to start, suspend/resume and join a service session. This component is responsible for connecting a VASP application to an end-user. The *Connection Component* handles the establishment and management of network connections via the functionality of Parlay APIs. The *Authentication/Security Component* provides functionality for the authentication of end-users and providers, while also offering some security features. The *Profile Component* provides functionality for maintaining and managing data pertinent to the end-user personalization and service usage preferences and constraints. The *Accounting Component* is responsible for the management of user accounts and charges the usage of the VHE services. The *Discovery Component* is responsible for providing a dynamic search –according to the user preferences and location– of services available on the VHE system. The *Subscription Component* provides functionality for management of user subscriptions to services. The *QoS Component* monitors the network performance and informs the user accordingly, enabling adaptation to QoS. The *Location Component* provides functionality to determine location of a user terminal within an active session via the OSA/Parlay interface. The *Statistical Component* constantly monitors the behaviour of the user by his/her service provision requests, while it is responsible of the extraction of patterns on service usage. The *Loading Component* provides mechanisms to transfer all data to the end system or any node, in a controlled way. The *Transaction, Logging, Persistency, Error* and *Trace Components* are not VHE specific and can be provided by an application server or an execution platform.

2.2 Profile management and service personalization in VHE systems

The existence of various profiles is not a new issue in service provision. In legacy systems, service profile management is standardized and is an integral part of the network technology used. In GSM, for instance, the service profile is maintained in the Home Location Register (HLR) and copied to the Visited Location Register (VLR) of the service area, where the end-user is roaming. It should be noted that, in these systems, the profile and logic of the services are associated with the point of attachment in the fixed networks and the terminal in the mobile networks.

One of the focal points in VHE systems is the dynamic profiling mechanism they rely on. In addition to the traditional service profiles, there is consideration for the personalized “look-and-feel” of the services from the end-user's perspective. This implies the management of a collection of properties on the user interface of the device in use, as well as particular preferences for the service user interface (man-machine-interface, user-perceived QoS). Furthermore, the VHE services –being both mobile and personalized– associate the service profile and logic with the end-user himself.

An efficient profile information model was designed and validated within the VESPER project [19]. The profile management constituents and the relations (inheritance, composition, or generic associations) between them, that altogether define the profile model, will be described here. The profile model is built on several entities, each maintaining different pieces of information: the User Record, the User Profile, the User Interface Profile, the User Services Profile and the Service Preferences.

The *User Record* (UR) is the “root” component of this model, related exclusively to one user of the VHE system. It maintains all user-specific information, such as the user credentials, which can be used to uniquely identify the user when he/she enters the system. It is used by the authentication and authorization mechanisms, providing the means to establish a trusted relation between the user and the VHE provider. Each user can have many *User Profiles* (UPs) (i.e. for private or business usage), which are made available for him/her to choose from upon his/her entrance to the VHE system. The UP is the unique combination of two information elements that manage specific preferences on the user's environment and state the preferences for the set of subscribed services. These two elements are the *User Interface Profile* (UIP) and the *User Services Profile* (USP). The contents of each profile are restricted to exactly one UIP and one USP.

The UIP includes all information that expresses the user's generic usage preferences and interface personalization features for any type of terminal. It defines a generic look & feel of the user environment, specifying the desired configuration of the terminal settings (i.e. display setting, audio parameters, menus, buttons, ring tones, etc), which is automatically adapted to the current terminal capabilities. The UIP is associated via "dependency relationship" with the *Terminal Profile* (TP) entity. The TP uniquely identifies a specific terminal, it reflects its capabilities and makes them accessible to the interested parties. A TP typically incorporates a set of hardware/software attributes, each of which includes a fixed range of values specified by the vendor. The set of attributes included within a TP is typically a superset of those defined within the UIP. All terminal specific information needs to be acquired and maintained, so that adaptation to the current terminal is provided. It should be noted that the TP attributes are automatically modified each time the user changes his/her terminal device, while the new values may be retrieved directly from the terminal (in cases of MExE or USAT enabled devices), a database locally maintained within the VHE system, or the terminal vendor (through an available API or by using a Mobile Agent platform).

The USP, on the other hand, comprises all data that delimit the service specific characteristics such as service logic customisation parameters, application user interface characteristics, security constraints and –where applicable– content-based personalization details. It is related to the Service Profile entity of the subscription model –that is selected by the subscriber– and is instantiated per service. Each USP can be associated with more than one *End-User Services Profiles* (EUSP), which are part of the subscription model and each of them corresponds to a subscribed service. The EUSPs represent the service templates defined by the subscriber of the VHE system, while they state the range of possible end-user specific settings for a customised service usage, often restricting certain user options. Alternatively, they may be directly offered by the service provider and constrained by the service contract imposed by the subscriber and through which the end-user or group of end-users is declared. The EUSP can be considered as a set of attributes which can be parameterised and which characterize the service personalization. The preferred service attribute values are defined through instances of the *Service Preferences* (SP) class. The set of attributes defined within an SP object is a subset of those defined in the EUSP corresponding to the same service. Basically, the Profile Component will not interpret the content of an SP object; SPs are seen as opaque data, received from the service and delivered back to it on demand without any interpretation. The Profile Component will only manage the SPs and maintain the relations to USPs (and UPs), providing management functions.

The VHE Provider can provide both the UIP and the USP, by default, while the user can update them at any time, subject to subscription constraints. The UP is the entity that enables the user to a personalized interface and service environment, and may be maintained irrespectively of the terminal and network types utilized for service usage.

The described profile model is depicted in *Figure 3*.

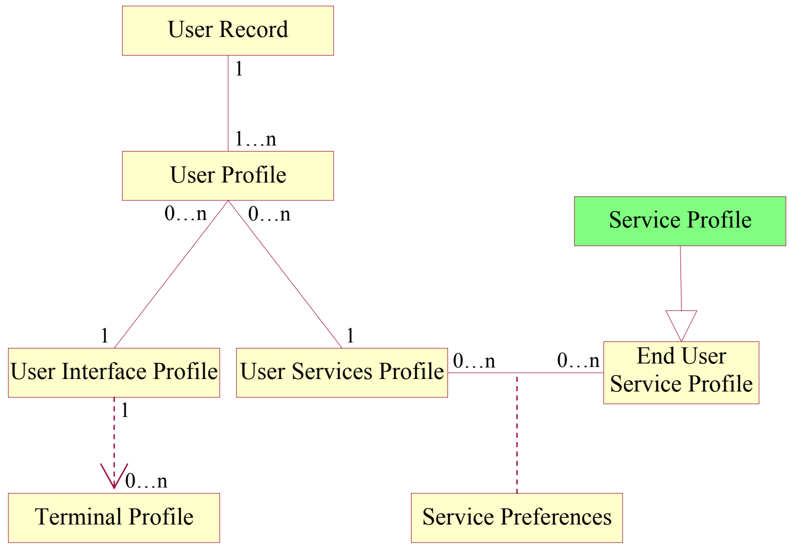

Figure 3. *VHE Profile Model*

This profile model mainly serves for satisfying the personalization requirements of the VHE systems. From our point of view, profile management is a critical aspect of context-awareness. More about the relationship between the aforementioned VHE profiles and the "context" concept can be found in the next section.

3 Context Awareness

There are many different types of context information. In subsection 3.1 we present a classification of context data and metadata, with explanatory examples. An illustrative future usage scenario is then presented in subsection 3.2, where all context types are interweaved in a complicated example of what our world may look like in a few years.

3.1 Context definition and modeling

In the field of application development, context is a term introduced at the beginning of the 90's, comprising information relative to user environment aspects, application or computing environments characteristics, and physical environment features. Most researchers comprehend the substance of context through the prism of their own area, and model it accordingly. In order to use context effectively and work towards context-aware systems, it is crucial to obtain a clear and deep understanding of this concept. A global context model should be developed, applying to all service categories and types. Thus there will be a uniform way to design context based applications, focusing on the determination of the context type to be used, and the selection of the mechanisms required to support context-aware computing. Up to now, several research groups have provided context models, which were either too general and vague ([20] – [26]), or too adapted and restricted to the fields of the researchers' interest, missing some important aspects of the overall issue ([27] – [31]).

In this section we aim to present a classification of the context types that addresses the overall needs for information acquisition and processing in service provisioning, spreading from simple static applications to the more sophisticated applications of the 3G/4G wireless networks. A first level classification can be made according to the nature of the context information. Thus the following context categories can be distinguished: (i) user "fingerprints", (ii) user preferences, (iii) user agenda, (iv) location, (v) time, (vi) user current state, (vii) terminal, (viii) network and (ix) any other data.

The *user "fingerprints"* context characterizes the user as an entity in his/her social, professional and personal life, and generally they only have to be provided to the system once. It includes *contact information* and *user identity* details (i.e ID number – given by the system, name, home address, email address, telephone, fax), user *professional context* (i.e. occupation, company – institution, position, boss, director, colleagues, office location / phone number, working hours), *social context* (i.e. family, relatives, friends, classmates, yearly income), *health* and *medical context* (i.e. disabilities, chronic diseases, allergies, medication, specific health precautions / dangers), *physical features* (i.e. height, weight, age, eyes, hair, colours, stamina), *psychological features* (calm, patient, short-tempered, persistent, egoist, friendly, loner), user *background - skills - capabilities* (i.e. education, languages spoken, car driving), user *personal preferences & interests* (i.e. task, habits, special occasions, hobbies, interests, news related preferences, science, research, business related preferences, entertainment, travel, art, sports, food preferences) and *user history context* (i.e. previous jobs, residences, visited places and locations). This type of context contains at least all of the elements included in the User Record of the client of the VHE system. Also, the user's habits and behaviour –included in the personal preferences context here– may be monitored by a statistical component as shown in the latest component models of the VHE server, in order to produce patterns of service usage and preferences, particularly important for efficient support for roaming users [32].

The *user preferences* context comprises temporary information about the user's current preferences and may be manually updated by him/her quite often. It consists of the *interface preferences* (i.e. layout, menu, buttons, font size, colours, element positions, background, images, display settings, audio settings, volume) and the *service preferences* (i.e. application specific preferences like accounting choices, presentation level details, multimedia parameters, and overall service preferences including which services can contact the user and when, or which contextual information can be made available to different types of services). The main body of information in this type of context is also included in the User Interface Profile and the User Service Profile described above.

The *user agenda* context refers to the calendar information of the user (i.e. activity type, appointments date & time, duration, meeting location & participants, hotel &

transport arrangements, non professional activity or date reminders). This type of context was not included in the VHE kernel, but was assumed to be organized and maintained by a third party service provider, requiring user subscription.

The *location context* addresses all information relevant to the current user location. This dynamic information may change constantly and can be acquired/detected automatically by the system's interface and recognition/identification hardware. This type of context includes *location information* (i.e. geographical position, country, domain, city, terrain, outdoors/indoors, building, floor, room, car, ship, airplane, train, traffic conditions, neighborhood), *"civilization" information* (location based content i.e. historical information, police stations, embassies, banks, transportation, site seeing, hotels, restaurants, entertainment, hospitals, pharmacies, doctors, gas stations, vehicle renters), *physical environment information* (i.e. weather, temperature, humidity, wind, lighting, noise level) and *user surroundings information* (identities of nearby people and objects, features of and changes to those objects, available infrastructure and resources). In the VHE systems, only the location information is used in cases of roaming.

The *time context* comprises all information related to the current time (i.e. time of day, working day or weekends, month, season of year, holidays). Originally, time was not taken into account by VHE systems. However, in the latest component models of the VHE server, time is considered as a parameter that affects the user's activities and habits, and is recorded, in order to produce patterns of service usage for efficient selection of the most appropriate replicated Home VHE provider or Visited VHE provider for users in their home domain and roaming users, respectively [32].

The *user current state* context includes all information relevant to the physical state of the user during the service usage. It consists of the *user biological state* (i.e. sick, fevered, tired, sleepy) and the *user emotional state* (i.e. calm, angry, bored, worried, interested, focus of attention). This type of context is not taken into consideration in current VHE system specifications.

The *application context* refers to the information relative to the applications provided. It contains *application specific information* (i.e. subscriptions, accounts, SLAs, preferences, necessary resources, operating system, distribution model, interworking requirements), *server specific information* (i.e. type, status, replicated instances). This data is generally not used in present VHE systems.

The *terminal context* includes all static and dynamic information related to the current terminal of the user, valid only within the ongoing access session. It refers to details on the terminal type, manufacturer, model and capabilities (i.e. screen size & resolution, bandwidth support, processing power, available software, control capabilities), as well as status parameters of the terminal (i.e. battery status, signal efficiency). The static information of this type of context is handled by the TP of the VHE system, and is used to adapt service provision to the capabilities of the terminal device [33], while the dynamic part of it is currently not taken into consideration.

Finally, the *network context* comprises all information related to the underlying network. It consists of network type and parameters (i.e. protocols, provider, connectivity bandwidth, delay, jitter, noise, costs, round trip time, available APIs, traffic – load, QoS level). This type of context is also monitored and used in all VHE systems by the Connection and the QoS Components as the dynamic adaptation to heterogeneous underlying networks is a focal point in this field [33].

The structure of the context model is depicted in Figure 4, including the main context types and the relationships between them.

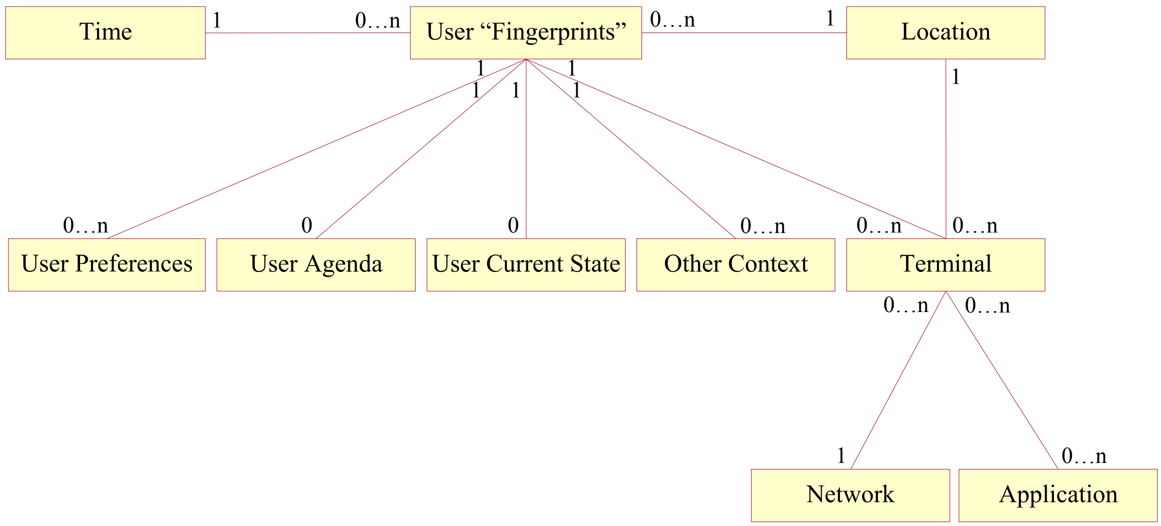

Figure 4. Context Model

The way that the information is provided to the system is very important, as well as the duration and updating frequency. Some pieces of information can only be *provided manually* –on-line forms or voice recognition techniques– and some may also be *acquired automatically* –via sensors, cameras, or microprocessor signal interpreters. In our classification we distinguish between context types that are *permanent* –unlikely to be changed– or *temporary* –expected to be changed or updated often–, *static* –invariant within an access session– or *dynamic* –change constantly.

	Context type	**Manually entered**	**Automatically acquired**	**Permanent**	**Temporary**	**Static**	**Dynamic**	**Service Example**
User "Fingerprints"	Contact information / User identity	•		•		•		Directory service
	Professional context	•		•		•		Emergency service
	Social context	•		•		•		Chat service
	Health / medical context	•		•		•		Medicine Discoverer
	Physical features	•		•		•		Dating service
	Psychological features	•		•		•		Partner selector
	Background / skills / capabilities	•		•		•		Employment Agency
	Personal preferences / interests	•	•	•		•		Travel Agency
	User History context	•	•		•		•	Tourist Guide

	Context type	Manually entered	Automatically acquired	Permanent	Temporary	Static	Dynamic	Service Example
User Preferences	Interface preferences	●			●	●		Any service
	Service preferences	●			●	●		Any service
User Agenda	Agenda / Calendar information	●	●		●		●	Meeting Scheduler
Location	Location information	●	●		●		●	Transportation Route Discovery service
	"Civilisation" information		●		●		●	Theft Recovery service
	Physical environment information		●		●		●	Excursion Planner
	User surroundings information	●	●		●		●	Professional Presentation Consultant
Time	Date & time information		●		●		●	Dinner Planner
User current state	User biological state	●	●		●		●	Health Monitoring
	User emotional state	●	●		●		●	Telephone Call Filtering
Application	application specific information	●	●	●	●	●	●	Any service
	server specific information	●	●	●	●	●	●	Any service
Terminal	Terminal parameters and variables	●	●		●	●	●	Any service
Network	Network parameters and variables	●	●		●	●	●	Any service

Table 1. *Context types classification and service examples*

Table 1 presents the classification of context according to the above criteria, while an example of a service that is likely to use each type of context is also provided.

3.2 Scenario describing context-aware services

The provision of context-aware services seems to be very attractive for consumers, since this embedded smartness on services promises to make life easier and bring us closer to what Donald Norman describes in his latest book, The Invisible Computer [34] as the "third generation of PC, the generation where the machines disappear from sight, where we can once again concentrate upon our activities and goal in life". On the other hand, on behalf of service providers and telecommunications network vendors, the implementation of context-aware services represents a challenging and also lucrative opportunity to attract users. In this perspective, much effort has been spent during the last years towards the specification of context-aware services, as well as their development and validation. Several research teams have worked towards this direction and implemented services with context-aware features ([35] – [41]). Based on the context structure presented in the previous subsection, a scenario that involves several context-aware services is presented here, aiming to clarify the usage of the different context types and demonstrate the attractiveness of a fully enabled context-aware system.

John Pappas is an executive of an international construction company, who works at the company department that resides in Athens. John is informed that he will have to attend a meeting in Paris for a project he is currently involved in, so he activates his electronic agenda entering the meeting date, place and scope to check his availability. Since he is not available, the meeting scheduler application contacts the agendas of all other meeting participants and comes up with a date and time that suits everybody. After the meeting is confirmed, John proceeds with his travel and hotel arrangements. The e-travel agent checks the flights program and reserves a ticket for John, taking into consideration John's personal preferences, like traveling in business class, in the morning, with a specific airline company. The e-travel agent books the most appropriate airplane ticket and charges the company account for travel expenses. Additionally, the airline's catering is automatically informed about John's eating preferences: John happens to be a vegetarian. Now John's flight is fully organized, and what remains to be arranged is the transportation means to take him to the airport. According to the departure time and the data regarding the expected traffic, a reminder is held, so that a taxi arrives at his house to take him to the airport. All John has to do is confirm the arrangements. After the flight reservations are settled, the e-travel agent proceeds with the accommodation arrangements. Taking into account the meeting place and duration, John's hotel preferences –4 star hotels and a non-smoker rooms– and the company's policy on maximum accommodation expenses for employees, the e-travel agent searches for the appropriate hotel and room and makes a reservation.

The flight day arrives and John is ready to travel. The electronic devices that John carries are his mobile phone and a PDA-like device that supports wireless connection. This device includes a location detector and a health sensor and interacts with various micro-signal-generators that are spread in the environment (on other people, furniture, buildings, etc). He receives a message on his mobile phone that a taxi is arriving in 5 minutes to transfer him to the airport, as scheduled. After reaching the airport, everything proceeds as planned and his flight departs on schedule. During the flight, the health sensor detects that John is suffering from stomachache and since this is a frequent health problem for John, it reminds him of his regular ulcer medicine in his cabin luggage. While John takes his pill, the airhostess is informed about John's health problem, through the passengers detail system. She is instructed to replace John's meal with a lighter one, in order to avoid possible aggravation of his situation. The airport traffic management system reschedules the flight's arrival time to half an hour earlier than expected. As a result, the hotel reception rearranges the hotel driver's program, so that John doesn't have to wait. As soon as the airplane lands on the Charles de Gaulle airport, the driver is informed about John's exit gate and heads to pick him up at the nearest possible spot. John is notified about the driver's location and with the taxi position is delivered to him.

Finally, John arrives at the hotel, goes up to his room, switches off his PDA and rests for a while. The meeting is rescheduled for the next morning, instead of this afternoon, so John receives a message on his mobile phone asking him to switch his PDA on. He proceeds accordingly and obtains a notification about the changes in the meeting plan. Since the rest of the day is free for him, his ulcer is not bothering him and the weather is detected to be very pleasant, he receives a proposal regarding a sightseeing walk in Paris and an electronic map, where interesting siteshe has never visited before are highlighted. Considering John's interests, it is recommended to him to visit the Eiffel Tower. John accepts the proposal and asks for a taxi to take him there. The taxi availability dynamic database is searched and a taxi is booked to pick him up in 20 minutes. John gets dressed, goes downstairs and enters the taxi waiting for him. At that moment, the charging system starts counting. The payment is deducted according to duration and distance. An electronic map for the route to follow, in order to avoid traffic jams and arrive on time to the destination comes up at the satellite trip-computer of the taxi. They reach the Eiffel Tower and John gets off the taxi, while his payment agent automatically charges his credit card. When John's PDA detects that he is next to the Eiffel Tower, it delivers to him relevant history details and information on its construction. As John approaches the entrance of the tower, the Automatic Debiting System detects his presence and finds that John's company's employees are entitled to a reduced fee, which is once again automatically charged on his credit card. The location sensor detects that John is using the elevator up the tower and he is advised not to go higher than the first level, as he suffers from hypsophobia. John reaches the first floor and enjoys the beautiful view. He receives a notification that an old friend of his is also present at the same floor of the Eiffel Tower and the two friends get to meet.

The above scenario, depicted in *Figure 5*, gives an example of the use of almost all of the context types in real life. Even though this scenario may seem to refer to a distant future, similar applications are to be commercialized within this decade.

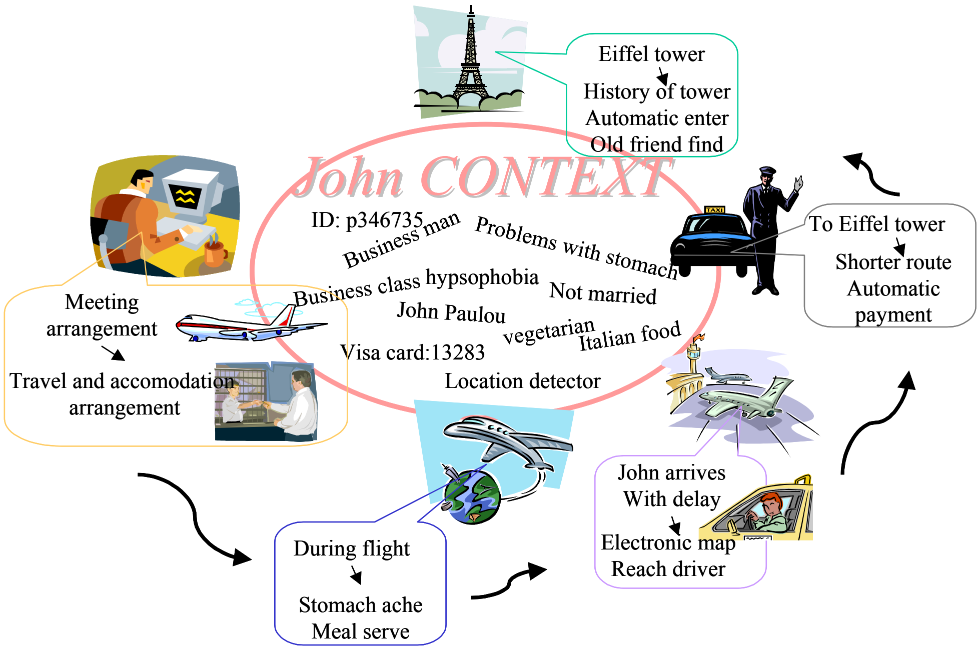

Figure 5. *A context-aware service scenario*

4 Proposed Enhancements in VHE Architecture

In Section 2 of this paper we presented an appropriate architecture for VHE provision. Then, in our effort to add uniform context manipulation capabilities to VHE systems, we went on and analyzed context information in Section 3, providing some interesting service usage scenarios for the various types of context. Based on this analysis, we will now propose ways to enhance VHE systems, so that they will be able to handle all types of context more efficiently.

As far as context management is concerned, the most useful features of VHE systems are the service personalization and profile management mechanisms, as we have already emphasized. All VHE components have something to do with context, but especially the Statistical, Profile, QoS and Location components' functionality is too closely related to context management to ignore. We propose that they become sub-components of a new "Context" super-component, responsible for the coordinated acquisition and management of all contextual information in a uniform manner. By concentrating all context manipulation functionality to the Context component we can avoid code repetition, as almost all components currently need to have similar methods for contextual information storage and retrieval. Furthermore, by including all these operations in a single interface, we enable the service developers to create lightweight context-aware applications, which only need to communicate with one component for anything related to context. This is very important for applications that deal with dynamic context, because if they have to waste a lot of time contacting two or three components for the information they need, they will not be able to keep up with the data input rate. It makes sense to create a separate component to take care of the context information requests and updates, as they are a significant percentage of the total number of method calls in VHE systems. Finally, in a real context-aware system, the Statistical, Profile, QoS and Location components will have to exchange a lot of information with each other, so we prefer to include these objects in the same package, in order to relieve the network and increase the quality of service. The distribution model of the VHE components needs to be carefully designed, to avoid creating multiple bottlenecks. As soon as the system specification is finalized, we will proceed to an extensive simulation procedure, the results of which will be used for further evaluation purposes.

We believe that the enriched and re-arranged set of VHE components shown in *Figure 4* is most appropriate for efficient context management in VHE systems. The Context sub-components briefly described in the next paragraph only cooperate with each other; the Context component provides the functionality of the old Statistical, Profile, QoS and Location components to the other VHE components and implements the context-oriented APIs to the users, subscribers, administrators and services.

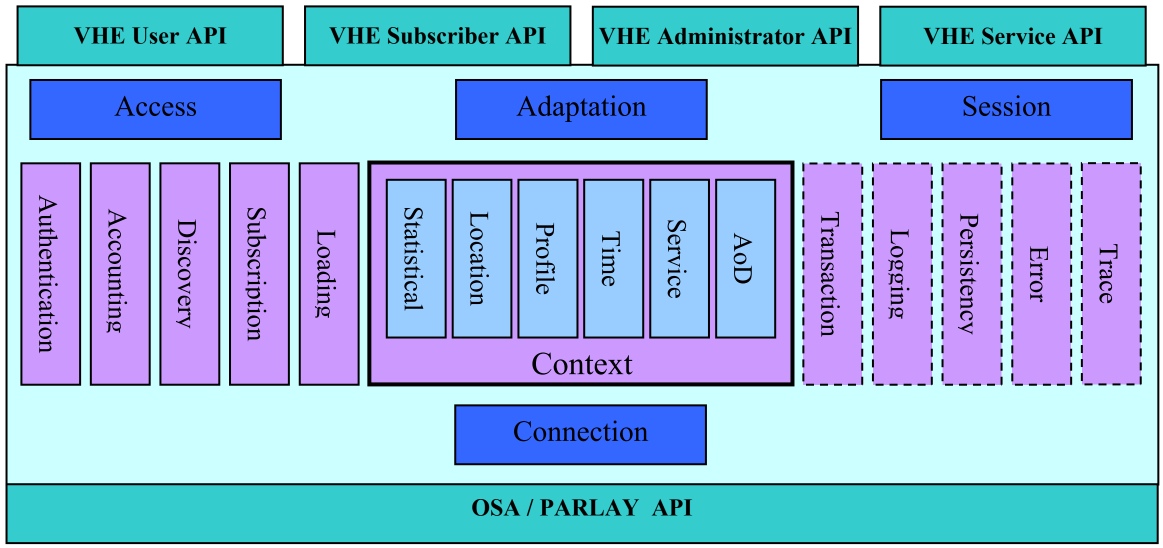

Figure 6. *Re-arranged Components for Context management*

As explained earlier, the *Statistical Sub-Component* constantly monitors the user's actions, records his/her preferences and decisions and is responsible for the extraction of patterns in his/her behaviour. These patterns can then be used to produce automated context input for the other sub-components.

The *Location Sub-Component* addresses all information relevant to the current user location. It acquires this information automatically through sensors or tracking applications and makes it available to the other components as location context. It also translates this information internally and constantly updates the user surroundings context, notifying the user whenever someone or something classified as "important" is near. Of course issues like what "important" means, or which notification method should be triggered each time, are entirely up to the user to define in his/her own profile. Any user could deactivate all such notifications, or choose, for example, to be notified with a short "beeping" sound whenever one of the people in the "friends" category is less than 5 meters away. With or without notifications, this sub-component is responsible for maintaining an image of the immediate environment of the user and making it available to the system and the services as user surroundings context.

The *Profile Sub-Component* is used for the maintenance and management of data pertinent to the user personalization and service usage preferences and constraints. This sub-component is the heart of the Context component, as it is responsible for the user fingerprints, preferences and agenda context information management.

The *Time Sub-Component* is responsible for keeping track of time and date information. Taking as input the user agenda information from the profile sub-component, it is responsible for announcing the current events to the user, according to his/her preferences. This sub-component is also able to use advanced algorithms to predict possible delays and problems and alert the user whenever he/she may be late for an event that has been given high priority.

The *Service Sub-Component* is responsible for monitoring all aspects of active services and managing context relevant to the applications, servers, networks, clients and terminals. The QoS Component currently monitors the network performance and informs the user accordingly, enabling adaptation to QoS. The role of this component should be transferred to the Context super-component, as part of the Service sub-component.

Finally, the *AoD Sub-Component* deals with all other context information. Any context mechanism not defined above can be taken care of within this sub-component in an ad hoc manner. AoD stands for "Any other Data" and we use this sub-component to accommodate the immense diversity of possible context information. We have tried to define a system that will be able to manage efficiently the most important types of context information. We considered all aspects of human activity and located the context information needed most often, but context is still an overwhelmingly rich concept.

Therefore, it is important to make our system flexible for future use, by providing a mechanism that will be able to incorporate any new specific type of context, according to the users' needs. In telecom-oriented communities, for example, AoD can include telecom-specific context management, or within a large company, AoD can take care of the company scope specific context.

The *Context Super-Component* offers a uniform way of managing context in service development and usage. This abstraction can help service developers when they create state-of-the-art context-aware services by offering elaborate models and context databases, significantly reducing the time they have to spend in context analysis. As far as the users are concerned, context becomes more and more complicated and they will soon need a tool to manage all context information. With the proposed mechanisms, they will only need to provide links and grant admissions to their own context info. Until now, context information was dispersed in many different components and used in many different ways.

The Context component aims to extend the Virtual Home Environment framework, in order to handle context in an organized, uniform manner. Technology has gone a long way and is now ready to deal with all context information, so the next step is for services to evolve accordingly. Location-aware services and cartography have presented remarkable progress the last years and Geographical Information Systems have provided very efficient databases and interesting applications. Extensive research effort has also been spent in Profiling and Calendar services. In fact, people working in many research areas, have already presented specific context models and infrastructure. The VHE is an appropriate framework for the integration of all these approaches and this was one of our main incentives for the work presented in this article.

5 Conclusions and Future Plans

The framework described in the previous section facilitates the provision of context-aware VHE services, based on the context model presented in 3.1. A substantial investment in context infrastructure is necessary for the realization of scenarios like the one described in 3.2. Large context databases are already being created and made available to the users through advanced systems and services. A very good example is the provision of location context (maps, Geographical Information Systems etc.) and location-aware services, which take advantage of the latest mobile terminal tracking techniques and the Global Positioning System. As soon as we are able to manage all types of context information efficiently, all human activities will be significantly enhanced and accelerated. However, this technology does not come without cost or danger, for example privacy is especially threatened by automated context information acquisition mechanisms. A lot of questions rise and a lot of work lies ahead.

As an academic research team, we have already set our targets for the near future and joined relevant research projects. Within these projects we intend to elaborate further on context definition and develop state-of-the-art mechanisms and visual tools for context management in service engineering. One of these projects is CONTEXT [42], a new IST project, for which we are already designing a context-enabled service creation platform. The context types are to be associated to appropriate management mechanisms and the service developer will be able to automatically create high-level context-aware services, using context-aware objects just like common software components. Of course, as far as the terminals and underlying networks are concerned, our system will maintain the VHE features, or even enhance them.

6 References

1. 3GPP TS 22.121 v5.3.0: "The Virtual Home Environment "(Release 5), March 2002.
2. UMTS Forum Report 1: “A Regulatory Framework for UMTS”, June 1997.
3. GSM MoU: "3G Service Requirements and Concepts", PRG TG.21, V3.1.0, 1998.
4. ETSI: UMTS 30.01, UMTS baseline document, positions on UMTS agreed by SMG including SMG#27 UMTS 30.01 version 3.5.
5. ITU-T: “Draft New Recommendation Q.1711, Network Functional Model for IMT-2000 (previously known as Q.FNA)”, Version 12.2, May 1998.
6. TTC: TD-3GA-23.127 (R99), “VHE/OSA”, March 2000
7. N. Houssos et al., “A VHE architecture for advanced value-added service provision in 3rd generation mobile communication networks”, Proc. Globecom 2000 Workshop on Service Portability and Customer Premises Environments, San Francisco, 2000.
8. L. Hagen and M. Breugst, “Impacts of Mobile Agent Technology on Mobile Communication System Evolution”, IEEE Personal Comm. Magazine, August 1998.
9. P. Bellavista, A. Corradi and C. Stefanelli, “A Mobile Agent Infrastructure for the Mobility Support”, Proceedings of ACM SAC 2000, Como, Italy, March 2000.
10. P. Farjami, C. Gorg and F. Bell, “A Mobile Agent-based Approach for the UMTS/VHE Concept”, Proceedings of Smartnet'99, Bangkok, Thailand, 1999.
11. S. Lloyd, R. Hadingham and A. Pearmain, “Virtual Home Environments to be Negotiated by a Multi-Agent System”, Chapter 9 of “Agent Technology for Communication Infrastructures”, Wiley 2000, pp. 111--121.
12. Eurescom Project P920-GI, Deliverable 2: “Investigation of architectures and protocols for UMTS network evolution”, October 2000.

13. F. Daoud and S. Mohan, “Strategies for Provisioning and Operating VHE Services in Multi-Access Networks”, IEEE Communications Magazine, January 2002, pp. 78-88.

14. M. Torabi, “A Shift in the Mobile Network Service Provisioning Paradigm”, Bell Labs Technical Journal, Vol. 5, No. 3, July–September 2000, pp. 112--129.

15. VESPER (IST-1999-10825): http://vesper.intranet.gr.

16. A. Moerdijk, L. Klostermann, “Opening the Networks with Parlay/OSA APIs: Standards and aspects behind the APIs”, submitted to IEEE Comm. Magazine.

17. 3G TS 22.038 v5.2.0: “Technical Specification Group Services and System Aspects; USIM/SIM Application Toolkit (USAT/SAT); Service description” (Stage 1), (Release 5), June 2001.

18. 3GPP TS 23.057 v6.0.0: “Technical Specification Group Terminals; Mobile Station Application Execution Environment (MExE); Functional description”, (Stage 2), (Release 6), June 2002.

19. S. Caokim and S. Sedillot, “Profiles Management in Next Generation Personalised Services”, Proc. 2nd European Conference on Universal Multiservice Networks, ECUMN'02, Colmar, France, April 2002.

20. A. Dey and G. Abowd, “Towards a Better Understanding of Context and Context-Awareness”, Proc. Workshop on the what, who, where, when and how of context-awareness at CHI 2000, April 2000.

21. D. Petrelli, E. Not, M. Zancanaro, C. Strapparava and O. Stock, “Modelling and Adapting to Context”, Springer-Verlag London Ltd, Personal and Ubiquitous Computing Magazine, Vol. 5, Issue 1, 2001, pp. 20--24.

22. A. Park, S. Lipperts and M. Wilhelm, “Location Based Services for Context Awareness - Moving from GSM to UMTS”, Proc. SSGRR 2002w, L'Aquira, 2002.

23. T. Rodden, K. Chervest, N. Davies and A. Dix, "Exploiting Context in HCI Design for Mobile Systems", Proc. First Workshop on Human Computer Interaction with Mobile Devices, Glasgow, U.K., May 1998, pp.12--17.

24. A. Ward, A. Jones and A. Hopper, "A New Location Technique for the Active Office", IEEE Personal Communications Magazine, Vol. 4, No. 5, 1997, pp. 42--47.

25. J. Pascoe, "Adding Generic Contextual Capabilities to Wearable Computers", Proc. Second International Symposium on Wearable Computers, Pittsburgh, Pennsylvania, USA, October 1998, pp. 92--99.

26. B. Schilit, N. Adams and R. Want, "Context-Aware Computing Applications", Proc. IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, USA, December 1994, pp. 85--90.

27. R. Klemke, "Context Framework - an Open Approach to Enhance Organisational Memory Systems with Context Modelling Techniques", Proc. PAKM2000, Basel, Switzerland, October 2000.

28. K. Henricksen, J. Indulska and A. Rakotonirainy, "Infrastructure for Pervasive Computing: Challenges", Proc. Informatik 2001: Workshop on Pervasive Computing, Vienna, Austria, September 2001, pp. 214--222.

29. D. Chalmers, D. Michaelides, L. Carr, N. Dulay, W. Hall, D. D. Roure and M. Sloman, "Contextual Information and Physical Environments", to be presented in MDM 2003, Melbourne, January, 2003.

30. J. Benedito dos Santos Junior, R. G. Edson dos Santos Moreira, G. Blengini Faria, "The modeling of structured context-aware interactive environments", Journal of Integrated Design and Process Science, Vol. 5, No. 4, December 2001, pp. 77--93.

31. P.J. Brown, J.D. Bovey and X. Chen, “Context-Aware Applications: From the Laboratory to the Marketplace”, IEEE Personal Communications, Vol.4, No. 5, 1997.
32. I. Roussaki, M. Chantzara, S. Xynogalas and M. Anagnostou, “The Virtual Home Environment roaming perspective”, accepted for the ICC 2003.
33. I. Roussaki, H. Jormakka, S. Xynogalas, A. Laikari, M. Chantzara and M. Anagnostou, “Multi-terminal and Multi-network Access to Virtual Home Environment”, IST MWT Summit 2002, Thessaloniki, Greece, June 2002.
34. D.A. Norman, “The Invisible Computer”, MIT Press, 1998.
35. A.K. Dey, G.D. Abowd and A. Wood, “CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services”, Proc. International Conference on Intelligent User Interfaces (IUI '98), San Francisco, California, USA, January 1998, pp. 47--54.
36. K. Cheverst, K. Mitchell and N. Davies, “The Role of Adaptive Hypermedia in a Context-Aware Tourist Guide”, Communications of the ACM, Vol. 45, No. 5, 2002.
37. D. Mandato, E. Kovacs, F. Hohl and H. Amir-Alikhani, “CAMP: A context-aware mobile portal”, IEEE Communications Magazine, Vol. 40, No. 1, January 2002.
38. N. Ryan, J. Pascoe and D. Morse, "Enhanced Reality Fieldwork: the Context-aware Archaeological Assistant”, in V. Gaffney, M. van Leusen and S. Exxon (eds.) Computer Applications in Archaeology, 1997.
39. B. Schilit and M. Theimer, “Disseminating Active Map Information to Mobile Hosts”, IEEE Network Magazine, Vol. 8, No. 5, 1994, pp. 22--32.
40. M. Karagiozidis et al., “Location Aware Visually Enhanced Ubiquitous Services”, IST Mobile & Wireless Telecommunications Summit 2002, Thessaloniki, June 2002.
41. IST Advisory Group, “Scenarios for Ambient Intelligence in 2010-full”, 2001.
42. CONTEXT IST-project (IST-2001-38142), http://www.cordis.lu/ist/projects.htm.