Ubiquitous Computing Infrastructure with a Container Active Map and a Naming System serving the Home Care environments

John C. Sarivougioukas, Manolis Tsiknakis, Aristides Th. Vagelatos, Dimitrios I. Fotiadis

Abstract—**Home Care practice, provided to the elderly and the chronic diseases' patients, expects to acquire an intelligent upgrade to both the nature and the quality of offered services by the employment of ubiquitous computing infrastructures. Two important building blocks are introduced, the Container Active Map and the Naming System, complementally to the already proposed, suggested, and announce in the relative literature, architectural design options for the implementation of ubiquitous computing environments. The parameters of location and identification were among the first that examined by the scientific community in the development of context-aware software applications. Ubiquitous computing applications in the demanding Home Care environment have to address the peculiarities related to the participating entities' position and presence accordingly for the development of context-awareness among the coexisting software applications. Approaching from the systems' point of view, the internal structure of a typical ubiquitous computing application is described including the additional modules covering specific requirements and the operational characteristics are commented.**

I. Introduction

HOME Care Systems for the elderly and the disabled today is a necessity due to medical and financial reasons. Clinical costs and hospitals' internal infections make this necessity stronger and drive the potential for Hospitalization at home. The elderly and the patients with chronic diseases, can't stay at, require attention, treatment and support from specialized nursing personnel and monitoring by intelligent medical systems. The coexistence and collaboration of heterogeneous systems, for sensing and processing information, scaled down and transparently in-woven into the environment, with a common aim to interact with the ubiquitous computing users in a natural way raises many social, economic, and technical issues. Home Care systems must either detect the development of a crisis, or perceive the health status, or rectify the elderly or the patients' behaviors and habits [26]. The required ubiquitous computing infrastructure must supervise a large set of objects, persons, entities, or even services; at home it has to take the form of a context-aware system, including entities' identification and location, in order to make the appropriate decisions at the proper time.

Sharing of validated information is among the primary concerns of the collaborating software applications attempting to create joint context. The suggested architecture introduces a single common reference for all coexisting applications to satisfy the appearing needs of location and identification, extended to include the participating entities' attributes, properties, and functioning. For example, two or more software applications having incompatible accuracy referring to a point in a room's space, they will resolve the matter only if they are designed specifically to handle such a co operational problem related to accuracy, which is not normally the case. A common reference service must exist and be contacted to reconcile the differences creating context-awareness. Also, identification of the entities in the ubiquitous computing environment has to present the same characteristics to all participating software applications. Suppose that a thermometer measures the patient's temperature supplying data to two software applications using different temperature's scales, the one of them is concerned with the absolute reading of the device, while another application is occupied with the temperature's rate of change. A coexisting software service is required to identify the particular thermometer, among the possible others, providing besides the identification information, additional functional attributes assisting to resolve ambiguity. Coexisting, independently running and collaborating software applications striving to develop common context must use the same set of information, including location and identification, regarding the nature and the format of the handled data. Therefore, the ubiquitous computing applications' information has either to be normalized in a commonly acceptable format to set it comprehensible to all running applications or to appropriately program all vendors', existing or future, applications to interoperate, which is impossible without some sort of widely acceptable standardization.

According to a conducted survey [1] numerous context-aware projects have been developed presenting systems with an underlying infrastructure considering location as part of the developed context and employing maps as a visual aid. When a map application discovers and displays the controlled entities is considered as an active map and it is claimed that a container active map (CAM) is the map that besides the display projection and the entity's discovery is aware about the entity's characteristics too. Specialized

Manuscript received June 30, 2006.

John C. Sarivougioukas, is the Head of the MIS Department of the State Hospital of Athens 'G. Gennimatas', Athens, Greece. (phone: +302107783040, e-mail: jcsari@ath.forthnet.gr).

Manolis Tsiknakis, is a Researcher at ICS-FORTH, Heraklion, Greece (e-mail: mtsiknakis@csi.forth.gr).

Aristides Th. Vagelatos, is a researcher at RACTI, Athens, Greece (e-mail: vagelat@cti.gr).

Dimitrios I. Fotiadis, is a Professor at the Computer Science Department, Unit of Medical Technology and Intelligent Information Systems, University of Ioannina, Greece (e-mail: fotiadis@cs.uoi.gr).

technologies have been applied [2] estimating the differential position's distance values of an entity from the installed wireless beacons and then displaying the location of the controlled entity on active maps such as Wayfinder, Intentional Naming Systems or Viewfinder which fail to interoperate due to the lack of standard common references. The underlying infrastructure must present certain characteristics as it is the result of the comparisons performed in [3] examining in parallel Schilit's and Pascoe's layered context-aware supporting infrastructures.

In the home care environment, internet allows the interconnection of sites and it satisfies asynchronous monitoring needs relying on reliable locally supporting infrastructure. In [7], five layer infrastructure is suggested while in [8] a set of proper layer interfaces presented and in [9] another design approach is presented with an infrastructure in the form of a stack data structure [9]. The internal operations of a context-aware application require specific syntax, developed semantics, related logic [9, 10] and mechanisms for adapting the system's behavior [10]. Also, the developed perception within a system may be expressed by entities' roles and relations [11] requiring managerial operations [12] interpreting the common perception of the connected heterogeneous systems [13]. In addition, issues are raised with respect to privacy and security in context-aware systems in and out doors [15].

For identification and tracking purposes, names and synonyms are assigned to entities along with a set of functions and mechanisms [16] that return entities' names as query results addressed to names' storing databases. Appointing names based on entities' characteristics may be extended to relate entities' names with the storage of software components [17]. Another approach is based on syntax that leads to hierarchical structure of names [18] including even the describing entities' interfaces. By another method, the entities names are handled as an abstraction of tags [19] following strict syntactical rules to assist the applied queries to relational databases. A Naming System, part of the internal infrastructure, resolves applied queries [20] finding the stored and interrelated names and the existing services sharing such information with the co-existing driving applications. In [21] a four layered infrastructure is examined and the developed semantics are close to our design missing standard and common references among the co-existing applications and thus, reducing the interoperability opportunities. In [22], another consideration for the infrastructure's semantics uses a tag's abstraction to follow the participating entities' names which are handled by the maps and the events that take place. Also, the ubiquitous computing applications apply recursive relationships on the available maps [24, 25], thus forming a hierarchical but not universal structure of maps. In this paper it is suggested a similar hierarchical but universal structure of maps which is standardized, syntactically, with a uniform system of coordinates and a commonly accepted interfacing among the running applications, to support the semantics requirements.

II. CONTAINER ACTIVE MAP (CAM)

A map depicting a home care environment is considered as a set consisted of the elements included in the employed foreground, of dynamic nature, and exposed over a static background with the participating entities contributing into the developed context. Applying in or out zooming operation on a map, the applied scale on the display either hides or shows additional entities determining the level of detail which is directly proportional to both the background and foreground. Moreover, the appearance of each element, including its shape, size, and other appearance properties, depends on the applied scaling factor while each element is followed by its coordinates and the identifying attributes and properties.

The accuracy of the position of the items exposed on a map depends on the demands of the formed or intended context. Certain maps' applications require extreme accuracy while for others it is sufficient to include loosely related coordinates for the position of the described items on the background and foreground.

A series of computations is required to issue an adequate presentation of a map using data from the participating items in the foreground and background, the map's scaling factor, and the display area where the map is going to be exposed. Outlining the required computational operations, first, the map's presenting area is calculated. Second, a filtering algorithm is applied in order to decide the subset of the items of the background falling into the area specified in the first step. Third, a modulating algorithm is applied in order to decide which of the foreground items will be included on the map taking into consideration the scaling factor which in turn determines the level of detail. Forth, the attributes and the properties of the chosen items contribute to developing their adequate presentational form connecting the above computational pieces together forming the desired map. The superset containing all background items and all foreground elements constitute the universal map from which each of the requested maps is drawn applying scaling operations.

The maps' content changes with elements entering or leaving the projection space, and periodic update is required to present the dynamic nature of the foreground entities processing only the entities' data with changed state and status. A map may be depicted either using a photograph or a graphic item adapted with one of the known file formats, satisfying compatibility issues for storage, transmission, presentation, and interoperability. Given a picture of a map the zooming operations are limited up to the point of the image's acceptable resolution. On the other hand, if it is required to construct a map in a graphical format based on the available information of the participating entities, two advantages are observed, related to: the capability to construct non-existing views of maps based on the reliably collected information and there is a continuous space of scaling factors to choose from obtaining the desired map's size.

Maps represented as software objects provide the capability to facilitate in a single software unit the map's data, foreground and background entities. There is a recursive relationship, a member function, operating between maps allowing each map to result from its previous state. Hence, given information about a map, successive zooming operations may be successfully applied depending on the data availability for image construction and the ruling logic for the development of different views. The interrelated items contained in a map provide their accurate position using the GPS technology, RFID techniques, and infrared beems which consist the currently most popular technological choices. The map consumers demand from either a centralized CAM or a structure of collaborating CAM systems complete maps with integrated the interesting participating entities. The elements of maps' content present discrete attributes and properties forming the maps index which is beyond the scope of CAM and the collaboration of another software unit called the Naming System is required, whose description follows.

The CAM operational requirements refer to three building blocks. First, the communications' block interfacing clients' software applications to contact with CAM to request and receive complete maps.The second fundamental block is dedicated in realizing the required processing to produce the map which satisfies the clients' request with an output of a graphical or textual construct. The last of the building blocks deals with the classification of the processed or known information pointing at an external Naming System. In order to retrieve information from the associated Naming System, a sound and complete query is issued which is resolved with adequate syntax and compatible semantics. Each map's scaling factor determines the set of items included each time on its display. Communication among the clients and CAM is accomplished by message passing over the existing network while the interfacing mechanism is equipped with caching to improve the overall turn around time of satisfied requests. The CAM system receives requests for delivering maps. If an identical, with respect to its semantics, request is not found in the employed cache memory, the query is addressed for resolution to the co-existing Naming System to appoint names, attributes and properties to the participating map's entities.

Stationary and mobile clients of CAM systems are classified into categories according to a number of well defined criteria. Allowing either the clients to subscribe into discrete services or to create hierarchies of groups of clients with common characteristics, it provides the advantage to achieve communication with either broadcasting messages to all users, or multicasting messages to specific groups or even unicasting messages to a particular recipient due to the scarce resource of the network's bandwidth. Clients' context-awareness has to be asynchronously updated according to occurred events in the ubiquitous home care environment. Each client's request for a map is actually a query that looks for resolution characterized by the used syntax. A parsing service, at the processing stage of a CAM system has to analyze the received query into simpler ones that are known and cached by the CAM system. The rate of updating the clients' maps depends on two factors:. First, the degree of criticalness of the map's update and second the availability of the network's bandwidth. Depending on the underlying network's bandwidth, CAM system may broadcast an update to all concerned clients, or CAM system periodically transmits information to all subscribed clients of a service, or just a single demanding client is informed. Keeping updated the clients' applications with adequate policies, congestions are avoided.

III. NAMING SYSTEM (NS)

All participating entities of a home care environment need to be discretely identifiable and operating on user's behalf in the subconscious. For the entities' identification and referral codes, humans use names in order to imply and appoints a number of properties and characteristics describing its capabilities and usage within each application's environment. A Naming System (NS) requires a composed architectural structure along with appropriate functioning to support, administer, and handle the managed elements, tending to mimic the operation of humans' implication. Retrieving information from a NS about an entity may be accomplished by either providing the entity's specific name or supplying the entity's characteristics as searching criteria, if known, and then searching among the available entities to distinguish the requested ones that satisfy the given conditions. The absence of a NS obliges the ubiquitous computing applications to hardwire the involved entities' specifications within the executed programming code with particular data structures instead of manipulating such information as externally supplied data carrying out the intended operations. Hence, a NS works as a decrypting mechanism for the ubiquitous computing applications keeping all the known data about each entity, allowing entities to have multiple existence or the same characteristics. The various software applications consult or address the services of the cooperating NS about an entity's identity; either it is going to be about an already enlisted entity with known qualities or about a new entity intending to take part in some computation declaring its properties. The Naming System keeps stored the identities along with the properties and the characteristics of the participating entities which dynamically enter or leave the formed distributed computing environment. The administration of names' attributes is getting even harder taking under consideration the fact that there may exist attributes for which it is impossible to present them quantified, expressed as adjectives. For example the term 'better' or 'wormer' have to be quantified in order to perform computational comparisons.

There are at least three schemes for appointing a name to an entity. First, the entity itself may provide its name. Second the locally ruling software application may assign a name to an entity supplying the Naming System with data. Third, the participating user may declare a name for an entity as a nickname. A set of operations is required to maintain the integrity of the Naming System while another set is necessary to allow interoperability among the Naming Systems pointing at each others' elements. Nevertheless, an NS holds the "experience" of the entities which participated in a ubiquitous computing environment in the past and provides information about the entities whenever it is required by the driving software applications. The entities' names may be classified according to their type: persons, objects, places, and services. Each type may further be analyzed into subtypes depending on the demands of the driving home care applications developing its representative taxonomy. An NS has to provide the means and the capability for the interaction among the stored types, syntactic compliance and semantic matching, obeying the logic rules of pointing at each other through dynamic templates. In its simplest form, NS is required to contain two software modules, implementing temporal logic principles, one that examines the soundness and another one that examines the corresponding completeness of each request.

A classification aspect of Naming Systems is the complexity of the participating entities' syntactic naming rules applied to store or invokes them. The employed internal infrastructure of a Naming System is designed under such principles to facilitate and resolve the issued and requesting queries from the running CAM software applications. The NS has to be designed in such a way that both its architectural structure and followed policies allow the coexistence and distinction of entities sharing names, attributes and properties but keeping their identification autonomy. Binding is another essential characteristic of a NS that is the operation of assigning names to entities and vice versa and it must be facilitated allowing the entities' identification discovery based on the assigned names and provided entities' properties. The internal to NS operation of resolution, revealing entities' identification may be accomplished by successive comparisons even among the output of entities' properties, not just names.

Possible discontinuities in the network services constitute causes of problems in the NS smooth operation since it may miss events and the entities' related data. To overcome such problems, buffering along with time stamps provide an adequate and tolerable solution. Asynchronous mail-boxing software mechanism with a structured memory space capable of accommodating the exported data is required. Any two co-existing and cooperating NS realize the presence of another NS through the running CAM and they have to agree on some synchronization policy since pointers are allowed to address entities from the one of the NS systems to the other. Once, the entity's data happen to be removed from the residing NS, independently from the occurring reasons, the neighboring NS pointing to a removed entity has to stop pointing at that entity anymore, updating its pointer list. Also, an assisting software mechanism has to be applied in such a way that any pointed entity has to be aware that it is pointed. In such a case, whenever a removal of an entity is decided, the NS storing that entity ought to inform the NS holding a pointer at it and specifically the pointed entity. Implementing such a software mechanism requires the NS to feature capability of dedicated space to hold the pointers of the pointing NS entities. Contemporary relational database systems provide a set of options to implement such a synchronization design between any two cooperating distributed databases such as raising rules, exceptions, or even writing directly programming code. Since different NS reside on different machines on the underlying network, the deployment of buffers and polling are considered as possible remedies to overcome the possible network failures.

Home Care applications running in a ubiquitous computing environment refer to the same single service obtaining all of them, the same information in the appropriate format. The CAM system provides in an individualized fashion to each coexisting application the required set of data supported by the NS system. The framework of the proposed system remains in each Home Care installation the same while the administrator is concerned with programming only the system's parameterization.

IV. Conclusions

The NS framework promotes logic inference from the set of available data while CAM models the situations' perception. A minimum of a three layered structure is necessary. The first layer is dedicated to the physical connection of the spread devices. The second layer performs signal processing while the last layer interfaces with the driving software applications. Increasing the number of overseen elements, the network's bandwidth turns to be a scarce resource. Three design measures keep off undesired effects. First, the employed structure organizes elements into classes processing groups' queries. Second, the inclusion of caching improves the promptness. Third, logical reduction and simplification on the queries' complexity to standard forms obtains the same results. Coding temporal logic rules assures both the soundness and the completeness checks of the supplied query saving computational power.

The design principles attempt to set ambiguity as measurable distinguishing three types: Ambiguity of information, closely related to the data validation noticed due to poor coupling with the sensing devices. Ambiguity of context, encountered in the case of incapability to return resolved query results. Ambiguity of relationship occurs when comparison of two or more elements present incompatible data. Quantifying ambiguity with fussy logic principles along with temporal logic rules assist in discovering and remains to be examined.

References

[1] M. Korkea-aho, "Context-Aware Applications Survey", Internetworking Seminar (Tik-110.551), Spring 2000, Helsinki University of Technology. Available: http://users.tkk.fi/~mkorkeaa/publications.html

[2] N. Priyantha, A. Miu, H. Balakrishnan, "The Cricket Compass for Context-Aware Mobile Applications", *Proceedings of the 7th annual international conference on Mobile computing and networking*, MIT, MOBICOM 2001, Rome, Italy, pp: 1–14, 2001.

[3] A. Dey and G. Abowd, "Towards a Better Understanding of Context and Context-Awareness", Presented at the CHI 2000 Workshop on the What, Who, Where, When, Why and How of Context-Awareness, Georgia Tech, April 1-6, 2000.

[4] A. Dey and G. Abowd, "The Context Toolkit: Aiding the Development of Context enabled Applications", Proceedings of the

SIGCHI conference on Human factors in computing systems: the CHI is the limit, Pittsburgh, Pennsylvania, United States, 1999, pp: 434 – 441.
[5] G. Abowd, "Software Design Issues for Ubiquitous Computing, IEEE Proceedings of the 1999 International Conference on Software Engineering, pp: 75-84.
[6] G. Abowd and E. Mynatt, "Charting Past, Present and Future Research in Ubiquitous Computing", ACM Transactions on Computer-Human Interaction (TOCHI), Volume 7 , Issue 1 (March 2000), Special issue on human-computer interaction in the new millennium, Part 1, Pages: 29 – 58.
[7] G. Austaller, J. Kangasharju and M. Muhlhauser, "Using Web Services to Build Context-Aware Applications in Ubiquitous Computing": Lecture Notes in Computer Science, Publisher: Springer Berlin / Heidelberg, Vol. 3140/2004, pp 483-487.
[8] A. Dey, D. Salber, G. Abowd, and M. Futakawa, "An architecture to support context-aware applications", Technical Report GIT-GVU-99-23, GVU Center, Georgia Institute of Technology, June 1999 Georgia Tech. Available: http://www.cs.cmu.edu/~anind/publications.html
[9] W. Li, "A service oriented SIP Infrastructure for Adaptive and Context-Aware wireless Services", Royal Institute of Technology, MUM 2003 Norrköping, Sweden, pp 1-58
[10] B. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications", IEEE Workshop on Mobile Computing Systems and Applications, December 8-9 1994.
[11] J. Crowley, J. Coutaz, G. Rey, and P. Reignier, "Perceptual components for Context-Aware Computing", Proceedings of the 4th international conference on Ubiquitous Computing, Göteborg, Sweden, 2002, pp: 117 – 134.
[12] L. Ronnie, M. Johansson, and N. Xiong, "Perception Management: An Emerging Concept for Information" University of Stockholm, Sweden, 2002, Available: http://www.nada.kth.se/~rjo/pubs/perception_management.pdf
[13] F. Meneses, "Context Management for Heterogeneous Administrative Domains", Conference Pervasive 2004, University of Minho, Portugal 2004.
[14] K. Synnes and P. Parnes, "An architecture for Location Aware Applications, James Nord", HICSS. Proceedings of the 35th Annual Hawaii International Conference, 7-10 Jan. 2002, pp: 3805- 3810.
[15] S. Long, R. Kooper, G. Abowd, and C. Atkeson, "Rapid Prototyping of Mobile Context-Aware Applications: the Cyberguide case study", International Conference on Mobile Computing and Networking, Rye, New York, United States, 1996, pp: 97 – 107.
[16] M. Minami, H. Morikawa, and T. Aoyama, "The Design of Naming-based Service Composition System for Ubiquitous Computing Applications", Applications and the Internet Workshops, 2004, pp: 304-312.
[17] J. O'Toole and D. Gifford, "Names should mean What, not Where", Proceedings of the 5th workshop on ACM SIGOPS European workshop: Models and paradigms for distributed systems structuring, Mont Saint-Michel, France, 1992, pp: 1–5.
[18] B. Bayerdorffer, "An Analytical Taxonomy of Naming Systems", ACM Technical Report: CS-TR-92-48, 1992.
[19] M. Bowman, "Reasoning About Naming Systems", ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 15 , Issue 5 (November 1993), pp: 795 – 825.
[20] W. Adjie Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, "The design and implementation of an intentional naming system", ACM SIGOPS Operating Systems Review, Volume 33, Issue 5 (December 1999) pp: 186 – 201.
[21] M.J. O'Grady, R.P. O' Rafferty, G.M. O'Hare, "The Tourist Centric Mechanism for Interacting with the Environment", Proceedings of the First International Workshop on Managing Interactions in Smart Environments (MANSE'99), Dublin, Ireland, December 1999, pp. 56-67.
[22] J. Roth, "Context-Aware Web Applications using the Pinpoint Infrastructure", IADIS International Conference WWW/Internet, Lissabon, Portugal, Nov, 13-15, 2002.
[23] P. Nichols, "Location-Aware Active Signage", Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Engineering in Computer Science and Engineering at MIT January 2004.
[24] T. Rantakokko and J. Plomp, "An Adaptive Map-Based Interface for Situated Services", VTT Electronics Oulu Finland, 2002, Proceedings of Smart Object Conference, pp 1.
[25] Y. Li, J.I. Hong and J.A. Landay, "ContextMap: Modeling Scenes of the Real World for Context Aware Computing", 5th International Conference on Ubiquitous Computing, 2003.
[26] B.N. Schilit and M.M. Theimer, "Disseminating Active Map Information to Mobile Hosts", IEEE Network, pages 22--32, September/October 1994.