ASTRA: An Awareness Connectivity Platform for Designing Pervasive Awareness Applications

Ioannis Calemis
Computer Technology Institute
Patras, Greece
calemis@cti.gr

Achilles Kameas
Hellenic Open University &
Computer Technology Institute
Patras, Greece
kameas@{eap, cti}.gr

Christos Goumopoulos
Computer Technology Institute
Patras, Greece
goumop@cti.gr

Erik Berg
Telenor Research and Innovation
Trondheim, Norway
erik.berg@telenor.com

Abstract - **Awareness systems are a class of computer mediated communication systems that help individuals or groups build and maintain a peripheral awareness of each other. Awareness systems for informal social use are still in their infancy as a technology and as a research area. Such systems promise to address pressing social problems: elderly living alone, families living apart for large parts of the working week, monitoring the well being of an ill relative, etc. The ASTRA platform, which is being developed in the context of the EU research project ASTRA, provides a generalized solution to the development of awareness applications that are based on the concept of pervasive awareness. In this paper, we shall present how smart objects in a person's environment can be used to capture and convey awareness information under this person's control.**

I. Introduction

Pervasive awareness systems are computer mediated communication (CMC) systems whose purpose is to help connected individuals or groups to maintain awareness of the activities and the situation of each other. In the domain of group-work where awareness systems were first studied, awareness can be defined as "an understanding of activities of others that provides a context for your own activities" [2]. In a more social context, interpersonal awareness can be considered as an understanding of the activities and status of one's social relations, derived from social interactions and communications with them [5]. Awareness systems promise to address pressing social problems: elderly living alone, families living apart for large parts of the working week, monitoring the well being of an ill relative, etc. [8]

An approach for conceptualization of awareness systems in the current domain research proposes the description of the awareness in reference of the activities that a person is made aware of [13]. Based on this approach, Metaxas and Markopoulos [11] introduced an abstract formal model of awareness systems that incorporates related concepts and supports reasoning regarding social aspects of using awareness systems. Their model draws the basic notions of *focus* and *nimbus* by the work of Rodden [12], who applied them in a spatial model of group interaction, in order to address mutual levels of awareness within a virtual environment. Early works in the domain of informal social communication like the concepts developed by the Presence project [6] or the Casablanca project [7] were created as installations that users could use as they were.

The ASTRA platform, which is being developed in the context of the EU research project ASTRA [1], provides a generalized solution to the development of awareness applications that are based on the concept of pervasive awareness, i.e., where awareness information is automatically generated as a result of using personal and home devices and smart objects, which capture and exchange information about the user semi-autonomously. The ASTRA platform and the assorted end-user tools implement the principles of Theory of Connectedness [17], an extension to the focus - nimbus model. Briefly, focus represents a sub-space within which a person focuses their attention. One is more aware of objects inside one's focus and less aware of objects outside of it. An entity's nimbus is a sub-space within which it makes some aspects of itself available to others. This could be its presence, its identity, its activity or some combination of these.

In this paper, we shall present how smart objects in a person's environment can be used to capture and convey awareness information under this person's control. In the next section, we shall give the basic approach and notions we use in order to represent the problem domain. Then, we shall describe how the solution to the problem is supported by ubiquitous computing technology, give a presentation of the ASTRA awareness platform and provide an example scenario using the proposed technology. Finally, our conclusions are outlined.

II. Basic Modeling Framework

In order to support the development of awareness applications, we consider that people conduct their activities within an ambient intelligence space using smart objects and that it is possible to access and combine the services offered by these objects. Our approach is based on the following concepts:

- **AmI space:** An AmI space is to a physical space the same as to what an artifact is to an object. To be more precise, an AmI space embeds sensing, actuating, processing and networking infrastructure in a physical (usually closed) space and offers a set of digital services.
- **Artifacts:** An artifact is a tangible object augmented with computation and communication capabilities. Its properties and services are digitally expressed.
- **Services:** These are offered by an artifact or the AmI space. They could be considered as virtual artifacts. Our approach assumes a service-oriented architecture that

T. Sobh (ed.), *Innovations and Advances in Computer Sciences and Engineering*,
DOI 10.1007/978-90-481-3658-2_32, © Springer Science+Business Media B.V. 2010

enforces a clean service oriented design approach, with a clear distinction between interfaces and implementation [15].

- **Synapse**: A logical connection between services offered in an AmI space. A synapse defines a service composition.
- **Ambient Ecology:** It is the set of artifacts contained in an AmI space and services offered therein; artifacts may be connected through synapses, thus offering more complex services.
- **Spheres:** An activity sphere is deployed over the Ambient Ecology of an AmI space and uses its resources (artifacts, networks, services etc.) to serve a specific goal of its owner. It usually consists of a set of interrelated tasks; the sphere contains models of these tasks and their interaction. The sphere instantiates the task models within the specific context composed by the capabilities and services of the container AmI space and its contained artifacts. In this way, it supports the realization of concrete tasks.

III. An Example of Ambient Ecologies

Let's consider the following scenario:

Students of the Distance Education University (DEU) usually live in disparate locations all over the country. Each of them has his personal matters, but they all have in common their studies at DEU. Punch and Judy are two students of the "Software Design" Teaching Unit; in the past week they have been collaborating in order to study and submit a common project.

Punch is 33 years old, single, working hard and overcommitted. He likes technology and is keen on using new gadgets he discovers in the shops. Judy is a 27-year old single woman, who lives in a small apartment in Santorini. She is a travel agent, and not so fond of technology. Both have installed in their smart homes an Ambient Ecology to support their study.

Punch's Study sphere consists of the following objects: a Book, a Chair, a DeskLamp and a Desk. All of those objects have been augmented with hardware and software in order to provide their services to the ASTRA system. The Desk can sense light intensity, temperature, weight on it, and proximity of a chair. The Chair can tell whether someone was sitting on it. The DeskLamp can remotely be turned on and off. The Book can tell whether it is open or closed and determine the amount of light that falls on it. Collective artifact operation is accomplished by establishing synapses between the constituent artifacts, in order to realize the following behavior:

WHEN this *CHAIR is NEAR the DESK*
AND *ANY BOOK is ON the DESK*,
AND *SOMEONE is sitting on the CHAIR*
AND *The BOOK is OPEN*
THEN *TURN the DESKLAMP ON.*

On the contrary, Judy's sphere is rather simple and only uses the services of a Clock, a Lamp and a Picture Frame. Whenever she starts her study, she sets the Clock timer to 90 mins and connects it to the Lamp; after 90 mins, the Clock alarm goes off and forces the Lamp to flash two times, via their connection.

IV. Awareness System and Applications

The purpose of an awareness application is to convey a person's condition, need or want to a community of users who have subscribed to this application. Usually, an awareness application is developed by a person, who subsequently publishes it to a community, or invites people to subscribe to it.

To the ambient ecology concepts described above we add two basic concepts that originate from modeling of awareness:

- **Focus:** A person's focus is the set of conditions, situations or events that this person is interested in. A person's focus may include another person's nimbus. It is modeled as a set of events that happen in this person's AmI space.
- **Nimbus:** A person's nimbus is the set of conditions, situations or events that this person makes public, i.e. makes them available to become part of some other persons' focus. A person may interact with his nimbus by causing events in his AmI space.

Consider a very simple example. Suppose that Punch every now and then likes to go to the local pub, but he hates going out alone. So he has created a simple awareness application that he calls "out to the pub" and he has invited his friends to join this application.

An awareness application can be considered as a set of conditions and events that convey specific meaning to a defined community of persons. So, regarding the "out to the pub" application Punch has created some rules which signify when he wants to activate this application; for example, he wants to go out when he is not studying or not sleeping, but he does not want to go out when he has company at home. His friends have done the same; of course, each person can create his own rules that activate this application.

So, in order to define an awareness application, a person has to:

- Provide a short textual description of the application and describe its various instances
- Define the conditions that trigger the application and consequently the awareness information to be conveyed – this is his nimbus

- Define the other persons he wants to be aware of this application – they have to include this applications in their focus

So a community is the set of persons that a person allows to have access to his nimbus.

Note that a person may subscribe to an awareness application published by another person. In this case, he has to include this application to his focus.

Based on this framework, we then describe a ubiquitous computing awareness application as an activity sphere, which is instantiated on the ambient ecologies in the different AmI spaces of the various application users [14]. Each instantiation makes use of the different resources in each AmI space and of the artifacts in each ambient ecology and is realized as a set of synapses between the artifacts and the provided services into the AmI space. In order to manipulate their focus, nimbus and awareness applications, people use the artifacts in the AmI space.

V. ASTRA System

In order to support the realization of ubiquitous awareness applications, we have developed a three-tier architecture in which Figure 1:

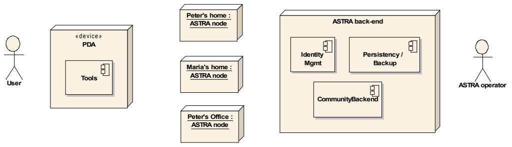


Figure 1. Illustration of ASTRA System

- **End-user tools** implement the presentation layer;
- **Local (user) nodes** are used to support the instantiation of focus and nimbus, and encapsulate the application business logic in terms of rules (application/system logic layer);
- **A centralized remote server** is used to store the awareness application resource descriptions and the community profiles (resource management layer).

Furthermore each local space is comprised of a set of artifacts and services that compose each user's Ambient Ecology.

A. Local Node

A computing system in the person's local space runs the ASTRA node, which provides different end-user services and is responsible for integrating system services, such as context management and service discovery. The platform adopts the Service Oriented Architecture principles (SOA) [15] and makes its resources available as independent services that can be accessed without knowledge of their underlying platform implementation

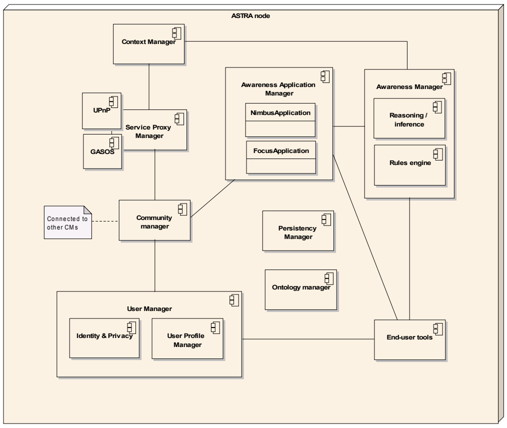


Figure 2. ASTRA Node's Architecture

It has been developed based on the OSGi platform (www.osgi.org) for Service Oriented Computing in Java. OSGi was chosen due to its very elegant and easy way of deploying services. The system is comprised of several different components each one assigned to do a specific task for the whole system.

The functionality provided by the components of the local system is divided in the following categories:

- **Device/Service Manipulation:**

In order to connect the local awareness devices/services to the system we need to identify them, integrate the services to the ASTRA platform and finally analyze and understand their data parameters in order to be able to create applications. This requires, a *Service Proxy Manager* (SPM) which is responsible for discovery of local services. The SPM provides a set of interfaces and implements service proxies for each different communication/middleware technology provided in the local AmI space.

Two service proxies have been implemented in order to connect local artifacts to the system. A GAS-OS service proxy and a UPnP service proxy.

The former service proxy integrates to the ASTRA platform GAS-OS enabled artifacts. GAS-OS enabled artifacts are objects and devices that follow Gadgetware Architectural Style (GAS) [10], a generic architectural style for activity spheres.

GAS adopts the principles of software component technology and service oriented architectures and applies these to the domain of ubiquitous computing, in order to describe the process whereby people configure and use complex collections of interacting artifacts [8]. Each artifact provides its functionality in the form of well-defined services; these are accessible via interfaces. The GAS-OS service proxy tries to identify these services in order to be used by the ASTRA local system.

The later service proxy integrates to the ASTRA platform devices based on the UPnP protocol. Universal Plug and Play (UPnP) is a set of computer network protocols promulgated by the UPnP Forum [16]. The goals of UPnP are to allow devices to connect seamlessly and to simplify the implementation of networks in the home (data sharing, communications, and entertainment) and corporate environments. UPnP achieves this by defining and publishing UPnP device control protocols built upon open, Internet-based communication standards.

All devices and services that are identified by the service proxies are processed by the *Context Manager* whose responsibility is to keep the current state of all artifacts that are identified in a unified format. This knowledge can be accessed in order to connect the local AmI space with awareness applications.

- **Local Awareness Assessment:**

Data gathered by the device/service manipulation components should be combined in order to extract current awareness state of the user. This requires the interaction of the following two components: The *Ontology Manager* (OM) whose responsibility is to provide a common understanding between services and the *Awareness Manager* (AM) which uses the business logic added to the system in terms of rules and infers the current awareness state.

The Ontology Manager’s purpose is two-folded: First it tries to provide a common dictionary for identifying similar services. Thus the system may get for example a lamp that provides the service luminosity and another lamp that provides the service light. OM’s purpose is to identify the similarity of these to services. Thus if the system for any reason needs to replace a light service the ontology is responsible to provide a list of all matching services for replacement. Secondly Ontology tries to identify higher level meanings considering the data of the specific context. Thus, if the system receives data for a service that describes location that state that the user is at his office, this automatically means that he is also in a specific building in a specific town, in a specific country.

The purpose of the Awareness Manager is to get any changes that affect the data of the system, whether those come from the local devices (local context) or from the Community (change of someone’s nimbus) run the rules that the user has specified for his local sphere and take decisions that may affect the current state of the system. Those changes are either transmitted to the Awareness Application Manager (change of local nimbus – let my community know) or to the Context Manager (change of local focus – trigger the appropriate device).

- **Transmission of Awareness Information:**

Any awareness assessment made by the Awareness Manager that affects an application must be transmitted to the involved parties (users or communities). This is done by the Awareness *Application Manager* (AAM) who is responsible for storing and managing the local awareness applications. The AAM supports users in controlling incoming and outgoing awareness information (i.e. their focus and nimbus) by implementing a publish/subscribe model [4]. People who subscribe to the same awareness application are regarded as a virtual community.

An awareness application is a mapping of an awareness state to a service that can be made available to a community, in other words it is a service representation of a specific awareness state. The AAM provisions two types of awareness state applications: nimbus applications and focus applications. Whenever a user wants to share his/her awareness state, she must make a nimbus application. The creation of the application creates the tie with the Awareness Manager, and publishing it will create the link with the Community Manager and the specified community. Any other members of the community can now see that the user has published that awareness state. Every other member can choose to focus on that awareness state, and that is done by creating a focus application.

Apart from the ASTRA system components the local AmI space includes a set of artifacts and services able to perceive user activities (sensing units) or to provide interaction with the system (actuation units) Each of these artifacts and services runs either UPnP [16] or GAS-OS middleware [3]. This enables the discovery of artifact services by the local subsystem and their composition in the context of awareness applications.

An ASTRA user has to define how his focus and nimbus will be realized within an AmI space. For example, Punch has to describe the rules that trigger the awareness application “out to the pub” and also those rules that help the local system deduce his current condition. These rules are defined using the services and states of artifacts in the AmI space. So, Punch could define a rule stating that “when I am not working and it is Sunday evening and I switch my TV set off, then I want to go to the pub”. The information necessary for the system to evaluate this rule can be gathered as follows:

- *Not working:* this information describes Punch’s current condition; it can be received from a central “point of intelligence” in the house, or deduced as a result of a different set of rules, the description of which lies outside the scope of this paper

- *Sunday morning:* this piece of context refers to time and can be received from a calendar service of the AmI space
- *TV set off:* this other piece of context can be directly retrieved from the artifact TV set

When this rule fires, then an event is sent by the local ASTRA subsystem to the ASTRA server. This event is associated with the "out to the pub" awareness application and is propagated to all of its subscribers by the Awareness Manager.

In a similar manner, Punch can define rules describing how he wants to be notified of events that are caused by the other subscribers in the "out to the pub" application. Examples of such rules are: "when I am in the living room, and the TV set is on display a message on the TV screen, otherwise flash the floor lamp twice", or "when I am in the kitchen, show message in the photo frame", etc.

B. Remote Server

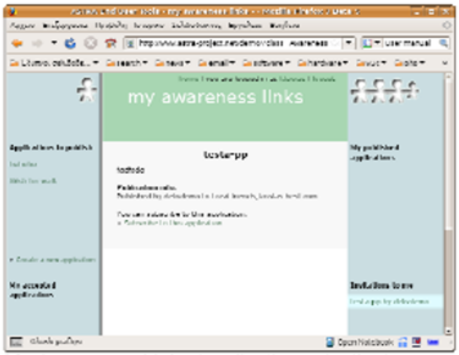
The remote, or back-end, server of the ASTRA platform is responsible for managing identities and providing a backup / persistency solution to components. In addition it also provides the back-end component of the Community Manager, which is responsible for synchronizing community states between the distributed Community Managers of the local ASTRA nodes, as well as providing persistent storage of this state for bootstrapping purposes of the local nodes. This state is comprised of: a) *community members* ***b)*** *shared awareness applications* and c) *shared community space*. The community back-end also supports eventing for changes to any of these parts.

C. ASTRA tools

The ASTRA end user tools use a web interface and connect to the ASTRA platform via a specific API. The tools support user profile management, rule editing, and application management, in a way that semantically relates to the focus / nimbus model, albeit using of a more familiar terminology for end users (Figure 3).

The tools contain the following interlinked modes: *Awareness Connections manager* (where the user can define their Focus or Nimbus), *Pervasive Application manager* (where the user can associate their awareness focus or nimbus to a ubiquitous awareness application), *and User and Communities manager*.

The approach taken in the ASTRA project, when scaled up, has the risk of imposing a heavy semantic load to the user, as she will have to be able to distinguish between various notifications that she will receive and interpret correctly the different events they represent. An obvious solution to this is to use the screens that are available in the ambient ecology (i.e. TV set, mobile, PDA etc) to display semantically rich textual messages, or to use voice synthesis to explain the meaning of the notification.



My Awareness Links (application selection)

Figure 3. Sample ASTRA end user tool

Both these approaches, however, are intrusive, in the sense that they will require the person to shift his attention to the event. A more complicated approach that we are looking into in the project is to use ontologies to create semantically rich descriptions of events and services and then use user-defined policies to deal with conflicts of event interpretation. For example, when the notifications for two events from different awareness applications are similar, then more detailed information has to be conveyed, so as the person can distinguish between them.

VI. An Example Ambient Awareness Application

Now we shall develop the previous example, so as to use the ambient ecology to convey awareness information:

. Recently, DEU is offering an awareness service based on the ASTRA platform for a trial period of one academic year to the student of one specific teaching unit. The ASTRA platform enables subscribed users to form communities and to exchange awareness information and applications between the members of a community.

Punch and Judy have taken advantage of the new DEU service. So, Punch created a DEU Study awareness application and Judy subscribed to it. Punch included in his Nimbus the 'Now Reading' state of his sphere and focused his system on Judy's 'Reading' state. On the other hand, Judy included in her Nimbus the state of her eClock and her PictureFrame; her Focus was set on Punch's 'Now Reading' state.

In Punch's side, whenever he turns on his Study sphere, as his eLamp is switched on, his awareness system sets the value of his ‘Now Reading state’ in his Nimbus. The ASTRA system communicates Punch’s Nimbus to Judy. Judy has Focused on Punch’s ‘Now Reading’ state, and has connected it to her PictureFrame; whenever it changes, her eLamp flashes and Punch’s picture appears. In parallel, as Punch has set his Focus on Judy’s Reading state, whenever she takes a break (as a result of her eClock’s timer reaching zero), his eLamp flashes. Figure 4 shows the awareness system configuration described in the example.

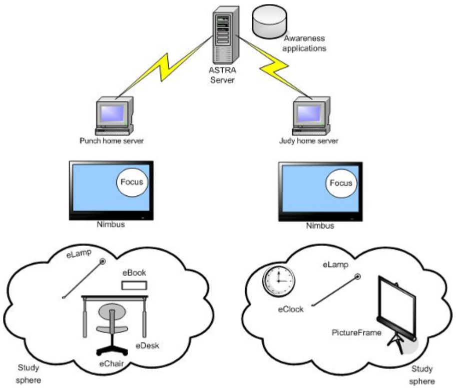


Figure 4. The awareness system of the example

VII. Conclusions

In this paper, we have presented a service oriented architecture that supports the composition and management of ubiquitous computing awareness applications. The aim of this class of applications is to support communication among people without interfering with their task-at-hand.

A three tier system has been developed to support this service: in the person’s local space, interaction among smart objects is achieved with the use of GAS principles; these allow the integration of artifacts running specialized middleware GAS-OS, or other commonly available systems, such as UPnP. In the server side, the specialized ASTRA platform was developed, which offers services for the management of applications and user communities. The architecture presented uses smart objects in the person’s space as conveyors of this person’s awareness condition or want. The person can configure these objects using special end user tools, which support the discovery of artifact services and their combination in a simple way, as well as the definition of awareness applications using first-order logic rules on these services.

Research on ASTRA project continues in order to evaluate and improve the concepts and tools presented in the paper.

Acknowledgment

The described work has been partially supported by EU FP6 IST STREP project ASTRA (Awareness Services and Systems – Towards theory and ReAlization), under Grant agreement No. 29266.

References

[1] ASTRA IST/FET Open project, available in http://www.astra-project.net/

[2] P. Dourish and V. Bellotti, “Awareness and coordination in shared workspaces,” In: Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work (CSCW ‘92), ACM, pp. 107-114.

[3] N. Drossos, C. Goumopoulos, and A. Kameas, “A conceptual model and the supporting middleware for composing ubiquitous computing applications”. Journal of Ubiquitous Computing and Intelligence, American Scientific Publishers, 1(2), 1-13.

[4] P. Eugster, P. Felber, R. Guerraoui and A. Kermarrec, “The many faces of publish/subscribe”. ACM Computing. Surveys, 35, 114–131.

[5] B.A. Farshchian, Presence Technologies for Informal Collaboration, In: G. Riva, F. Davide, W. A IJsselsteijn (Eds.) “Being There: Concepts, Effects and Measurement of User Presence in Synthetic Environments”, Ios Press, Amsterdam, The Netherlands, 2003, pp. 209-222

[6] W. Gaver and B. Hooker, “The Presence Project. London”, RCA: CRD Publishing

[7] D. Hindus, S.D. Mainwaring, N. Leduc, ,A.E. Hagström, and O. Bayley, , “Casablanca: designing social communication devices for the home”. In Proc. CHI 01.

[8] W.A. IJsselsteijn, , J. van Baren and F. van Lanen, “Staying in touch: Social presence and connectedness through synchronous and asynchronous communication media”. In Proc. HCII 2003 volume 2.

[9] A. Kameas., S. Bellis, I .Mavrommati, K. Delaney, M. Colley, and A. Pounds-Cornish, “An Architecture that Treats Everyday Objects as Communicating Tangible Components”. In Proc. PerCom03.

[10] A. Kameas, I. Mavrommati, I. and P. Markopoulos, “Computing in tangible: using artifacts as components of Ambient Intelligent Environments”. In Riva, G., Vatalaro, F., Davide, F. and Alcaniz, M. (eds) Ambient Intelligence, IOS Press, 121-142.

[11] G. Metaxas., and P. Markopoulos., “Aware of what? A formal model of Awareness Systems that extends the focus-nimbus model”. In Proc. EIS 2007.

[12] T. Rodden., “Populating the Application: A Model of Awareness for Cooperative Applications”. In Proc. CSCW 1996.

[13] K. Schmidt, “The problem with “awareness”“: Introductory remarks on “awareness in CSCW”. In Proc. CSCW 2002.

[14] I. D. Zaharakis and A. D. Kameas., “Emergent Phenomena in AmI Spaces”. The EASST Newsletter, Volume 12 (March 2006 / No. 2006 - 12), pp. 82-96. EASST e.V.

[15] T. Erl, “Service-Oriented Architecture: Concepts, Technology, and Design”. Upper Saddle River: Prentice Hall PTR. 2005, ISBN 0-13-185858-0.

[16] UPnP™ Forum, available in http://www.upnp.org/

[17] N. Romero, P. Markopoulos, J. Baren van, B. Ruyter de, I. Wijnand, B. Farshchian, “Connecting the Family with Awareness Systems”, Personal and Ubiquitous Computing, Springer-Verlag London, UK, Volume 11, Issue 4, p.1, 2