# Using Heterogeneous and Social Contexts to Create a Smart Space Architecture

Ahmed Hasswa and Hossan Hassanein

Telecommunications Research Lab, School of Computing, Queen's University
Kingston, Ontario, Canada, K7L 3N6
{hasswa, hossam}@cs.queensi.ca

**Advances in smart technologies, wireless networking, and the increased interest in services have led to the emergence of ubiquitous and pervasive computing as one of the most promising areas of computing in recent years. Smart Spaces in particular have gained a lot of interest within the research community. Most smart spaces rely on physical components such as sensors to sense and acquire information about the real world environment. Although sensor networks can provide useful contextual information, they are known for their high degree of unreliability and limited resources. We believe that it is necessary to augment physical sensors with other kinds of data to create more reliable and truly context-aware smart spaces. In this paper we therefore utilize mobile devices and social networks to acquire more detailed useful contextual information that can help create smarter spaces. We then propose a Smart Spaces architecture that utilizes these new contexts and in particular the Social context.**

***Index Terms*—Smart Spaces, Social Networks, Context Awareness, Heterogeneous Contexts.**

## I. INTRODUCTION

Advances in smart technologies, wireless networking, mobile devices, sensors and the increased interest in services have led to the emergence of ubiquitous and pervasive computing as one of the most promising areas of computing in recent years. Stimulated by the recent popularity of wireless mobile communications as well as the increased availability of location-aware services and contextual information, pervasive computing aims at providing "what you want, when you want it, how you want it, and where you want it" [1] services to users and applications. These services are usually provided within the context of a smart space. A smart space can be defined as an environment that is capable of acquiring and applying knowledge about the environment and its inhabitants in order to improve their experience in that environment [2]. The goal of smart spaces is liberate the users from the mundane tasks that they manually perform to change their environment and meet their requirements. Via smart spaces these environments automatically customize themselves to adapt to user preferences. This is[1] achieved by having the environment detect the current context of entities within the environment at any given moment and providing tailored services. Smart spaces can provide numerous potential uses. They can automate objects within our environment to meet our needs, adapt to our preferences, increase our productivity, customize our shopping experiences, provide targeted services and hence increase our satisfaction and enhance our overall real life experience.

Most smart space research projects rely on physical components such as sensors to sense and acquire information about the real world environment. Wireless sensor networks can be utilized to sense, collect, and broadcast information of the surrounding physical environment, such as temperature, pressure, sound, light, motion, etc. Using this sensory data, the smart space modules can reason about the environment and take an action to change the state of that environment. There is a common belief within the research community that without physical components, the only other way of obtain contextual information is via theoretical algorithms which have very limited or no practical use [3]. Although sensor networks can provide useful contextual information, they are known for their high degree of unreliability and limited resources. Due to physical limitations, sensor nodes have limited energy, networking, computational and storage capabilities. These limitations lead to partial data and uncertainty in the sensed information [4]. Moreover, it is often unreasonable to congregate all the sensed data from each deployed sensor because of the overwhelming amount of data generated and the energy required to harvest such data. Furthermore, wireless sensors can usually provide contextual information about the physical environment but are incapable of accurately predicting social information such as the interests and preferences of users within the environment.

We believe that it is necessary to augment physical sensors with other kinds of data to create more reliable and truly context-aware smart spaces. In this paper we therefore utilize mobile devices and social networks to acquire more detailed useful contextual information that can help create smarter spaces. We then propose a Smart Spaces architecture that utilizes these new contexts and in particular the Social context.

This paper is organized as follows: In Section II we propose new criteria for obtaining context that involves more than physical contexts. Section III presents the system architecture. This is followed by the performance evaluation in Section IV and finally the conclusion in Section V.

This research is supported in part by the Natural Science and Engineering Research Council of Canada (NSERC) and by a grant from Qatar Telecom (QTel).

978-1-4244-7755-5/10/$26.00 ©2010 IEEE

## II. HETEROGENEOUS CONTEXTS

We propose a novel context aware architecture that captures multiple properties from the environment via different sources. The captured contexts are heterogeneous in nature and range from Social contexts acquired from Social Networks to Physical contexts acquired from Wireless sensors. Table 1 presents the different contexts captured by our pervasive computing framework.

| Context | Definition | Source |
|---|---|---|
| Social Context | Information about people within the environment. | Social Networking Services. |
| Location Context | Information about a device's current and future physical locations. | GPS satellite tracking, Cellular tower triangulation, Wi-Fi or Wi-Max networks. |
| Networking Context | Information about the networking capabilities of entities within the environment. This info can usually be used to infer more information about an entity. | Heterogeneous networking interfaces such as Bluetooth, Wi-Fi, Wi-Max, 3G, Cellular networks, etc. |
| Device's Physical Context | Physical information about the orientation of devices within the environment. | Accelerometers, Gyroscopes, etc. |
| Environme nt's Physical Context | Physical information about the environment and entities within the environment, such as pressure, temperature, humidity, light intensity, etc. | Digital Thermometers, Pressure pads, etc. |

Table 1: Heterogeneous Contexts

Social context refers to information about people within the environment. Social context reflects how people around something use and interpret it and consequently the Social Context changes as people within the environment change.

Some pervasive computing projects collect user preferences in an attempt to create a Social Context [4][5]. However these solutions required users to manually enter their preferences into an application via some sort of user interface. These solutions have usually failed at obtaining useful user information due to the users' reluctance to manually provide information about themselves and their interests [6]. Most users are simply not bothered to enter additional information while others are too busy. Asking the users to fill out long forms about their hobbies, interests and preferences is simply impractical and defies the spirit of pervasive computing that aims at building invisible technologies that require minimal user interaction [7]. We are interested in acquiring and creating genuine Social Contexts that truly describe people within an environment. We therefore exploit a recent technology that has significantly increased in popularity over the past few years and that is Social Networking. There has been a huge interest in social networking with networks like Facebook ranking within the top ten most visited websites daily in most developed countries. Social networks contain a wealth of information about the users' interests, preferences, education, career, etc. that can help build more knowledgeable and smarter spaces. By exploiting the fact that most users carry a mobile device wherever they go and by utilizing the wealth of information available through social networking, we are able to acquire accurate Social Context information that we believe can help create a truly ubiquitous, intelligent and adaptive environment that significantly enhances the users' experience.

We therefore focus on designing and developing an architecture that supports Social Contexts obtained from Social Networks.

## III. ARCHITECTURE

In this section the overall architecture is presented. We show how different modules interact and how data is retrieved from social networks.

### *A. Design Goals*

To get the most out of Smart Spaces, a comprehensive context-aware scheme that addresses heterogeneous contexts is necessary. The smart space should offer flexible adaptive service based on varying contexts and policies. This system has been inspired by the currently ongoing research in the areas of pervasive and ubiquitous computing and more specifically context and location awareness. To the best of our knowledge, no research has been conducted on using Social Networks to create a social context and having it combined with other forms of contextual information to produce a more reliable pervasive computing framework. Our design goals include:

*Abstraction:* The smart space should be divided into modules that communicate with each other through interfaces that hide a module's information and classifies them into public and private parts. The architecture should deal with public information and how the different modules function and communicate with each other. Private component details - which only have to do with the internal implementation - are only considered during the simulation.

*Feasibility:* Being able to practically implement and deploy smart spaces is a must. Our proposed system should balance economical and business constraints with technology and networking constraints to produce a cost effective feasible solution.

### *B. Overall Architecture*

In this section we describe the overall architectural concept and data flow through the system.

*On Startup:*

Once the smart space application is loaded it periodically searches for Internet access points. It also provides the user with the option of connecting to a Smart Space remotely. Not all Smart Spaces would allow remote connectivity, but the user would have the option to connect to those that do.

*On Entry into coverage of Network:*

Connecting – The phone operating system requests the user to choose an access point to use to connect to the network. This network access technology could be any accessible technology such as Wi-Fi, Wi-Max, 3G or others. The device must connect to the access point dedicated to the smart space. It is also possible to connect to the smart space via any kind of network. This can be achieved by following the foreign and home agent concept and is discussed in more details in future sections.

Registration with Smart Space – After the IP connection is established, the client must register the user with the Smart Space specific network. If the user has previously connected to a Smart Space then this simply involves logging in. If the user is new to the Smart Space then a registration process takes place.

Registration with Social Network – Once the user is connected to the Smart Space, they need to also register with their public Social Network such as Facebook, Twitter or others and give the Smart Space permission to access their Social Context.

User Listing – Once registration is complete, Smart Space retrieves information about users within the space. It classifies users as either Family, Friends, etc. This classification is based on other users' relationship to our user. Friends and family are the direct friends and family that were retrieved from Social Networks. Potential Friends are people within the smart space have similar interests or hobbies or are registered to the same groups as the user.

Group Listing – The user is also served a list of all active groups within the Smart Space. This list is requested from the IMS Group List Manager.

Presence – Since it is important to get updates on the presence status of all friends, it is necessary to subscribe to presence updates of the contact list. In addition to that, it is important to publish the user's current status too, so that all users can see the status of all users on their contact list. Status updates are carried out periodically and not just

SIP Service Session Initiated – In order to access and communicate with Smart Space services on the IMS AS the client needs to create an IMS session that needs to be managed for the whole duration of the application runtime. At this point it is necessary to therefore initiate a new session between the client and the Application server.

Smart Space Information Requested – With the established session, the client is able to request Smart Space application specific data from the server. For instance it can list all tags, smart zones and maps.

### C. Core Client Services

In this section we discuss the different client-side services and features and show how they will interact with the rest of the system.

***Groups*** – Groups menu feature allows a user to interact, communicate and collaborate instantly with a whole group of people of the same interests. After selecting a user, a user can choose to add this currently selected user as a friend. This results in a friend request being sent to the other user. If the other user approves the friend request the users become friends on the Smart Space and the users' public Social Networks if they are members of the same Social Network(s). Once a user accesses a group, they can leave a message on the group's public forum, they can send private messages to specific group members, they can start group events, share pictures or multimedia and more.

***Private Messages*** – The client consists of a private message inbox that stores any messages sent to the user. These messages can contain text as well as pictures and multimedia. Messages consist of ones sent by other users. Messages can also be system notification and announcements.

***Information Lists*** – There needs to be lists that provide group, presence and other kinds of different information about services provided by the Smart Space. For instance, there needs to be a list of friends who are currently online within the smart space. This list can act as an Instant Messenger. Another list can list all the groups currently within the smart space and maybe allow sorting by relevance to the user based on hobbies and interests. Other lists can include multimedia information, shared information, news, status updates and so on.

***User Profile*** – After selecting a user on the friend list or map, a profile details page is displayed. The profile contains basic information such as name, birthday, profile picture, education info, school, work, department, hobbies, interests, etc. The profile also contains other information such as wall messages and presence and location information. The profile also allows the user to access a buddy oriented menu that enables all interaction and collaboration features such as messaging, file sharing and session initiation. A user can start multimedia sessions such as voice calls or video conferencing.

### D. The Presence and Policies Server (PPS)

Managing presence and policies is a major requirement for providing a truly omnipresent smart social network services platform. Presence services should be capable of providing a multitude of different information to all users within the space. Changes to existing states within the system should be detected by the server and propagated to all users interested in that information. New services added to the space by third-party developers should be able to read, understand and interact with Presence and Policies Servers. Ultimately, services should be able to deliver presence information by understanding users' preferences and requirements.

The general understanding is that presence is a service that allows a user to be informed about the availability of another user. Users can set their status to online, offline, busy, away or anything they like. Users can also share information about their device and its multimedia and other capabilities. We extend this definition of presence to encompass more than just users: Services, on top of users, can now be considered as "users" that have states, features and capabilities. The same applies to environments. This enables users and services to easily interact with one another. Services can notify users of changes to their statuses and information and vice versa.

We develop a presence and policies framework that consists of a presence server and assigns different roles to different entities within the system. Users and services provide presence information and are therefore defined as presentities. The presence information supplied by users and services include properties and attributes, statuses, network capabilities, features, etc. Every device acts as a Presence User Agent and every service has a Presence Service Agent module. Presence User and Service Agents provide information about a user or service's current presence.

A presence user agent knows whether the user is logged in or not and has information about the user's registration status and what kinds of sessions they are currently running. If the user is engaged in any kind of communication the user agent is also aware of that. User agents can even have more information such as user's preferences, what kinds of communications the user is interested in or capable of accepting. And even when a user will be available or what the user is doing right now. All information gathered by User Agents is then forwarded to the Presence and Policies Server.

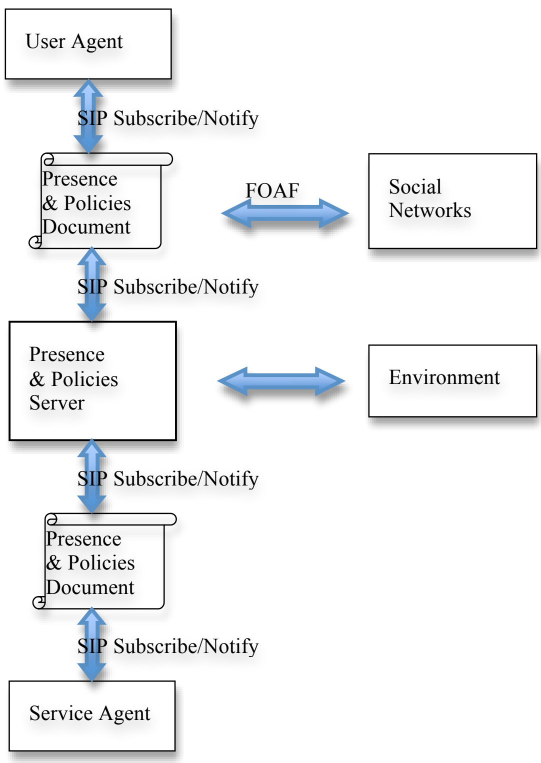


**Figure 1:** Social Network Presence and Policies Architecture

PPS gathers all information about a single user or service and creates a complete picture of that user. It does the same for all users and services within the space and hence creates a complete presence picture of the whole space. The server controls all the information flowing through the presence framework. It coordinates between presentities, presence user agents, service agents and watchers. PPS also acts as a proxy server for subscription requests.

User and Server agents can also act as Watchers that monitor the presence of other presentities that a user is subscribed to and requests updates about changes to their Presence information.

### *E. Representing Social Context*

For a social network dependent system to make use of users' social profiles, a mechanism for identifying different users, reading their profiles is necessary and then translating this information into a format that can be read by other entities within the system. It is better to use an agreed upon semantic representation format such as the Extensible Markup Language (XML). XML is a general-purpose specification for creating custom markup languages. In particular the Resource Description Framework (RDF) could be used. RDF is a family of World Wide Web Consortium (W3C) specifications, originally designed as a metadata data model, which has come to be used as a general method of modeling information through a variety of syntax formats. RDF can be combined with other technologies to describe people, objects, and the relationships between them in an abstract manner. Social networking websites can become rich data sources that are interpreted by the semantic RDF module. This information now becomes machine-readable and can be used to provide an enhanced view and rich set of information about a social network dependent system.

RDF defines a resource as any object that is uniquely identifiable by a Uniform Resource Identifier (URI). The properties associated with resources are identified by property-types, and property-types have corresponding values. Property-types express the relationships of values associated with resources. In RDF, values may be atomic in nature (text strings, numbers, etc.) or other resources, which in turn may have their own properties. A collection of these properties that refers to the same resource is called a description. At the core of RDF is a syntax-independent model for representing resources and their corresponding descriptions. For instance in a social network we can represent a member as shown in Figure 2.

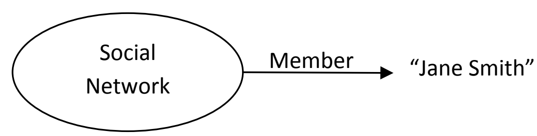


Figure 2: Representing a member in RDF

If additional descriptive information regarding the member were desired, e.g., the member's email address and affiliated network, an elaboration on the previous example would be required. In this case, descriptive information about Jane Smith is desired. Before descriptive properties can be expressed about the member Jane Smith, there needs to be a unique identifiable resource representing her. Given the directed label graph notation in the previous example, the data model corresponding to this description is graphically represented as in Figure 3.

The syntactic vocabulary to be used to describe the profile data extracted is FOAF (Friend of a Friend). FOAF is a decentralized semantic web technology, and has been designed to allow for integration of data across a variety of applications, web sites, services, and software systems. FOAF syntax is written in XML syntax, and adopts the conventions of RDF. By using FOAF syntax, modules within a social network dependent system will be able to exchange and represent social network data in a unified and machine readable manner since FOAF specifies ontology for representing people, objects and the relationships that they share. Using FOAF, people, their attributes and relationships can be modeled using these concepts:

- Person (Object): Each person is represented as a foaf:Person instance.
- Profile (Attributes): Each person has certain properties such as a name foaf:name, gender foaf:gender and profile picture foaf:img.
- Relationships: Each person has friends and relationships which are represented using foaf:knows.

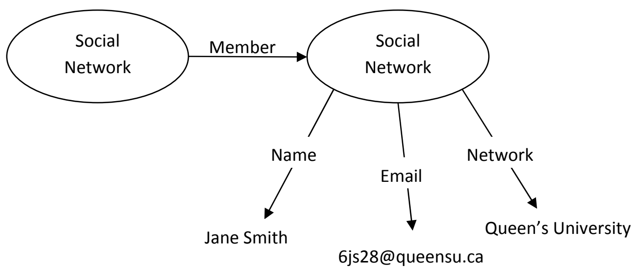


Figure 3: RDF Data model for Social Network

The foaf:Person class represents people. An element is tagged as a foaf:Person if it is a user of the system. foaf:name and foaf:knows are examples of an attribute and a relationship a foaf:Person. These attributes and relationships can also be called properties. FOAF will enable other modules within a system to exchange a large number of properties such as name, gender, age, interests, email, etc. Figure 4 shows a simple FOAF representation for a social network user profile.

We developed the social network based system to test our ontology. The system is device independent but an iPhone client and a Windows client are used for testing. These two clients were simply used as examples and proof of concept. The Presence and Policies manager filters information and only forwards necessary information to other entities within the system. Users are able to connect to any kind of social network, make sure of any service and connect via any access technology. We therefore meet the criteria mentioned in Section III.A.

## IV. Conclusion

There is a wealth of information within Social Networks, which if exploited properly and combined with rules and policies, can lead to a whole new level of smart contextual services. We proposed a mechanism to extract data from heterogeneous Social Networks, link profiles across different networks and aggregate the data obtained. We also designed a Presence and Policies Server that manages the information exchange between Social Networks, services and the environment and passes along the relevant information and rules to different entities. The Presence and Policies server is capable of querying, importing and aggregating data from across multiple Social Networks and services and then converting that data into standardized semantic information that can be interpreted and translated into meaningful information by other users and services.


Figure 4: User Client

## References


[1] Mohan Kumar and Sajal Das, "Pervasive Computing: Enabling Technologies and Challenges," Handbook of Nature-Inspired and Innovative Computing, pp. 613-631, March 2006.

[2] M. Youngblood, D.J. Cook, and L.B. Holder, "Seamlessly engineering a smart environment," in IEEE International Conference on Systems, Man and Cybernetics, Big Island, Hawaii, 2005, pp. 548- 553.

[3] Diane J. Cook and Sajal K. Das, "How smart are our environments? An updated look at the state of the art," Pervasive and Mobile Computing, vol. 3, no. 2, pp. 53-73, March 2007.

[4] Yonghe Liu and Sajal K. Das, "Information-intensive wireless sensor networks: potential and challenges," IEEE Communications Magazine, vol. 44, no. 11, pp. 142-147, November 2006.

[5] S. Counts and K.E. Fisher, "Mobile Social Networking: An Information Grounds Perspective," in Proceedings of the 41st Annual Hawaii International Conference on System Sciences, Hawaii, 2008, pp. 153-153.

[6] Marco von Arb, Matthias Bader, Michael Kuhn, and Roger Wattenhofer, "VENETA: Serverless Friend-of-Friend Detection in Mobile Social Networking," in IEEE International Conference on Wireless and Mobile Computing Networking and Communications, Avignon, France, 2008, pp. 184 - 189.

[7] Andrew T. Campbell et al., "CenceMe: Injecting Sensing Presence into Social Network Applications using Mobile Phones," in In Proc. of Ninth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08), Hong Kong, 2008.