

Context Aware Help and Guidance for Large Scale Public Spaces

Peer-reviewed author version

MAHMUD, Nasim; LUYTEN, Kris & CONINX, Karin (2009) Context Aware Help and Guidance for Large Scale Public Spaces. In: 4th International Workshop on Semantic Media Adaptation and Personalization (SMAP'09). p. 105-110..

Handle: <http://hdl.handle.net/1942/10315>

Context Aware Help and Guidance for Large-Scale Public Spaces

Nasim Mahmud, Kris Luyten and Karin Coninx
Hasselt University - tUL - IBBT,
Expertise Centre for Digital Media,
Wetenschapspark 2, B-3590 Diepenbeek, Belgium
{nasim.mahmud, kris.luyten, karin.coninx}@uhasselt.be

Abstract

Human activity is highly influenced by the context and environment of the user. Presence of other users in the vicinity can also influence the context of use and thus activity. In ubiquitous computing environments, people are surrounded by numerous computing devices helping them to carry out various tasks. However, users seek precise and fine-grained information that is not provided by the computing environment but required to carry out their tasks. Especially nomadic users have little choice than to call upon help from other people present in their vicinity. There are social issues (e.g. availability and interruptibility) along with issues of relevance and reliability to find the most appropriate person to ask for information. Identifying this person is still a challenge today. We have developed software prototype U-Help-System (UHS) to demonstrate how using a social network in combination with the user profile and preferences, can help nomadic person to approximate in selecting people that can support the user in a large scale ubiquitous computing environment.

1. Introduction

Awareness of context and activity provide ubiquitous computing environment with the ability to adapt its services to best meet its user's expectation by extracting the users need from the context. Here context is defined as any information that can be used to characterize the situation of an entity whereas an entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [2].

Ubiquitous computing sketches the future paradigm that promises computing be embedded into the environment in such a way so that the people will consider it as a natural part of their environment [16]. In order to provide help or facilitate human to achieve their goals unobtrusively, the

system should be able to recognize and be aware of the activities and context of the user.

Users activity and context are interwoven characteristics that give meaning to any natural form of communication or interaction. Despite the fact these are interwoven characteristics both context awareness [1, 15] and activity awareness [5, 10] are often considered separate from each other.

With the advancement of modern mobile technology computing devices are becoming ubiquitous, surrounding us virtually all the time. These devices give computational power and communication capabilities that we can exploit in our daily life activity. For example, nowadays searching on the web has become the preferred way of looking for information to support various tasks (e.g looking for a nearby pizzeria, when the next train to Brussels from Hasselt leaves, etc.). But still today the idea of the web as the largest source of information does not imply it is always the best source of information to use in a specific context. In a real life situation we often need to interact with people in our vicinity to seek for help; in many cases the web as an information resource is not an alternative. The web is not situation-aware; it does not take into account the context of use, the specific goals of the user or the current user activities. Moreover, in many situations we prefer asking another human being to whom we can relate ourselves by means of some social relations. This gives us comfort, certainty and also ensures immediate feedback.

On a train or in new city it is more likely a natural or common form of behavior to ask someone for something we need, we want to ask for any help or information from the local people or people relevant with the place, cause of interest or things. This is not only because we feel comfortable while we have access to authentic information; some times it is merely impossible to obtain fine grained information from any other source. But there also is the interpersonal communication barrier or threshold while seeking for information or help from a person we do not have some common ground. In this paper we provide a new way of selecting the most appropriate person by using the social

network of the user in combination with the users' context, goals and activities.

We argue that exploiting social network, user's personal preference, presence of other users in the vicinity and their preferences enable a new era of interaction that helps the user to accomplish tasks more efficiently. It allows to select a suitable person in the vicinity as a source of fine-grained information in a large-scale ubiquitous computing environment.

In our ongoing work we demonstrate how a suitable person within the neighborhood can be invited using *UHS* to come forward and to provide help.

2. Related Work

The era of context aware applications begin with the introduction of Active Badge Location System[15], an infrared technology based tracking system capable of tracking its users current location to forward incoming telephone call closer to the user. From then, most of the context-aware systems focus on the external context [11, 6]. External context or physical context is basically referred to the class of context where context data is collected via physical sensors. It includes context data of physical environment e.g. location, distance, time, temperature, air pressure, lighting levels, surrounding users etc [11, 4].

To provide personalized services according to user preferences focus on internal context is necessary. Internal context [4] describes state of the user. It comprises of task and emotional state of user, the cognitive domains, such as information retrieval, decision making, situation monitoring, and so on.

Model has been proposed to capture the internal context e.g. Schmidt et al. [13] propose hierarchical structure of context to codify both internal and external elements of context. At the top of the hierarchy there are human factors includes user, social environment and task; and physical environment includes conditions, infrastructure and location.

Helping nomadic people by providing context aware information has been proposed e.g. the Cyberguide [1, 9] mobile assistants or tour guides that are aware of the location and orientation of its user and provide information about the surrounding space.

A conceptual model has been proposed [7], showing how social awareness through instant messaging can help mobile learners by providing on-demand sharing of knowledge. In this model social awareness has been described as a *mental concept* where users become aware of social networking by accessing to the same social network or resources regardless of their location and learning context. Thus the learners remain aware of available tutors, lecture and knowledgeable peers should they encounter a learning problem for which they need to consult .

Currently most of the existing approaches exploiting context awareness focus on the external context and there has been little work that addresses internal context. Though this has resulted in some useful context aware and activity aware applications and systems, most of these approaches don't consider more personal and social issues. Existing approaches also don't consider the unpredictable type and amount of human need for fine-grained or too customized queries in mind that can be only addressed by other human being.

The U-Help-System addresses this limitation by using social network, user preference, and presence of other user in the vicinity.

3. Typical Usage Scenario

This section presents a typical usage scenario where UHS can help a user to find context aware help and guidance.

Linda is a regular commuter; she uses train to go to her office. She regularly passed by Brussels city by train. It is somewhere in the middle on her way to work from home. She has visited Brussels many times, but still this city is big enough for her to get lost! Today she needs to visit a University in Brussels. She has an appointment with Susan who works for the University.

Linda has a rough idea about how to go to the university campus. But she wants to know some simple but practical information about the university; where exactly is it, how to get there, which transport is cheaper, faster or safer? She wants to know which direction she should follow after she has reached the nearest station. Linda knows that she would be able to go to the university campus somehow but she also wants to know which path she should follow to get to the venue.

Linda was wondering if there is someone on the train who knows about that university; Linda would feel rather lucky if she could find someone on the same train who is also going to the university! And in the office hours it is not a coincident to find someone with a similar or same destination. But she was wondering how!

4. The U-Help-System

We have developed runtime software prototype the *U-Help-System(UHS)* that exploits two typical properties of end-users searching for help: *locality* and *common ground*. Locality implies users often seek help and guidance related to the area they are located in or heading to. For example, enquiring about the exact location of the baby food in a particular super market is less likely to appear in mind while someone is away from the super market. Similarly, people

```

<foaf:Person rdf:ID="me">
<foaf:name>Noah Jonson</foaf:name>
<foaf:title>Mr.</foaf:title>
<foaf:givenname>Noah</foaf:givenname>
<foaf:family_name>Jonson</foaf:family_name>
<foaf:mbox_sha1sum>617e3e9d9ac9239dbd5dc26c411e20fd6a393d83</foaf:mbox_sha1sum>
<foaf:homepage rdf:resource="http://research.example.com/~noah/">
</foaf:Person>

<foaf:knows>
<foaf:Person>
<foaf:name>Emma Jonson</foaf:name>
<foaf:mbox_sha1sum>b513b6f7a7718967523b69fb38133bd21a5a42a0</foaf:mbox_sha1sum>
</foaf:Person>
</foaf:knows>

```

Figure 1. A social user profile using the FOAF scheme

often want or need to know fine-grained information about destination station while they are on board towards the destination.

Asking a stranger for help is often difficult because people are usually hesitant to initiate interacting with people they are not acquainted with. The second property, *common ground*, indicates users feel more comfortable receiving help and guidance of other users they have something in common with. UHS uses a combination of the social network of a user and the context of use of all users in the social network to seek for the most appropriate help and guidance.

Techniques originating from the semantic web initiative as proposed by Tim Berners-Lee [8] are used to connect different people with each other. A viewer is identified by a Friend-Of-A-Friend (FOAF)¹ profile. This profile contains information about the social relations of the user, next to traditional information that identifies the user (such as name, address, hobbies, school or work environment,...). Figure 1 gives an example of a FOAF profile that is used by the UHS system. This small example shows that FOAF is a Resource Description Framework vocabulary(RDF) [3] for describing people and social networks. This allows us to easily query relations between different FOAF files and build a graph of related FOAF profiles (e.g. figure 2).

UHS uses the process enlisted below to obtain help and guidance. The different steps of this process are:

1. A user has a question and seeks for guidance. The user can access the UHS using her mobile device and input the question.
2. The UHS connects with the UHS server and sends the question together with the context data and user profile to the server.
3. Upon receiving enquiry from any client UHS through the server, the receiving clients run a matching algorithm (see algorithm1).

¹Brickley D., Miller L. FOAF specification versions 0.1, <http://xmlns.com/foaf/0.1/> - Last Accessed September 09, 2009 14:30 CET

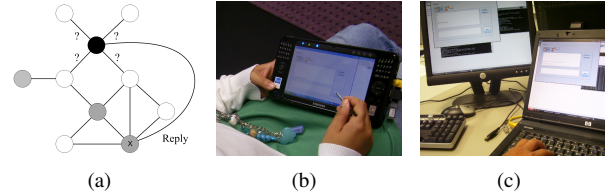


Figure 2. The U-Help-System in action. 2(a) visualizes the social network. 2(b) shows a mobile user querying for assistance and 2(c) shows another user receiving the request.

4. The algorithm takes input from two information sources: local information such as profile and preference information from its local device, which is basically information about the owner of the UHS, and information received from the enquiring UHS client which is the information about a user in the neighborhood who is seeking help.
5. Then the algorithm tries to find a match; upon finding any match the UHS client informs its user so that she can decide whether she wants to address the enquiry by answering using the available communication channel (e.g. the chat system included in UHS, instant messaging or even just talking with each other).

Figure 2 depicts this process where a nomadic user (black circle in the figure 2(a)) needs assistance and is seeking for guidance. She uses her UHS client (figure 2(b)) to connect to the UHS server. By exploiting FOAF the client sends her query to peers (gray circles in figure 2(a)) who are present in the vicinity and also share some common ground with the user. A peer (the gray circle with an X mark, in figure 1(a)) is found as a match and is ready to respond. It notifies its user about the query. The user responded and communication between the nomadic user, who needs some help and the relevant person who is willing to help begins.

In our approach, user profile(see figure 1) is stored on the UHS client device. User profile is described using FOAF, a descriptive vocabulary expressed using the Resource Description Framework(RDF) [3]. Each UHS client can process user profile and find friend of a friend in the neighborhood. This feature made it possible to avoid use of a centralized server, which is commonly used in social network systems where the server is responsible to determine relationship among the users.

The next section will provide more details on the architecture and the behavior of the separate components of the system.

Algorithm 1 UHS matching algorithm

```
1: Input (selfProfile, selfPreference, rcvProfile, rcvPreference)
2: while queryReceived = true do
3:   Parse(selfProfile)
4:   Parse(selfPreference)
5:   Parse(rcvUserProfile)
6:   Parse(rcvUserPreference)
7:   if isFOAF(selfProfile, rcvUserProfile) = true then
8:     if selfPreference.status! = busy then
9:       FindMatch(selfPreference, rcvPreference)
10:    end if
11:  end if
12:  if matchFound = true then
13:    NotifyUser(listWords, matchedRatio)
14:  end if
15: end while
16: FindMatch(selfPreference, rcvPreference)
17: selfKeywords ← Extract(selfProfile)
18: rcvKeywords ← Extract(rcvProfile)
19: matchFound = false
20: for each selfKeyword do
21:   for each rcvKeywords do
22:     if selfKeyword = rcvKeyword then
23:       matchFound = true
24:       countMatch ← countMatch + 1
25:       listWords ← addToList(selfKeyword)
26:     end if
27:   end for
28: end for
29: matchedRatio ← countMatch/selfKeywords * 100
30: return listWords, matchedRatio
```

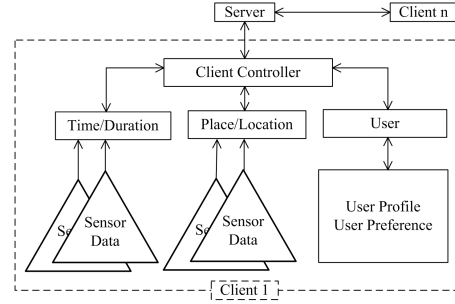


Figure 3. Client structure

5. UHS Architecture

The UHS prototype runs on mobile computing devices such as PDA's and laptops. It keeps track of the user's profile, preference, presence of other users in the vicinity, preferences of other users, current location and time. Figure 3 shows the basic structure of a client and communication with other clients in the neighborhood.

The client controller is responsible for maintaining all communication and computation. An HTTP/XML based communication framework (see section 5.1) is used to facilitate a UHS client to communicate with other UHS clients. The UHS client can send and receive plain text for chatting and also send and receive attached files such as a regular email. When a user needs help and asks query, the controller sends the query to other clients via server. The receiving client controller initiates processing for profile and preference matching and if necessary it notifies its user.

5.1. Web to peer

We have exploited web to peer (W2P) [14]² as foundation for the communication framework. W2P is a middleware framework that enables easy communication among the peers in a network. W2P server is designed as a web application, which is deployable in any servlet container such as Tomcat³ or Jetty⁴.

In the communication framework, a W2P client announces itself as a peer in the network. As a client declares itself a peer it gets incoming and outgoing message channels. Then it can communicate with other clients registered in the same group via the W2P server. The server acts as a message gateway for all the clients registered in a group. A client in the vicinity of a server, are assumed registered to

²Available at <http://research.edm.uhasselt.be/~w2p/> - Last Accessed September 09, 2009 02:30 CET

³<http://tomcat.apache.org/> - Last Accessed September 09, 2009 02:30 CET

⁴<http://www.mortbay.org/jetty/> - Last Accessed September 09, 2009 02:30 CET

the same group e.g. all the clients inside the train wagon are assumed registered to the group called train-server.

5.2. Registering with the server

When the user with the U-Help-System come within the communication range of a server it get notified and the user gets notification for registering to the new location or it can be done automatically depending on the user's preference. The UHS notifies the user about the change, if she agrees and wants to register as a client, she approves and the client declares itself as a Peer in the network. Now it can communicate and exchange information with other clients registered on the same group on the server.

This is necessary because vicinity or nearby depends on several factors that are hard to predict. For example, a person standing on the train platform is nearby to the person sitting inside the train. But as soon as the train leaves the station this relation is no more valid. In such a case, if a client is within the communication range of a server inside the train wagon, the user can effectively decline to registering to the "train-server" and remain registered as peer in the "station-server."

5.3. Selecting profile and preference

User profile is a description file in XML format. The system can automatically select or suggest a profile, e.g. a profile for at school or at home. The user can edit her own preferences. Initially a set of criteria is defined, for example one criterion is hobby and there is list of hobbies that a user can select from the list. Or the user can also insert any other property that is supported by the FOAF specification, so our algorithm can process this information.

For each profile selected there are different kinds of preferences e.g., a profile called out-door-profile may have a preference which might allow the user's friend to chat with her, but on the other hand another preference might not allow to do so.

The user preferences are specified by a XML description file that can be associated with (FOAF) profile. There is a predefined set of preferences to make it simpler, at the same time a user can also create custom preferences.

5.4. Query

Users can initiate a query and broadcast over the network; all the clients present in the vicinity those who are registered to the same group will receive the query. The client will process the query to match with its user profile and preferences. RDF [3] query language SPARQL [12] is used to allow efficient queries over the distributed knowledge base scattered in different client devices within the

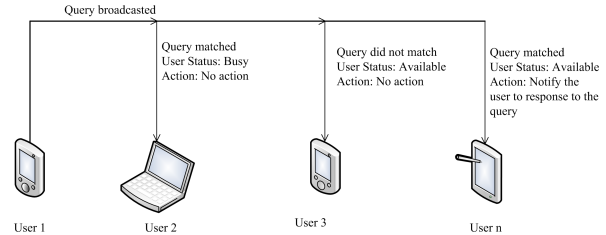


Figure 4. A query structure

vicinity. A matching algorithm matches the profile and preference and suggests its user whether she should respond to the query or not. For example, a query for searching someone who can provide information regarding a local university should be matched with a profile containing the university name and with key words e.g. student or employee. At the same time the client program will also look at the preference for interruptibility state. The client does not immediately respond to the matched query, rather it asks its user that someone needs help and she probably can respond (see figure 4).

5.5. Selecting Context

Context is not only location information. There are different notions of context which is not only location, rather location, time etc [13]. Presence of other users in the vicinity and the users preferences also play a good role in interaction. Selecting a context would be done by exploiting location sensors, time, preference. In our current prototype, the location information is manually specified by the user. Eventually this limitation will be replaced by a location sensing module that exploits UbiSense⁵ technology. Once the system has its location information, it can look into the database and suggest or select a profile depending on the users preference. A user can allow the system to select a profile automatically or she can control this herself by selecting a profile from the suggested set of profiles. For example, the system will select a profile such as "on the train," while the user is on the train and she allows the system to select a profile automatically depending on the available context information. Otherwise, depending on the available context information the system may suggest

- On the train
- On way to University
- On way to interview

And the user will be able to select one from the list.

⁵<http://www.ubisense.net/> Last Accessed July 06, 2009 17:01 CET

6. Discussion and Future Work

The UHS presented in this paper shows how social awareness and context awareness together can be utilized to help nomadic people finding reliable and fine grained information comfortably. In large public spaces such as train station, airport or train wagon, typically people are nomadic and need some situational information very often the queries are too specific. Moreover the type of people varies widely by different factors e.g. age, profession, skill level. So, it is a bigger research challenge to tackle users and their activities in such open public place. However tracking a nomadic user is not only a privacy issue, but also a matter of viability. That is why helping user by rapidly tracking her context, interest, need, preference, skill or ability and providing appropriate information help is difficult. UHS is a semi-automatic approach to handle this. It uses some algorithms to approximate, at the same time asks the user to validate the choice.

The web is not situation-aware, and real life scenario is yet too dynamic to be incorporated and made available for being searched. Moreover, web does not take the context of use into account. That's why a peer to peer interaction is necessary for many reasons. Not only certainty, comfort and immediate feedback but a peer to peer interaction is sometime necessary for people with different range of ability. This is another reason why we often prefer to ask another human being over web based information.

This view of context and context-awareness differs from many of the previous approaches, because it focuses on creating context-aware applications that integrates awareness of the users' social network as a central component to enable context-aware search for *human* guidance. It takes into account the internal context such as preference as well as external context such as location.

UHS enables a text based chat environment where the users can exchange messages either to send query, seek guidance or to summon help to a nomadic user. As the communication established between two users, both parties may agree to discuss in person. But in a large public environment, how can they find and recognize each other conveniently is another challenge which we will address in future work along with the evaluation of the prototype we have developed.

7. Acknowledgments

Part of the research at EDM is funded by EFRO (European Fund for Regional Development) and the Flemish Government. Funding for this research was also provided by the Research Foundation – Flanders (F.W.O. Vlaanderen, project CoLaSUE, number G.0439.08N).

References

- [1] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: a mobile context-aware tour guide. *Wirel. Netw.*, 3(5):421–433, 1997.
- [2] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 304–307, London, UK, 1999. Springer-Verlag.
- [3] D. Beckett and B. McBride. Rdf/xml syntax specification (revised). Technical report, W3C, February 2004. <http://www.w3.org/TR/rdf-syntax-grammar/>.
- [4] J. Gwizdka. What's in the context? *Computer Human Interaction CHI2000*, 2000.
- [5] M. R. Hodges and M. E. Pollack. An 'object-use fingerprint': The use of electronic sensors for human identification. In *Ubicomp*, pages 289–303, 2007.
- [6] J. Hong, E. Suh, and S. Kim. Context-aware systems: A literature review and classification. *Expert Systems with Applications*, November 2008.
- [7] R. M. Kekwaletswe and D. Ngambi. Ubiquitous social presence: Context-awareness in a mobile learning environment. *Sensor Networks, Ubiquitous, and Trustworthy Computing, International Conference on*, 2:90–95, 2006.
- [8] T. B. Lee. *Weaving the Web*. November 1999.
- [9] S. Long, R. Kooper, G. D. Abowd, and C. G. Atkeson. Rapid prototyping of mobile context-aware applications: the cyberguide case study. In *MobiCom '96: Proceedings of the 2nd annual international conference on Mobile computing and networking*, pages 97–107, New York, NY, USA, 1996. ACM.
- [10] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hahnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3(4):50–57, 2004.
- [11] P. Prekop and M. Burnett. Activities, context and ubiquitous computing. *Computer Communications*, 26(11):1168–1176, March 2003.
- [12] E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF. Recommendation, W3C, January 2008. <http://www.w3.org/TR/rdf-sparql-query/>.
- [13] A. Schmidt, M. Beigl, and H. Hans-W. There is more to context than location. volume 23, pages 893–901, 1999.
- [14] G. Vanderhulst, K. Luyten, and K. Coninx. Middleware for ubiquitous service-oriented spaces on the web. In *AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, pages 1001–1006, Washington, DC, USA, 2007. IEEE Computer Society.
- [15] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10(1):91–102, 1992.
- [16] M. Weiser. Some computer science issues in ubiquitous computing. *Commun. ACM*, 36(7):75–84, 1993.