REPUTATION IN USER PROFILING FOR A CONTEXT-AWARE MULTIAGENT SYSTEM

Virginia Fuentes Nayat Sanchez Javier Carbo Jose M. Molina

Group of Applied Artificial Intelligence (GIAA),
Computer Science Dept.,
Carlos III University of Madrid

Abstract

This paper aims at advancing one of the key elements in context-aware services that will make feasible the development of advanced applications of personalized agents: user profiles. These applications should be here understood as ones where mobile services, whichever they are, are provided to the final user in a distributed (agent-based) and personalized way. We examine how to provide this personalization through user profiles inferred mostly from unobtrusively monitored behavior, reducing as much as it is possible the human feedback required. Profiles will contain preferences about services, products, but also about agents representing providers and clients. This last type of preferences take the form of reputations. This part of the profile is then managed by a complex reputation model developped and tested by some of the authors in previous works. Finally, an ontological approach is applied to integrate reasoning about the three types of preferences of the user profile in the messages exchanged by agents.

1 Introduction

The widespread use of new mobile technology implementing wireless communications such as personal digital assistants (PDAs) and smart phones enables a new type of advanced applications. In the past years, the main focus of mobile services researches has been aimed at the anytime-anywhere principle. However, there is more to it. In fact, in several cases, the design and development of effective services should take into account the characteristics of the context from which a service is requested. The mobile computing paradigm in which applications can discover and take advantage of contextual information is the so called context-aware computing. Contextual information can be either the type of the device exploited to access a service, or the location of the user, or its personal preferences, etc. Applications using location are known as location- aware applications. When these applications need to make use of services which provide information according to a given location or approximate location, the term used is location- based services. One example of a location-based service might be to allow the subscriber or user to find the nearest business of a certain type, such as a hotel stand in an exhibition area.

The new mobile technology with wireless communications has similar performance than computers of 10 years ago, and mobiles and PDAs are little enough to carry around you all the time. Additionally numerous types of sensor surveillance our steps in daily life. Both, the permanent availability of this small computers and the possibility of being aware of the context of users in the real world make possible to offer online service to people on the move, wherever they are, this the so called 'ubiquitous computing'. This term denotes a promising environment that it is still premature, and has some weakness.

One of these problems is about coordination among services, since a lot of services can be simultaneously provided to a user (even in the same location), a choosing decision is required. So computerized devices should manage not only explicit information about user's interests but also physical information from sensors. They aim to provide services belonging to a particular domain of interest to the user and can be of use while the user remains in the area where he currently is located. So taking into account the user profile and other contextual data it is a key requirement to provide elaborated personalized services.

Since user profiles contain personal data, and user's preferences or interests, privacy may become a major concern in location-based services. So focusing on a view from users perspective, a centralized entity with complete knowledge should be avoided. In this line, a central server fuses data from all the sensors and provides location information in order to simplify and offer infrastructural support towards generic location retrieval and notification services. Other computational entities would represent providers and users, and would interact directly among them. These personalized autonomous programs that assist users in order to find the desired services would be 'agents' running inside user's mobile handheld devices. An agent-based approach supports adaptive context-aware services by exploiting the capabilities of these autonomous software agents, moving across a network, interacting and negotiating with each other.

So we are interested in the design and implementation of an infrastrure based on agent technology that may provide personalized location-aware services. First steps involved a general definition of an ontology [21], but in this paper we will focus on the link between the location-specific information with the construction and use of user profiles.

2 Related work

There is an increasing need to create context- aware multiagents systems because of the growing interest in mobile devices using different scenarios. At the moment, there are a lot of researches focused on ubiquitous computing and several approaches to previous platforms and applications based only in location-information and using specific domains ontologies.

CONSORT [3] is a multiagent architecture including a middle agent that translate sensor based raw representation of the locations into a conceptual one using a location ontology in two application domains that model the geographical space in a cognitive way: an intelligent assistant at a museum and a wireless-LAN based location system. Another application of location-aware services to museum visitors was proposed by [16]. A similar one is IrReal [4], an information and navigation system based on Palm Pilot PDAs and a set of powerful infrared emitters located throughout a building. Anotherone is Cyberguide [2] project provides user with context-aware information about the projects performed at GVU center in Atlanta. With TV remote controllers throughout the building to detect users locations and provide them with a map that highlights the project demos available in the neighbouring area of the user. Another location-aware application was designed for visitors of 'Filoli', georgian revival house. it highlights the interesting objects of the current room of the house with red borders, when the user clicks in the handheld device over an object, an audio and video becomes activated.

Impulse [27], addresses the problem of location-specific resource brokering, offering location-sensitive restaurant recommendations to mobile individuals. The Hippie system of the HIPS project locates users inside a building using an infrared grid that detects the nearest artwork to the user, and provides information about it.

By other hand,KORE architecture [11] allows recognizing user position by a mobile device in a specific domain and provides the user, according to user location and profile, with the services needed inside the domain. Digital Docent from Newbury Networks, also provides unique location-based content provisioning or push capabilities in an easy-to-deploy application management package. Suited for museums, exhibitions, education campuses and public hotspot operators. An infrastructure for location based services using WiFi is Herecast system [10]. It has applications that may allow the user to bring up a map of their location with a single click, access web sites relevant to their location, publish presence information to their friends. LEAP agent system [8] and SOMA multi-agent platform for mobile devices [9], supports of ubiquitous provisioning of context-aware services to mobile devices.

Although there is an approach to context aware services in all of these systems, personalization of services is only based in location-information, while user-profile-based personalization is often studied from the general view of recommender systems [24]. Examples of recommender systems are Entree [7] which uses a knowledge base and case-based reasoning to recommend restaurant data, and RAAP [18] that uses simple categories to represent user profiles with individual training sets for each user. Like many E-commerce sites, Amazon.comTM (www.amazon.com) is structured with an information page for each book, giving details of the text and purchase information. The Customers Who Bought feature is found on the information page for each book in their catalog. It is in fact two separate recommendation lists. The first recommends books frequently purchased by customers who purchased the selected book. The second recommends authors whose books are frequently purchased by customers who purchased works by the author of the selected book.. Another similar e-commerce recommendation application is seen at the Feedback Profile feature at eBay.comTM (www.ebay.com) that allows both buyers and sellers to contribute to feedback profiles of other customers with whom they have done business. The feedback consists of a satisfaction rating (satisfied/neutral/dissatisfied) as well as a specific comment about the other customer. Feedback is used to provide a recommender system for purchasers, who are able to view the profile of sellers. This profile consists of a table of the number of each rating in the past 7 days, past month, and past 6 months, as well as an overall summary (e.g., 867 positives from 776 unique customers). Upon further request, customers can browse the individual ratings and comments for the sellers.

None of them integrate the user profile into an ontology jointly with a reputation model that would represent user's preferences and interests. On one hand, Quickstep [24] recommender system applies inference to improve user profiling and recommendation accuracy where user profiles were integrated in an external ontology. And on the other hand several proposals of reputation models have been developpped in the last decade. From them, one of the pioneers was AFRAS which represents reputation as a fuzzy concept [12], makes the most from recommendations [13] [9]. This reputation model was designed and tested by some of the authors of this contribution and it recently participated in the ART testbed competition hold in the AAMAS 2006 conference.

3 Context-Aware MultiAgent System

Multiagent systems are adequate for developing applications in dynamic, flexible environments. The context of our multiagent location-based context-aware system (Figure 1) is an exhibition area where several stands act as providers of services, and visitors act as service consumers. The location-aware system could be different domains of services like books, clothes, loans, univ. degrees, laptops, etc. which may be provided and stands may host booksellers, shops, banks, universities, computer manufacturers, etc. A mobile device should locally run a user agent in charge of interfacing with the user and as a repository of local contextual information. It is also in charge of interfacing via FIPA messages with the central agent. The user agents and the central agent exchange information relating to the contextual information the user hold, like his location, his profile, his preferences, etc. So, additionally to the agents representing permanent static providers (they last as long as the system is running) and agents representing temporal mobile users (they last as long as the visitor is inside the exhibition area) there will be a central agent in charge of registering new users, computing the fusion of sensor data and warning provider agents about the closed presence of users.

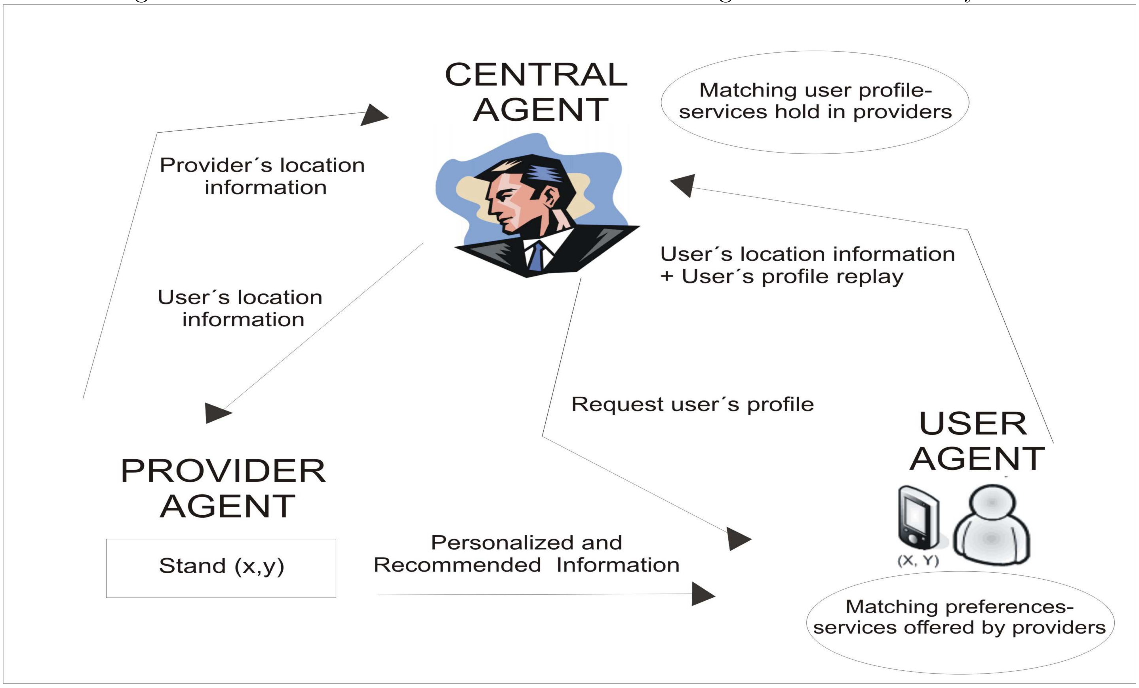
The role played by this central agent is very relevant, since this agent makes the decision of choosing whether to warn or not the closed providers to the user. It accomplish the next tasks:

- Register and ask a brief questionnaire to build an initial profile of the user
- Locate and identify each user moving inside the exhibition area
- Improve the user profile using the sequence (and time-spent) of visited stands of each user
- Match user profiles with features of services provided by closed providers, and if matchmaking succeed, warn providers about the presence of an interested user.
- Deregister users

The role of the other two types of agents is less close to user profile issue:

- Dialogue of users with providers: User agents make decisions, communicate, bargain and possibly make agreements with provider agents according to its internal user profile. Since this profile is just private, it can be assumed to be a complete and accurate model of the user, statically acquired even before this agent join the agent system of the exhibition area.

Figure 1: Overview of the architecture of our multiagent location-aware system



- Dialogue between users: User agents may cooperate with other users, typically asking for recommendations. In order to decide how to trust the other user agent to share opinions about providers, they could compute a private matchmaking of its internal user profiles using secure computations such as [11].

So the role of users, providers and central agents has been briefly described, and now we will explain how they will use the same terms and meanings in these dialogues.

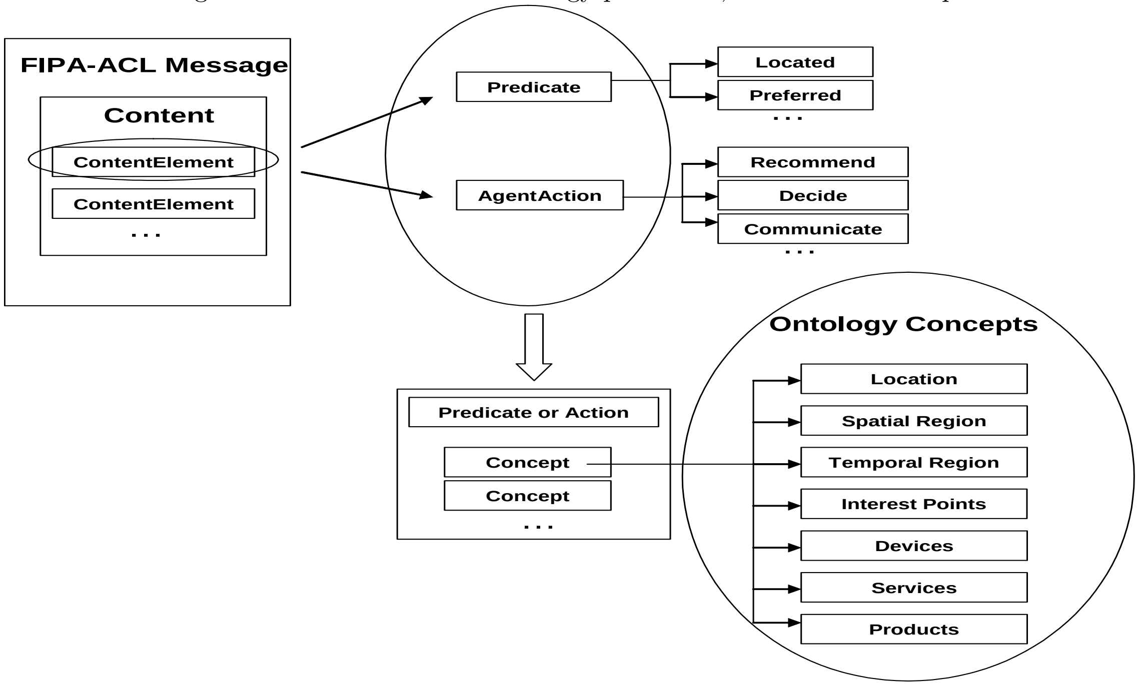
4 Ontology Definition for Contex-Aware MultiAgent System

The main aim of ontologies is to define common vocabulary to share information in the exchanged messages between agents [22]. Ontology application to a multi-agent system describes agents knowledge in communication process, and this communication is achieved by FIPA-ACL and uses ontological concepts for messages [8]. FIPA deals with ACL (Agent Communication Language) in communication process for heterogeneous and interacting agents and agent-based systems. This is the the case of most context aware applications, because visitors and users of airports, universities, exhibition areas, etc. would join the system with agents that may be implemented in whatever ad-hoc language/platform, which should perfectly be able to communicate with context-aware service providers.

The model of agent communication in FIPA is based on the assumption that two agents share a common ontology for the domain of discourse in conversation process. It ensures that the agents describe the same meaning to the symbols used in the message. [20]. To satisfy FIPA-ACL messages content, ontology should define a set of schemas of different types: predicates, concepts and actions.

This approach focuses in predicates and agent actions, related to user preferences and actions developed by agents to operate with these user profiles. Predicates represent a base of facts or expressions that can be true or false, like located, preferred etc. Agent action represents special concepts that indicate some action developed by agents like recommend some product according to user profile, make decisions, communicate etc. In FIPA-ACL messages, the content is a predicate (proposition) or an agent action. Predicates and agent actions are complex elements which are composed by concepts [25]. This relationship between these three aspects defined by ontology is shown in figure 2.

Figure 2: Relation between ontology predicates, actions and concepts



Normally, ontology represents a conceptualization of particular domains, but in this case, the context of use is not limited to a specific environment (each exhibition areas may belong to very different business sector). So, ontology for proposed multi-agent system is a meta-ontology focuses mainly on all concepts definition in order to be applied to a set of heterogeneous domains. The methodology for building the ontology includes four main steps: fix ontology goal, consider the integration of existing ontologies, ontological acquisition, and codification process. For the ontological acquisition step we have combined a Top-Down strategy with Bottom-Up strategy. Top-Down starts with a general vision of the problem, and go down to instantiate the specific concepts of the domain and Bottom-Up strategy obtains a high level abstract conceptualization from different specific domain applications. This combination is needed to define high level conceptualization ontology [21].

In order to develop this ontology for multiple, heterogeneous and dynamical domains in context-aware applications, is necessary to bear some important aspects in mind: indentify the suitable ontology concepts, develop mechanism for security, guarantee accesibility to context information, and integrate context-awareness [19]:

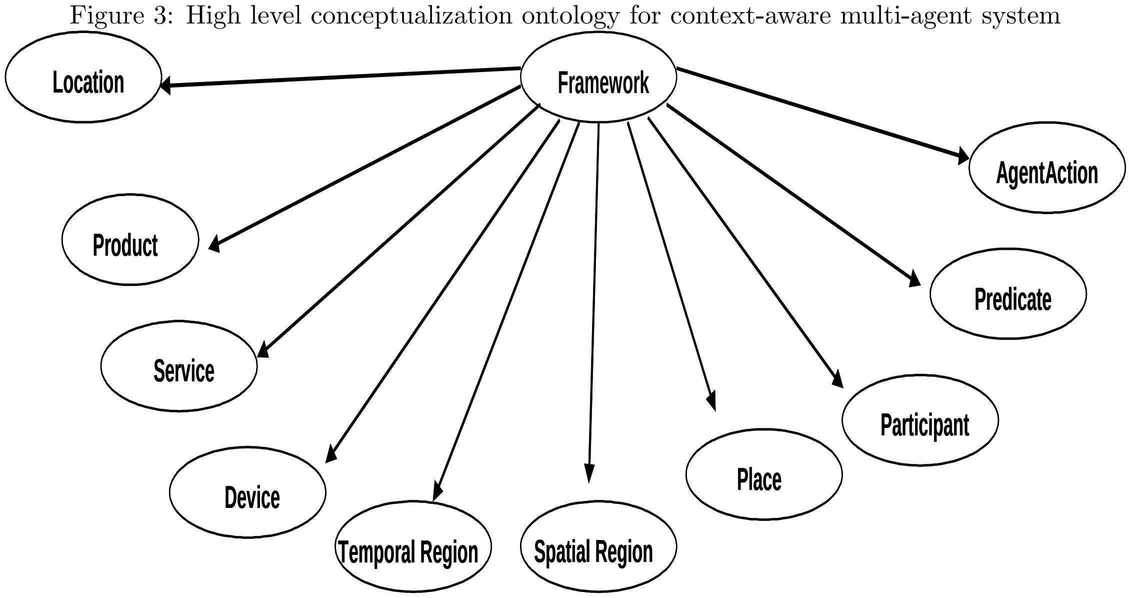
- Identifying the suitable ontology concepts: global created ontology identifies the next high level concepts: "Framework" is the general application concept which includes high level system concepts and it defines the different system domains with two properties: "sector" can be technology, commercial, entertainment etc. and "event" can be fairground, congress, conference, shopping centre etc. "Location" concept is the (x, y) coordinates of any place, participant or object. "Spatial region" and "temporal region", defines the area or map composed by segments according to a range of coordinates, and temporal system information about users in any localization in spatial region, respectively; "Interest points or place" concepts can be stands, departments, conference room, exhibition area etc. and they are defined in one segment or a group of segments, or in a specific coordinates. "Participant" concept can be people, companies with their correspondent role in the system. "Se rvi ce" concept can be any kind of system provision offered to users referred to products in places or interest points like notifications about publicity information, products information etc. "Product" concepts can be offers, solutions, applications, objects etc. that users want to be informed. And finally, "device" concept gathers information about different user's devices profiles and hardware (see Figure 3). [21]

- Develope mechanisms for security: context information exchanged by agents is only available to authorized agents [19]. Central agent is able to specify what information agents should know, and which ones are authorized to receive this information. User profile information is filtered by central agent and sent to provider agents according to the level of authorization of each one. For these reasons, there is a need to establish security mechanisms in central and provider agents. Another kind of security is secure compuntations used in recommendation process between user agents [11]
- Allowing accesibility to context information: users access information anytime and anyware, due to their mobility in different environments [21]. So context information have to be accesible in user mobile devices according to device profiles described in our ontology definition. Central agent is able to recognize what kind of device is, and provide information in the most suitable format that this device can accept.
- Integration context-awareness with user information: mobile devices as PDAs are opening us towards new environments where users can interact between them [15]. User profile and location is combined with context-aware information by means of the use of this kind of mobile devices.

To show how the proposed ontology for context-aware multi-agent system covers different heterogeneous domains, some example scenarios have been tested [21]. One proposed example framework to demonstrate how the system works, is a mobile fairground domain, where users obtain context-aware information according to their specific profile and location, as it is shown in Figure 4.

In this example, users walk freely across the environment, so, while they are in movement, they receive context- aware information according to their profiles and locations. For instance, one user come in a mobile phone fairground, and he registers in the fairground and asks a questionnarie about his profile (preferences information) using his mobile device. In this case, user preferences

Figure 3: High level conceptualization ontology for context-aware multi-agent system



are about the newest mobile phones in market. Central agent receives user profile from user agent, and it identifies user location. With this two parameters (profile and location), central agent matches user profile with information in the most nearby provider agents, and it warns to the appropiated provider agent. There are some stands in the example from different companies (Nokia and Siemens) that match with user profile, but provider agent in Nokia expositor is the most nearby agent. So it can provide context-aware service according to user profile. It sends to user agent a notification service about the last innovations in Nokia mobile phones. Finally, user receives in his mobile device information according to his preferences, so multi-agent system goal of providing context-aware information to users accoding to their profile and location is reached (see Figure 5).

Figure 4: Example domain application

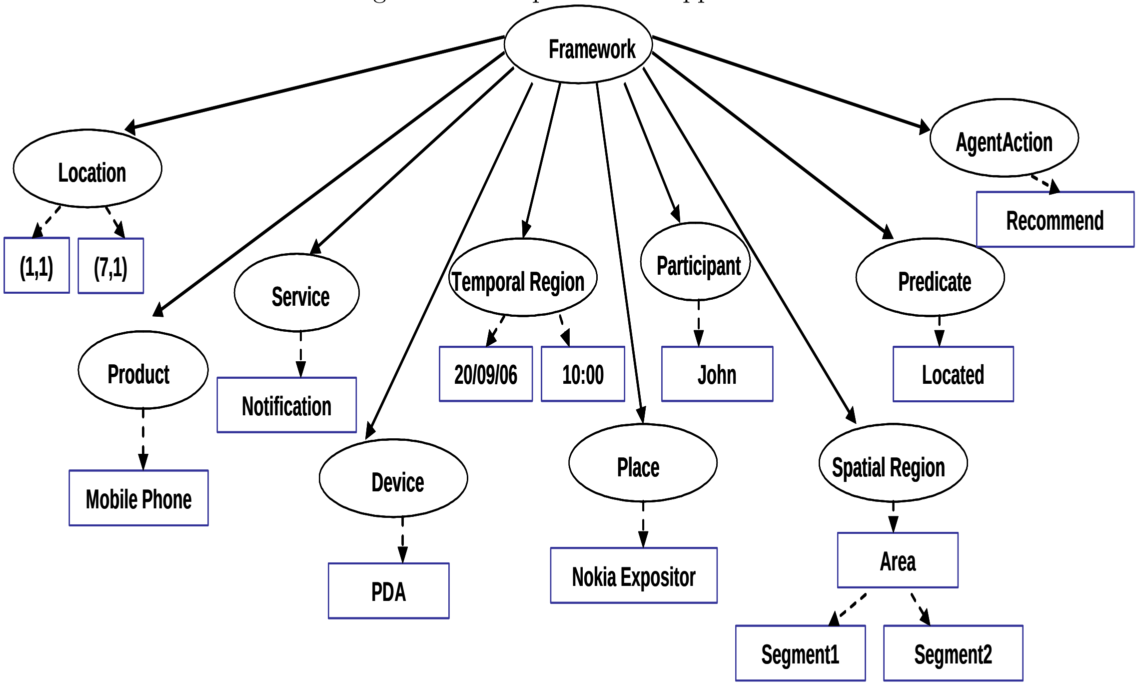
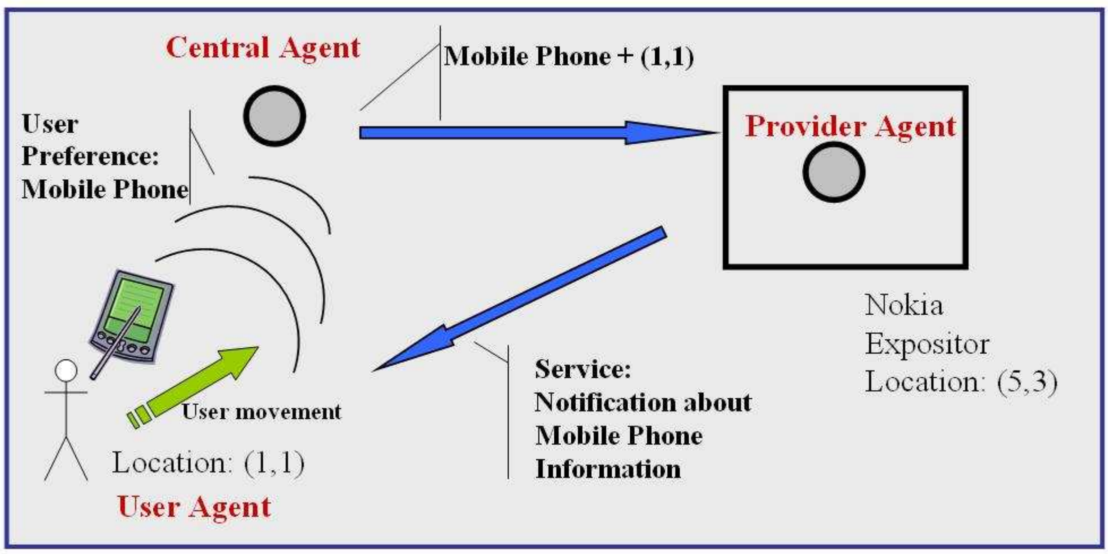


Figure 5: Example framework: fairground domain



On the other hand, there are some aspects about the ontology concepts and profiles that it are necessary to comment. Only services and products are directly linked to a user profile, from all the ontology defined concepts. The model of interests and preferences of users will be expressions about both concepts. Of course both are linked in the ontology, since the services usually $offers$ products. But preferences about products are not enough, since the characteristics of the service rather than the product may define best the interests of the user. These characteristics express the nature and quality of the provided service. It is not just a separation between the attributes of the provider and the attributes of the service, "intangibles" attributes form an integral component of the holistic product offering. In many domains which the service may be not easily subjected to objective evaluation, the ability to be able to ontologically segregate and prioritize various underlying factors related to the services gains relevance.

But they are just concepts; there is a need to express preferences about them using predicates. First of all, the simplest way to build a user profile is using white and black lists, using sets of two predicates: *hasInterest* and *hasDisinterest*. But this view is very limited, and first we could distinguish between hard and soft interests (noted as *requires* and $prefers$ in CommonKads [29]).

Furthermore, simple pair wise comparisons should be also possible using a binary predicate such as $prefers$ *than*. Such comparison may be enriched with linguistic terms that show how much is preferred one feature of product/service rather than other one. So then this predicate would have three arguments, where the third one could be for instance terms such as *a* few, *somewhat*, *much more*, *very much more*. And the other arguments are the user (belongs to participant class) and the object of the preference belongs to services or product class. When the object of the preference is another participant (user or provider) then, we will talk about reputation rather than preference. Predicates can be defined in a formal way, as a ternary relation: **P (participant, object, relevance)**
Some examples of this kind of predicates can be:

- PreferredService(Person(John),Service(Notification),Relevance(somewhat))
- PreferredProduct(Person(John),Product(Phone),Relevance(much more))

Finally the most expressive way to represent preferences is assigning weights (*relevance* predicate) to features but this requires an exhaustive and precise evaluation of the features of services/products, which could be too time-consuming for users while registering into the system. One possible heuristic to facilitate knowledge acquisition from users is to use fuzzy values rather than numeric ones. The use of these human-like expressions probably would make users easier to provide its preferences.

All this kind of predicates would be included in our ontology. Additionally numeric features allow the possibility of expressing preferred ranges rather than preferred values.

Finally, in order to facilitate induced knowledge about user profiles, the ontology has to make possible the specification of certain level of metrics between features. By metrics, we understand statements that may possibly show if a given feature is similar/close to another, incompatibilities between them, counter-productivity, complementarities, etc.

5 Reputation and User Profiles

User profiles are typically either knowledge-based or behaviour based. Knowledge based user profiles are static, and built using knowledge acquisition techniques (such as questionaries) before they are applied. On the other hand, behaviour-based user profiles use the users behaviour itself to build and improve the profile dynamically. The problem of the knowledge-based approach is that it is intrusive and time-consuming, while the behaviour-based approach requires much behaviour-data to build an accurate enough user model.

So the usual trade-off consists of a hybrid system, with a low number of questions applied when user agents join the system, combined with an unobtrusive monitoring the behaviour of the user. The limitations of this approach are that initially the user profile is not very accurate (since it is based in a few number of questions), and that often user's positions alone are not enough to indicate interest in the closest neighbourhood.

This hybrid approach is followed by the system agent. First this central agent will ask some questions about preferences when user agents register in the system. The answers will form an initial model of the user transforming any kind of the preference expressions mentioned before to a numerical rating (*relevance* predicate). Additionally, since the central agent monitors the physical movements of the user, it can induce from its decisions a more accurate user profile. The profiling algorithm performs correlation between time spent in particular stands and its interests. On the other hand, a time decay function will reduce the numerical valuation of relevance of features that did not succeed. We consider failure if the central agent observes that the user rejected to come to a stand after a suggestion from a provider agent. This suggestion was caused by a warning from the system agent because current user profile showed that this provider would be interesting for the user. Since the central agent only knows the path of the users (not what particular services/products they bought/rejected), the value of interest achieved from the observation of the physical path followed by the user, is linked to the provider, not to the services/products (in the case of providers with several services/products).

Finally, another key element of the system is the use of reputation of users to infer trust, and then promote cooperation between users acting as a social filter of cheating/deceptive experiences. Although there are different definitions of trust [17], we can state that trust is an abstract property applied to others that helps to reduce the complexity of decisions. Trust is a universal concept that plays a very important role in social organizations as a mechanism of social control. Unfortunately, when a set of universal objective evaluation criteria is not available, subjective and local trust should be asserted in some way. Although there are several ways to infer such trust, numerous studies have shown that in real life one of the most effective channels to avoid deceptions is through reputation-based mechanisms [5]. Usually, the group of people with good reputation (collaborators, colleagues and friends) that cooperates with a particular person to improve the quality of decisions forms an informal social network [28]. Therefore, modelling reputation in open distributed systems such as agent systems becomes a critical issue since their offline and large-scale nature weaken the social control of direct interactions. Several rust models have been proposed ([1],[23], [31], [26], [30], [14], [6], [12]).

According to this socio-cognitive point of view of reputation, in our context-aware application a reputation value of the user is achieved as a result of the matchmaking computed by the central agent between the user profile and the services of the provider. This represents the value that the central agent assigns to that user for the provider, so the provider may decide to offer different services (or no-service at all) due to this reputation value.

Furthermore, as we said before, dialogues between users would also take place. In them, user agents ask other users for recommendations. In order to decide how to trust in the other user agent to share opinions about providers, they could compute a private matchmaking of its internal user profiles using secure computations such as [11]. Again the result of this private matchmaking would become a reputation value of user agent A (which was requested for recommendations) for another user agent B (which was requesting for recommendations). Depending on the specific value of this reputation of agent A for agent B, agent B would give more or less relevance (weight) to the recommendation from agent A. In fact the recommendations from agent A would also take the form of a reputation value. In this case, the reputation of the provider for agent A.

So finally agent B (which was requesting for recommendations) would aggregate both values weighting the reputation of the provider for agent A with the reputation of agent A for agent B. This is one of the basic computations that form a reputation model, jointly with how reputation values from different sources are aggregated.

So our location-based agent system relyes upon a reputation model to provide a better filtering services. Since some of the authors of this paper have several previous contributions on this issue [12] [10], we plan to integrate such reputation model into this location-based agent system.

6 Conclusions and future work

In this paper we have proposed the general design of a context sensitive agent system. The goal is to provide support for mobile users who are visiting an exhibition area or a fair trade. There, agents that represent users, providers, and the central system, cooperate to avoid time-wasting of users and providers. The cooperation consists of filtering interactions according to the location of users and their interests and preferences (profile). Due to this final aim, we explained how to acquire user profiles reducing as much as it is possible the human feedback required.

Since this is a preliminary work, we do not have completed the implementation of such agents, and therefore there is a lack of experimental results. However this paper provides an innovative approach to location-based services that combines the use of a generic ontology, unobtrusive privacy-protected user profiles based on very different expressions of preferences, and a complex already-tested reputation model. In fact the use of a reputation model integrated in an ontology, and in a more general user profile may make Context-aware services gain relevance in the very next future.

Acknowledgements This work has been supported in part by by projects CICYT TSI2005-07344, CICYT TEC2005-07 186 and CAM MADRINET S-0505/TIC/0255.

References

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Procs. of 33th IEEE International Conference on Systems Sciences*, Maui, U.S., 2000.

[2] G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: A mobile context-aware tour guide, 1997.

[3] Koichi Kurumatani Akio Sashima, Noriaki Izumi. Location-mediated web services coordination in ubiquitous computing. In *2004 IEEE Int. Conf. on Web Services (ICWS 2004)*, pages 822–823, 2004.

[4] B. Andreas, B. Jrg, and K. Antonio. Different views on location awareness.

[5] S. Ba, A.B. Whinston, and H. Zhang. Building trust in online auction markets through an economic incentive mechanism. *Decision Support System*, 35:273–286, 2003.

[6] S. Braynov and T. Sandholm. Trust revelation in multiagent interaction, 2002.

[7] R. Burke, K. Hammond, and B. Young. Knowledge-based navigation of complex information spaces, 1996.

[8] Giovanni Caire. *JADE Tutorial. Application-Defined Content Languages and Ontologies.* FIPA web page, 2002.

[9] J. Carbo and J.M. Molina. Agent-based collaborative filtering based on fuzzy recommendations. *International Journal of Web Engineering and Technology, Special Issue on Web Agents*, 1(4), 2004.

[10] J. Carbo, J.M. Molina, and J. Davila. A fuzzy model of reputation in agent-mediated electronic commerce. In *Procs. of 5th International Conference on Autonomous Agents*, Montreal, Canada, June 2001.

[11] J. Carbo, J.M. Molina, and J. Davila. Secure matchmaking of fuzzy criteria between agents. In *Invited Session on Artificial Intelligence Applications to Information Security, 7th International Conference on Knowledge-based Intelligent Information and Engineering Systems. Lecture Notes in Artificial Intelligence 2773*, pages 1185–1192, 2003.

[12] J. Carbo, J.M. Molina, and J. Davila. Trust management through fuzzy reputation. *International Journal of Cooperative Information Systems*, 12(1):135–155, March 2003.

[13] J. Carbo, J.M. Molina, and J. Davila. Fuzzy referal based cooperation in social networks of agents. *Artificial Intelligence Communications*, 18(1), 2005.

[14] C. Castellfranchi and R. Falcone. Principles of trust for multiagent systems: Cognitive anatomy, social importance and quantification. In *Procs. of the 3rd International Conference on Multi-Agent Systems*, pages 72–79, 1998.

[15] C.Ciavarella and F. Patern. Design criteria for location-aware, indoor, pda applications. In *Proceedings Mobile HCI 2003, LNCS 2795*, pages 131–144, 2003.

[16] Carmine Ciavarella and Fabio Paterno. Design criteria for location-aware, indoor, pda applications. In 5^{th} *International Conference on Human Computer Interaction with Mobile Devices and Services, Lecture Notes in Computer Science 2795*, pages 131–144. Springer Verlag, 2003.

[17] R. Conte and M. Paolucci. *Reputation in Artificial Societies.* Kluwer Academic Publishers, 2002.

[18] Joaquin Delgado, Naohiro Ishii, and Tomoki Ura. Intelligent collaborative information retrieval. In *IBERAMIA*, pages 170–182, 1998.

[19] A. Dogac, G.B. Laleci, and Y. Kabak. A context framework for ambient intelligence. In *Technical Report, NO: EEEAG 102E035, Middle East Technical University*, 2003.

[20] FIPA. Fipa ontology service specification. 2001.

[21] Virginia Fuentes, Javier Carbo, and Jose-Manuel Molina. Heterogeneous domain ontology for location based information system in a multi-agent framework. In 7^{th} *International Conference on Intelligent Data Engineering and Automated Learning*, 2006.

[22] T. Gruger. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.

[23] S. Marsh. Trust in distributed artificial intelligence. In Castelfranchi and Werner, editors, *Lecture Notes in Artificial Intelligence 830*, pages 94–112. Springer Verlag, 1994.

[24] Stuart E. Middleton, Nigel R. Shadbolt, and David C. De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, 2004.

[25] Ambroise Ncho and Esma Aimeur. Building a multi-agent system for automatic negotiation in web service applications. In *Procs. of 3^{rd} International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, volume 3, pages 1466–1467, 2004.

[26] J. Sabater and C. Sierra. Regret: a reputation model for gregarious societies. In *Procs. of the 4^{th} Workshop on Deception, Fraud and Trust in Agent Societies*, pages 61–69, Montreal, Canada, 2001.

[27] Gaurav Tewari, Jim Youll, and Pattie Maes. Personalized location-based brokering using an agent-based intermediary architecture. *Decision Support Systems*, 34(2):127–137, 2003.

[28] S. Wasserman and J. Galaskiewicz. *Advances in Social Network Analysis.* Sage Publications, Thousand Oaks, U.S., 1994.

[29] B.J Wielinga, A.T.H. Schreiber, and J.W. Breuker. Kads: a modelling approach to knowledge engineering. *Knowledge Acquisition*, 4:5–23, 1992.

[30] B. Yu and M.P. Singh. A social mechanism for reputation management in electronic communities. *Lecture Notes in Computer Science*, 1860:154–165, 2000.

[31] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14:881–907, 2000.