A User Profile Ontology For Situation-Aware Social Networking

Johann Stan, Elod Egyed-Zsigmond, Adrien Joly, Pierre Maret

A User Profile Ontology For Situation-Aware Social Networking

Johann Stan[1] and **Elöd Egyed-Zsigmond**[2] and **Adrien Joly**[3] and **Pierre Maret**[4]

Abstract. Today, more and more people possess mobile devices. This enables them to have access to a wide range of services, but also to be contacted anytime, anywhere, which can cause discomfort. People should have full control on who can reach them and how, depending on their current situation: when at work, a friend's call or during a family dinner a call related to work is not always appreciated. Furthermore, situation changes need to be detected in real-time, since preferences change a lot. We present in this paper an ontology-based user profile model, that allows users to have a situation-aware social network, by controlling how reachable they are for specific categories of people in a given situation.

1 Introduction

With the emergence of mobile phones as a daily companion, people can be reached almost everywhere by everybody: a friend might call during a project meeting, or a commercial advert can arrive during a family dinner. Such situations can cause major discomfort. With the popular instant messaging software Yahoo Messenger, the user can define his status for a given contact or group of contacts, using the stealth settings, which allows him to choose for whom he is reachable. This feature is extremely useful, but requires manual settings each time a change occurs. When one has a big social network, this can be even more time consuming than an interruption during the task itself. We consider the following scenario to illustrate this problem:

John Smith is a research engineer. He is currently preparing a presentation for an important workshop, due in 2 days. Since this is a professional situation for him, he only wants office colleagues and some important family members -wife, babysitter- to be able to interrupt him directly (a phone call). Friends or other people should not contact him in a way that would require immediate answer. After work, John becomes less busy, so close friends can also contact him. However, he would be disturbed and annoyed if some work-related call interrupted him during the family dinner. When John is going to work with the train, anyone can contact him by any means.

Based on this example, we can draw several conclusions regarding social behavior:

- Situations often change, varying from professional to private, and for each situation, there is the necessity to have full control of how different categories of people in our social network can contact us: friends, family, office colleagues, the babysitter, and also of how they can do it: by a phone call, an SMS, instant messaging or just by writing an e-mail.
- Another key issue is to reduce the human intervention: situation changes need to be detected in real-time and a corresponding set of user preferences activated for each situation.

A summary of these conclusions could be that there is a necessity to have a straightforward relationship concerning reachability between current situations and the social network of the user. This relationship must be a real-time adaptable interface between the two entities. Similar to the advantages of a recommendation system [15], such a mechanism would reduce organizational activity and would optimize productivity, since from the user's point of view, having a full control of who can interrupt him and how in a given situation would be of real help to perform the current activity with less stress.

Inspired by existing and recently published semantical user profiles,

- We describe a user profile that allows to efficiently characterize the current situation of the user and to express social-network related reachability preferences in situational sub-profiles
- We explain how this model will detect situation changes in real-time and will allow the user to have a full control of his presence in the social network

This paper is organized as follows: after reviewing the existing user profile models in Section 2, Section 3 and Section 4 present in detail our approach. Conclusions and highlights of future work are given in the final Section.

2 Related Work

Recent developments in the field of the semantic web enable a new realm of applications. Semantic web made it possible to have the necessary tools to handle computer-understandable semantics. These tools, generally evolving from XML are used to enrich the description of web-pages, giving a deeper understanding of the relations between the concepts. OWL (Web Ontology Language) [3], RDF (Resource Description Framework) [1] are some of the most widely used representations. The advantage of these languages is that they are machine readable and strongly related to Description Logics. A state-of-the art on this subject can be found in [6].

The RDF language makes statements about resources in the form of triples: (subject, predicate, object). The subject denotes the resource and the predicate denotes the relationship between the subject and the object, which can be another resource or a literal. The OWL language is built on top of RDF, offering a larger vocabulary and stronger syntax.

FOAF (Friend-of-a-Friend) [7] is an ontology-based RDF vocabulary to describe users profiles, friends, affiliations, creations etc. The

[1] Alcatel-Lucent Bell Labs, email: johann.stan@alcatel-lucent.fr
[2] LIRIS CNRS, Lyon, France, email: elod.egyed-zsigmond@insa-lyon.fr
[3] Alcatel-Lucent Bell Labs, France, email: adrien.joly@alcatel-lucent.fr
[4] LIRIS CNRS, Lyon, France, email: pierre.maret@insa-lyon.fr

aim of FOAF is that of a completely decentralized machine-readable social network that is based on personal profiles. The profile contains mostly static data, like personal information, work history, links to contacts and services.

In their work, Carmagnola et al. [8] investigate how tagging allows to infer data about user preferences or interests. Tagging is the process where users label or annotate different resources (web-pages for example) with the objective to share, organize or diffuse them. The way users employ tags might give an insight on different issues like how interested they are in the given resource, whereas the type of tags used (many synonyms for example) can infer subjective details like the level of creativity.

Von Hessling et al. [19] propose a model where semantic user profiles are used in a peer-to-peer mobile environment. The user profile is relatively simple, consisting simply of the union of interests and disinterests. A common domain ontology for concepts in both services and profile description is used to operate the matching with a reasoner. What is interesting in this model is the fact that the system is completely peer-to-peer (profiles are stored on the mobile device), allowing a better privacy.

V. Mendis [12] argues that techniques like RDF and OWL together with ontologies are the key elements in the development of the next generation user profiles. The User Profile Ontology grew out from a quite simple model containing semantic contact information encoded in the RDF language. The proposed ontology is structured in three parts: Person Ontology (containing classes relevant only to the user), Organization Ontology (containing business oriented information) and a Common Ontology (containing information relevant to both persons and organizations), Personal information (e.g. e-mail, telephone, Instant Messaging identifier, physical addresses) is uniquely identified by a GenericContactIdentifier class. Social interactions inherit properties from an Event class. These interactions are classified into voice, text, real-time, online communications. The address book of the user is stored into the class ContactGroup.

Golemati et al. [11] present an application-independent user profile ontology. The objective is to create a "general, comprehensive and extensible" user model taking into account existing literature, user- and context models. It is important to stress out that the proposed ontology deals only with the static profile of the user, not the dynamic or contextual one (like current position, occupation or type of terminal used). The proposed ontology, where the main class is "Person", employs these concepts to create a static profile applicable in any kind of domain or application.

Vildjiounaite et al. [18] address the issue of modeling users in a context-aware smart home environment. Static and dynamic user profiles are distinguished. The Amigo project aims to develop services for context aware house, which offers proactive services to inhabitants according to current context or situation. The user model is separated into two components: the static user profile (preferences, personal data, in interests, disinterests), the context-aware dynamic user profile : this profile learns user behavior from history of activities, learning meaning the ability to recommend a given topic in a given situation (for example a movie when Bob is alone at home on Friday night).

An interesting approach is described in the Doppelgäger User Modeling System [13]. In this approach, the user profile is divided into particular domain submodels and conditional submodels, each containing particular information about the users behavior or context where a set of preferences should be applied. Although the model is very flexible, this system lacks a concrete vocabulary for the user profile.

UPOS (User-Profile Ontology with Situation-Dependent Preferences Support) [17] is part of the Spice project [2]. It is probably the newest user profile ontology, addressing both static and context-aware aspects. This ontology, defined in OWL, allows creating situation-dependent sub-profiles. A user has a profile and a context (location or activity) associated. The notion of condition is defined, which includes a user, an operator and a context-value. For example, a condition can be: "if the context of user Bob equals the MyOffice location...". According to this condition, a corresponding sub profile can be applied that contains all personalization indications for services (e.g. not to use sms). This approach was inspired by the ETSI human factors group, which defined guidelines for designing context-aware user profiles. The most important guideline is to structure the profile into sub-profiles, each containing user preferences that correspond to a specific situation, as seen in the previous example.

Existing user model approaches allow to specify a great variety of static concepts, like personal data, interests, preferences, but they do not allow sufficient expressivity for real-time situation changes. [18] considers dynamic aspects, but this is reduced to the logging of the user activity enriched with context. Tagging user actions could be helpful in the identification of the users activity. However this requires content analysis of interactions (subject of a mail, voice analysis of a phone call), which has technical limitations and presents some privacy concerns. We propose a model based on UPOS (which integrates static and dynamic concepts in a single term) to represent the current situation of the user. The next section shows how our model extends the UPOSs by considering conjunction of context dimensions in order to better identify in real-time the situation of users.

3 Definitions

Researchers attempted to define context in various ways, mostly using terms like location, surrounding people and environmental data, like temperature, period of the day and time. A more comprehensive definition of context can be found in [16], where the user's environment is divided into three parts, the computing environment (network capacity, accessible devices to the user), the user environment (location, social situation) and finally the physical environment (noise level, temperature). [5] defines context as: *"Context is any information that can be used to characterize the situation of an entity. 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"*.

As stated in [20], it is necessary to qualify the information resulted from different context sources (location, time, environment) in a more high-level and time-invariant way, called the situation of the user (eating at home, driving the car, working in the office). A situation abstracts from the context dimensions by translating specific contexts (location, time, temperature, environment, number and list of available network devices) into logical situations. We can not claim that a professional working situation always happens in a location-based context, since one can work also at home or in the train. According to the same article, context-awareness is defined as a capability of a system to provide relevant information or service to the user, relevancy meaning that the provided information helps to better and easier perform the current task in the current context. Derived from this definition, situation-awareness allows a better adaptation of information or services, since the user's situation is much better described, in a time-invariant and concise way.

We consider that the social network of a user is the list of people who had, have or could have a relationship with him/her. This relationship (friends, family, office colleagues), can be direct or indirect (social distance), weak or strong, professional or private etc. It is important to note that these relationships can evolve over time.

A user-profile can be defined as a structured set of entities which cover different dimensions to characterize the user. Such entities can be personal information, interests, disinterests. We define a sub-profile, as a subset of the profile. In our case, such a sub-profile contains a set of social network related preferences, like how (by phone call, by an email, by an SMS) a given person or category of persons can reach him/her in a situation.

4 The Situation-Aware Ontology

4.1 Conditional Sub-profiles in the UPOS Ontology

In the UPOS ontology, a conditional sub-profile is a set of preferences that the user has in a given context. The class *Condition* creates a link between a *Location* (example of a contextual data) and an *Entity*, which is a *Person* with a *Profile*. The *Profile* stores the preferences in that context. These preferences can describe what kind of assistance the user needs in that context or how to adapt the information to better satisfy him. The relation between these concepts is created with an operator, that can take different values (equals, lower, higher,etc. - Figure 1). A drawback of this model is that although the term situation is used, it refers generally to a single context dimension. For example, a conditional expression in UPOS is the following: "if the context of user Bob equals the MyOffice location". We consider that this is rarely sufficient to express situations where the user could have different preferences related to the social network. Instead, situations, like "Working", "Project Meeting, "With family" need to be expressed. These situations are the result of a conjunction of context dimensions related to the physical environment of the user and to his activity or agenda. The number of such situations depends on the occupation, age and many other factors, but rarely exceeds 15-20 (working, eating, driving, watching a movie, walking,...) [17]. An important thing to note is also the fact, that a situation might occur in an unexpected context: one can work in the train or have a project meeting at home. In such exceptional cases, where there is a derivation from routine, the situation is more dependent on the activity or agenda than on the location or environment. An important dimension, that has not been integrated in previous models is the current task of the user. This can be deduced by the kind of objects the task is manipulating: e-mail client, a document-editor, a web-browser. For example, Microsoft Word is manipulated in the scope of the task "Preparation for workshop". Only the integration of all these context dimension concepts can efficiently infer the most probable current situation. Situations can be recognized from raw-data containing context, and this is necessary to keep rules up-to-date with the continuously changing user-habits, but exceptions as described before can only be learned by taking into account semantic information, like the agenda or activity of the user.

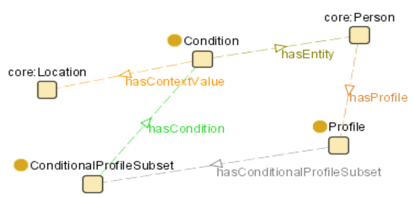


Figure 1. Expressing conditions in the UPOS ontology

4.2 Extension of UPOS for social situation-awareness

We replace the concept Condition with Situation. A Situation defines the current state of the user, but in a more time-invariant way. Context dimensions change frequently, but the situation can still be the same. A good example for this is when John Smith moves from is personal office to the project meeting room. *Location* Context changes, but the user is still in a Working Situation. In our model, we bind a *SituationalProfileSubset* to a *Situation*. This subset of the profile will contain user preferences that need to be applied when that situation occurs. In the example, three situations can be observed: the first is when John Smith is working, the second when he is at home with his family and the third when he is going to work. For each situation, John Smith defines in a *SituationalProfileSubset* his preferences. The advantage of this architecture is the fact that sub-profiles can be easily added or removed. The structure of a condition that describes when a sub-profile can be activated in our case is similar in structure to that of UPOS, but it takes into account more concepts to characterize a situation, since this needs to be time-invariant as much as possible. A fragment of the ontology is shown on Figure 2. A Situation is a ternary relation between a Person, that is the user, a Context and a sub-profile (*SituationalProfileSubset*), that describes preferences in that situation. This gives a dynamic aspect to the user-profile. The Context is a set of multiple contextual dimensions, categorized into two classes:

- The *PhysicalContext*, which describes physical contextual dimensions: *Location*, *Time of the day*, *Environment*. Location can be identified by GPS coordinates or GSM cell identifier. In our case the environment is an indicator of whether the user is surrounded by people or not. This is retrieved by the number of available Bluetooth, Wifi or IR peers.
- *The UserContext*, which contains user-related personal information, which can influence his current situation: the main concepts are the *Activity* and the *Agenda* (scheduled tasks).

The *Activity* is the concept that allows to describe the current state of interaction between the user and his environment. This is extremely important to decide in what kind of situation the user is in. We consider in our *Activity* model, that an interaction between the user and his environment is realized through *Tasks*, which manipulate *Objects* [9]. Therefore, the *Activity* is divided into *Task_Context* and *Object_Context*. A task always concerns one or more objects. John Smith's current task can be to finish the slides for the workshop, and this task manipulates probably a document editor. Interaction tasks however, like calling or having a conversation, concern also people. Due to space limitations, in this paper we only show a fragment of this part of the ontology, since it is a very domain-dependent area that must be specialized for each user. The Task can be extended using task models, like ConcurTaskTrees (CTT) [14] or other, Petri-net based approaches. The *Profile* has a number of *SituationalProfileSubsets*, each of them corresponding to a specific situation, as described before.

The example in Figure 3. gives a better understanding of the core structure of the ontology. We use in the following a Concept:Instance notation to explain the example. *Person:John_Smith* has the following physical context: he is situated

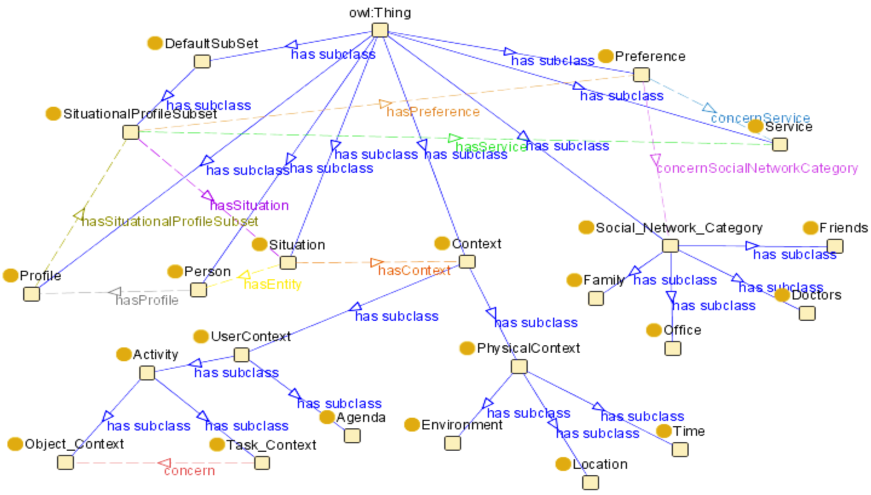
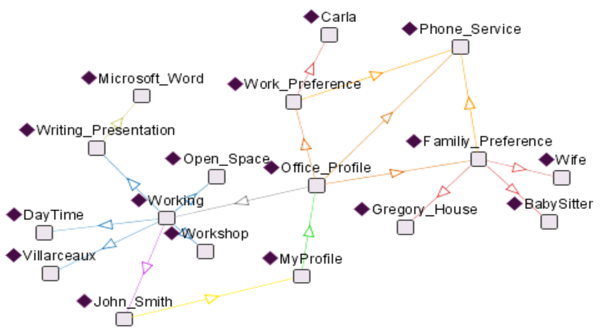


Figure 2. Fragment of the Situation-Aware User Profile Ontology (SAUPO)

in *Location:Villarceaux* (near Paris), at *Time:DayTime* and is in an *Environment:Open_Space*. This means that he is surrounded by people. He has currently an *Agenda:Workshop* on schedule, and has therefore an *Activity:Workshop_Preparation*. This activity has a *Task_Context:Writing_Presentation*, which concerns the *Object_Context:Microsoft_Word*. These last two concepts define his current user context. The different contextual information allows to conclude that John Smith is in *Situation:Working*.

In his corresponding sub-profile (*SituationalProfileSubset: Office_Profile*), he defined two preferences (*Preference:Family_Preference, Work_Preference*). The first defines that if a family member calls (*Family: Gregory_House, Wife, BabySitter*), the phone will ring. The second defines that if a friend calls (*Friend:Carla*) calls, the phone will only *Preference_Option:vibrate*.




```
<Preference rdf:ID="Work_Preference">
  <concernSocialNetworkCategory>
    <Friends rdf:ID="Carla"/>
  </concernSocialNetworkCategory>
  <Preference_Option>vibrate</Preference_Option>
  <concernService rdf:resource="#Phone_Service"/>
</Preference>
```

Figure 3. Snapshot of the ontology instance after the inference of the Working situation (properties were removed for better visibility)

4.3 Preferences in A Situation-Aware Sub-Profile

The user-profile (*Profile*), which contains general information about the user (we reuse the concepts proposed by [11] for expressing static aspects of the user: personal information, interests...), has a set of *SituationalProfileSubsets* associated. Each of them is linked to a *Situation* (*hasSituation* property) and a Service (*hasService* property) containing a list of user preferences. A *Preference* is always related to a Service (*concernService* property), like phone, voice messaging or other and a category of the social network (*Social Network Category* class, (*concernSocialNetworkCategory* property), like *Friends*, *Family* and an option of how that category of people can reach the user via the specified service (Figure 4). This option is stored in an

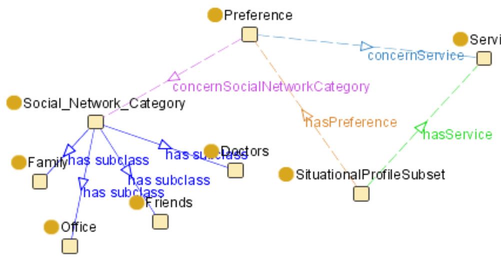


Figure 4. Structure of a preference in the ontology

attribute of the Preference class: *Preference*, which has a set of predefined values that describe notification modalities of a cell phone (vibrate, ring, transfer, voice message,...). For each service, there is list of predefined options for user preferences.

5 Application of the model

After presenting the ontology model for the user profile, we describe in this section how the instances of this profile can be filled with information learned and extracted from user-habits. Learning from user habits is necessary because in this way user the profile expressed with the ontology can adapt itself to the user. We analyzed a number of learning algorithms, based on log files containing context data of users (Frequent Pattern Mining, Decision Trees, Clustering, Support Vector Machines, Bayesian Networks, Hidden Marked Chains, Neural Networks). It resulted in several conclusions regarding the requirements for the learning mechanism, from which the most important are: *online-learning* (no training phase is required, the learning is unsupervised and continuous) and the ability to make *fuzzy classification* (more than one situation can be active at the same time- for example *Working* at *Home*). Clustering algorithms gave the best results in classifying contextual data and in recognizing situations in real-time, especially the Growing Neural Gas, an unsupervised incremental clustering algorithm, first introduced by Bernd Fritzke [10]. We also use frequent pattern mining to process user logs offline after a given period of time to extract frequent associations between the context dimensions. This combination of learning algorithms (clustering and data mining) allows to keep the user profile up-to-date with the user's continuously changing habits, since the concept instances in rules that trigger a situation in the profile will be updated according to the learned patterns. The input of the algorithm is the raw-data containing the context snapshot of the user (location, environment, time, accessible devices) which we preprocess. This phase consists of a set of operations (bitvector generation, normalization of numerical values) that modify the input data (for example from the list of available Bluetooth peers we create a bitvector). The output of the algorithm is a list of detected situations with a corresponding probability value. With this algorithm, usual situations are learned in a satisfactory way, and concept instances in rules are continuously updated in the profile. However, this mechanism is not sufficient to detect exceptions. Therefore, rules containing concepts, like the Agenda or the Activity are checked and privileged, if necessary. This requires a manual intervention of the user, like filling his Agenda. For example, context-data shows that John Smith is at home, but Agenda and Activity show that he has a project meeting and working. In this case, the "Working" sub-profile will be activated.

Once the most probable current situation cluster is recognized, the situational sub-profile is activated. This contains the reachability preferences of the user for the different categories of his social-network. Social-network categories can be manually labeled, and contacts put in a category. The other option is to automatically qualify situational relationships *from the user's point of view* (i.e. how reachable he/she should be to a member of his/her social-network when he is in a given situation). Several criteria need to be considered: frequency of an interaction, means of communication services used. For example, an association with an important weight is created between the babysitter and John Smith's working situation since there is frequent or usual interaction between them in that situation. The qualification of a relationship and the underlying social strength can be refined by taking into account the means of communication services used for the interactions between two people: if the phone, the SMS and the mail is used, then the probability that there is a "friend" or "family" relationship between the two people is higher. Hence, when only one service (like phone or mail) dominates, this relationship is probably more distant. According to these criteria, situational social-strengths can be calculated and updated in real-time. Based on thresholds, the person trying to contact the user will be qualified on-line and a reachability preference depending on the current situation will be activated.

6 Conclusion and future work

We address in this paper the issue of how to model a dynamic, situation-aware user profile that enables real-time situation-awareness and a better control on the reachability of the user. Inspired by existing semantical profiles, we claim that an ontology-based model that allows to specify preferences for a situation in a sub-profile is the best solution for this problem. This profile will be kept up-to-date with information provided by algorithms that learn from user habits. Our model extends UPOS because we consider conjunction of context dimensions in order to better identify in real-time social situation of users. When a situation-change is detected, a view of the social-network will be proposed to the user, that will allow him a better control on who can reach him and how. Currently, our work continues on several directions. In order to better categorize contacts, we consider their belonging to a given social network on the internet, knowing that each network is addressed to specific categories of people: Facebook, Hi5 contains mostly friends, LinkedIn professional acquaintances. The recent OpenSocial Initiative [4], can help to retrieve this information for networks that are compliant with this standard.

REFERENCES

[1] 'Resource description framework (rdf)', *http://www.w3.org/RDF/*.

[2] Spice - http://www.ist-spice.org.

[3] 'Web ontology language (owl)', *http://www.w3.org/2004/OWL/*.

[4] 'Opensocial', *http://code.google.com/apis/opensocial/*, (2007).

[5] Gregory D. Abowd and Anind Dey, 'Towards a better understanding of context and context-awareness', *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, **1707**, 304 – 307, (1999).

[6] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, *The Description Logic Handbook*, Cambridge University Press, 2003.

[7] Dan Brickley and Libby Miller, 'Foaf vocabulary specification 0.91', (November 2007).

[8] Francesca Carmagnola, Federica Cena, Omar Cortassa, Cristina Gena, and Ilaria Torre, 'Towards a tag-based user model: How can user model benefit from tags', *11th International Conference on User Modeling*, **4511/2007**, (2007).

[9] Elod Egyed-Zsigmond, Alain Mille, and Yannick Prie, 'Club (trefle): a use trace model', *Proceedings of ICCBR 03*, 146–160, (2003).

[10] B. Fritzke, 'A growing neural gas network learns topologies', *Advances in Neural Information Processing Systems 7*, 625632, (1995).

[11] Maria Golemati, Akrivi Katifori, Costas Vassilakis, George Lepouras, and Constantin Halatsis, 'Creating an ontology for the user profile: Method and applications', *Proceedings of the First IEEE International Conference on Research Challenges in Information Science*, (2007).

[12] Venura Mendis, 'Rdf user profiles - bringing semantic web capabilities to next generation networks and services', *Proceedings ICIN*, (2007).

[13] Jon Orwant, 'Heterogeneous learning in the doppelganger user modeling system', *User Modeling and User-Adapted Interaction*, **4**, 107–130, (1994).

[14] Fabio Paterno, 'Concurrent task trees environment (ctte)', *http://giove.cnuce.cnr.it/ctte.html*.

[15] Joshua Porter, 'Watch and learn: How recommendation systems are redefining the web', *Available at: http://www.uie.com/articles/recommendation_systems/*.

[16] Bill Schilit, 'Context-aware computing applications', *IEEE Workshop on Mobile Computing Systems and Applications*, (1994).

[17] Michael Sutterer, Olaf Droegehorn, and Klaus David, 'Upos: User profile ontology with situation-dependent preferences support', *Proceedings of the First International Conference on Advances in Computer-Human Interaction*, 230–235.

[18] Elena Vildjiounaite, Otilia Kocsis, Vesa Kyllonen, and Basilis Kladis, 'Context-dependent user modelling for smart homes', *Proceedings of the 11th International Conference on User Modeling*, (2007).

[19] Andreas von Hessling, Thomas Kleemann, and Alex Sinner, 'Semantic user profiles and their applications in a mobile environment', *roc. Artificial Intelligence in Mobile Systems'04 (UbiComp'04)*, (2004).

[20] Norbert Weissenberg, Agnes Voisard, and Rudiger Gartmann, 'An ontology-based approach to personalized situation-aware mobile service supply', *Springer Netherlands*, **10**, 55–90, (2006).