

# Efficient Reachability Management with a Semantic User Profile Framework

Johann Stan<sup>1,3</sup>, Elöd Egyed-Zsigmond<sup>2</sup>, Pierre Maret<sup>3</sup> and  
Johann Daigremont<sup>1</sup>

<sup>1</sup> Alcatel-Lucent Bell Labs France, Centre de Villarceaux,  
Route de Villejust, 91620 Nozay, France  
johann.stan; johann.daigremont@alcatel-lucent.com

<sup>2</sup> Liris UMR CNRS 5205, Université de Lyon, 20 avenue Albert Einstein, 69621  
Villeurbanne, France,  
elod.egyed-zsigmond@insa-lyon.fr

<sup>3</sup> Laboratoire Hubert-Curien UMR CNRS 5516, Universit Jean-Monnet, 18 Rue du  
Professeur Benoit Laurus 42000 Saint-Etienne, France  
pierre.maret@univ-st-etienne.fr

**Abstract.** The possession of mobile devices eased a lot our communication habits: contacting somebody in almost any situation is becoming more and more easier. However, this implies the fact that unexpected interruptions can occur that may cause discomfort to the user. A typical example is the case when an important meeting is interrupted by an unexpected phone call, which is not important to consider in the given context. The need for an efficient control of unexpected incoming communications is immediate. This paper proposes the efficient handling of interruptions by providing a new reachability preference model that considers two dimensions for a reachability preference: (i) the current situation of the user and (ii) the social category of the communication partner. We propose a mechanism to build user preferences that specify the level of interruptibility during an activity, situation or event. The proposed system leverages semantic technologies for modeling the user and data mining techniques for populating the user model.

## 1 Introduction and Related Work

Today, as smart phones become a daily companion for most people, user modeling must address several new challenges. In our point of view, one of them is the issue of how to provide meaningful assistance to end-users in communication needs. One domain is the reachability of the user in a given situation. We define reachability as the preference of the end-user to be interruptible (by a given service, like a phone call, sms, IM, notification) at a given time or time interval. This choice for the definition is mainly based on [3], which claims that users actually discriminate situations (in terms of caller, time, activity, situation etc.) when they have to make decisions about accepting an interruption. Also, an interesting approach of handling interruption is proposed by Schmidt et al.

[8]. In this approach, users of a mobile phone can share their context with the caller with the help of predefined keywords (Home, Meeting, Work etc.). The user has now the choice - knowing the context of the communication partner - to establish a connection, to leave a message, or to cancel the call. Hofte also studies the context information users are willing to share with a person about their current situation in the case of an incoming phone call [9]. [10] uses unsupervised learning, specifically co-clustering, to derive latent situation-based patterns from usage logs of user interactions with the device and environments and use the patterns for task and communication mode recommendations.

From the user's point of view, assistance in reachability is an efficient filtering of interruptions to his current state of mind or activity. As pointed out in [1], interruptions are found mostly intrusive and have negative effects on the mental state of the individual. Few solutions have been proposed for reachability management. The Negotiator system [2] attempts to make interruption handling less cognitively demanding. The system allows the receiver of an incoming call to negotiate with the sender an appropriate time for the interaction, but does not prevent them from occurring. Traditional IM services (Yahoo Messenger, Google Talk, MSN) allow users to explicitly define their availability as a status message: "available", "busy", "on the phone". Another approach is to gather information about the user with sensors (keyboard or mouse activity or environmental data) and thus automatically infer availability. The major problem in existing solutions is that reachability preferences cannot be sufficiently personalized and that there is no automatic control mechanism. In our view, reachability preferences depend mainly on 3 major parameters: the current situation of the user, the social network category of the person behind an attempt to interrupt the user and the device used. These dimensions of a reachability preference were also highlighted by [7]. These 3 parameters allow to specify rich rules like the following: "When (Situation = *Meeting*) and (SocialNetworkCategory = *Friends*) and (Device = *Phone*) then Block". This means that friends will not be able to call the user when he/she is in a meeting. However, the preference could be different for an important family member.

For interoperability purposes, our user model is based on semantic technologies (OWL as description schema, SPARQL query language for information retrieval). This choice is based on current research initiative to leverage semantic technologies for a richer user profile representation. Several user profile models, like GUMO (General User Model Ontology) [4] or UPOS (User-Profile Ontology with Situation-Dependent Preferences Support) [5] are highly relevant to this initiative. We propose in this paper a user profile model design that contains reachability preferences based on the 3 parameters described before. Our user model framework is composed of an ontological front-end for the general problem domain and a back-end that uses data mining and learning techniques to assist the user in the definition of reachability preferences.

The rest of this paper is organized as follows. Section 2 presents the ontology for the user profile model. Section 3 presents an upper-view of the framework. Future enhancements of the system and conclusions are presented in Section 3.

## 2 The User Profile Ontology

The objective of this user model is to allow end-users to have personalizable preferences in terms of reachability that are automatically activated. Preferences are personalized according to three dimensions: the current situation of the user, the service (phone, IM, SMS, Mail) and the social network category (friend, family, professional etc.) of the person who is behind the incoming interaction. We divide the user model into submodels (Figure 1) (*Concept: SituationalProfileSubset*). A submodel contains user preferences that are activated when that specific situation occurs (*Concept: Situation*). A situation is described by several contextual information (*Concept: Context*), composed of the Physical Context of the user (Location, Environment etc.) and the context relative to the user itself (Activity, Agenda, Tasks etc.). Each submodel contains the reachability preferences for the given situation. A preference concerns a *Service* (Phone, SMS, IM etc.) and a *Social Network Category* (Friend, Family, Professional etc.) and contains the notification modality for the Service in case it is activated by a contact belonging to that category. An important aspect of this model is the categorization of the address book into categories (SocialNetworkCategory). This needs a manual intervention of the user but the benefits are considerable.

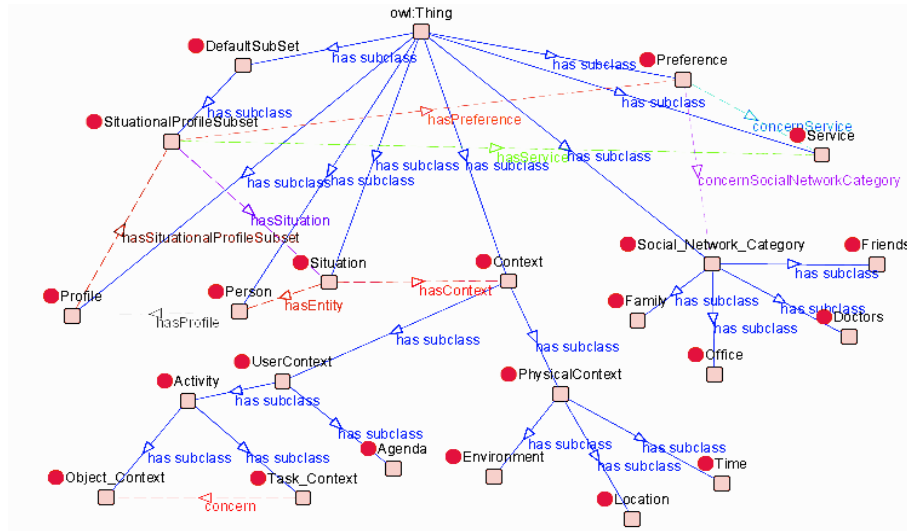
To illustrate the model, we consider an instance with two sub-profiles : (*Home\_Profile* and *Office\_Profile*). Each sub-profile is activated when the given situation occurs (e.g. the user is at home or at work). The *Home\_Profile* is associated to the *At\_Home* situation, which has the following context: the location of the user is Massy (a French town near Paris) - inferred from the Cell Tower Identifier from the mobile phone- , it is evening (Time), the user is with his son (Bluetooth peer detection) and the user is preparing to watch a movie(Agenda). Here each value is an instance of a context concept (Location, Time, Environment, Agenda). In this profile, the user has two preferences, the *Home\_Preference*, which defines that when friends (Carla) and family members call, the phone will ring. The other preference is the *Office\_Preference*, which consists in the fact, that if an office colleague tries to call, he/she will receive a message: "I am currently available on E-Mail.". The second sub-profile is the *Office\_Profile* with 2 preferences. A *Work\_Preference* states that when the manager calls, the phone will ring. The *Family\_Preference* states the same thing about important family members. However, the *Friend\_Preference* states that if a friend calls (Carla), the phone will only vibrate.

## 3 The Reachability Framework

The reachability framework contains tools for both filling the user profile with information from different data sources and for retrieving this information. These tools are separated in a front-end (information retrieval) and a back-end (profile filling, data sources). Both components will be described in the following paragraphs.

The front-end contains SPARQL query templates that are instantiated to retrieve the notification modality from the user model. Such a query template

contains three parameters in input: the situation, the service and the category of the incoming caller and the result is the notification modality for the service. SPARQL was the optimal choice for this, since the user model has an OWL representation.



**Fig. 1.** The user model ontology (main concepts are shown only)

The back-end of the framework contains the necessary mechanism for the automatic profile construction and evolution. Our choice was for a flexible user model in terms of preferences. An argument for this is the fact that neither situations (the location for the Home situation can change), nor social network categories are stable (a friend can become a family member, professional relationships can change etc.). The framework uses a technique that combines data mining and machine learning in order to learn and update context of the different situations of the user. The context of a situation is the primary source to trigger a preference, since it contains machine-interpretable raw-data (a location coordinate, a Bluetooth address). It is important to stress out that this algorithm is only able to find general or regular situations of the user (habits), like Home, Work, Shopping. Another data source is used to handle social events, that occur occasionally, like going to the cinema on Friday night. A situation has a rich contextual description. In the current state of the framework, we are able to capture phone cellular identifiers for location and the list of available Bluetooth peers, that give information about the social environment of the user (single or in an open space):  $Context\ Snapshot = \{Time\ of\ the\ snapshot, list\ GSM\ Cell\ ID, list\ Bluetooth\ Peers\}$ .

In the back-end of the framework, we use a multi-layered algorithm that extracts habits of the user to learn situations and populates the ontology front-end with the learned situations. Situations are strongly related to user habits. A habit is a frequent pattern or item in the user’s context or activity. A frequent pattern is an association of multiple context dimensions that frequently occur together. The signature of a situation is a list of context dimension instances, that indicates in which context the situation occurs for the user. In this way, if we log the user context, finding situations in the context log is a problem similar to finding frequent patterns in any kind of data log and can be solved with data mining techniques [6]. Our choice was to adapt association rule mining [6]. In our context model, generally one Cell ID co-occurs with one or more Bluetooth peer. A frequent pattern is for example the co-occurrence of 40737 (a Cell ID where Bell Labs France is situated) with ROLyyx0021 (Bluetooth peer ID- smart phone of a colleague at work). When these two context instances are present together frequently, it can be inferred that the user is at work: {CellID: 40737, Bluetooth: ROLyyx0021} : Situation: At Work. The feature extraction algorithm uses two parameters, the support and the confidence of an itemset to find frequent patterns. Support is an indicator of whether the item is frequent or not. First the algorithm retrieves all items from the context database where the support measure is superior to a minimal threshold (MinSupp):

$$FrequentItems = \left\{ \bigcup Item, Supp(Item) \geq MinSupp \right\} \quad (1)$$

The second step in this layer is to combine elements from FrequentItems and generate association rules. An association rule of two or more frequent items creates rules with high confidence (superior to a MinConf threshold) in that if the LHS (left-hand-side) is present in a context snapshot, the RHS (right-hand-side) will also be present. Confidence can be interpreted as an estimate of the conditional probability of RHS, knowing the probability of LHS, that is the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.

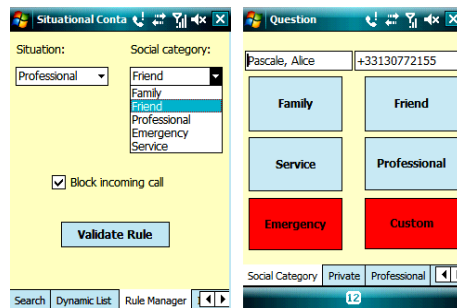
$$FrequentRules = \left\{ \bigcup (A \Rightarrow B), \frac{Supp(A \cup B)}{Supp(B)} \geq MinConf \right\} \quad (2)$$

The result of the first layer is a set of associations between context dimensions in form of rules. These rules encode the signature of a situation candidate with two measures, the support and the confidence. The second layer is responsible of classifying these rules. The classification Layer allows merging similar situations and erasing those that did not occur for a long time. This layer allows the method to be incremental (the database can be erased after extraction, which is very important because of the limited storage space on a smart phone). First, a merge of the LHC and RHC of association rules is realized, creating a signature from the rule. A situation class candidate is created for each signature. For each signature, the algorithm computes the distance to all other (existing already or extracted) classes, except the signature itself. Distance is computed by the number of co-occurrences in the two signatures. Then, the signature is unified

with the class that is the most similar. A local age parameter is associated to each situation class. This parameter allows to erase classes that no longer appear.

The result of this classification step is a limited number of situation signatures, which have the following properties: each context dimension respects the minimal support threshold, the signature is a union of very similar association rules and contains strongly correlated elements since this union is based on association rules respecting the minimal confidence level and this signature contains all possible context dimensions that have a high probability to occur when the user is in the given situation.

As mentioned before, this algorithm is able only to find regular situations. The detection of exceptions (in this case occasional events) is a big challenge in any data mining approach. We consider the user calendar (Agenda concept in the user model) as an important, yet unexplored source for this category of situations. Users often introduce their social events in a calendar application (Google, Outlook). The framework proposes a mechanism to the user to define the reachability preference for a social event. This can be considered as an extension of the calendar template, where typically one specifies the name of the event, the participants and the time interval. Our framework proposes a mechanism that prompts the user before the event to specify his/her reachability preference. The only requirement for this to work is that the user fills regularly in his/her calendar application the upcoming social event. Currently an implementation for mobile devices is deployed and we are gathering user feedback. The implementation also contains a user-friendly interface that allows to tag the social network category of a contact. The tagging happens after an interaction with the given contact through pop-ups with buttons (Friend, Professional, Family...). This allows the user to rapidly include all contacts into a category, which is an important requirement for the system to work. Screenshots of the current implementation are shown in Figure 2.



**Fig. 2.** Definition of reachability rules and assigning a category to an incoming caller

Our partial feedback resulting from experiences made on 5 persons during different time periods has allowed us to validate the fact that social network categories play an extremely important role in such preferences. Typically, in

the case of a professional meeting almost every person using the application blocked phone calls from friends. Family seems to be the category that is always allowed to interrupt. Another observation is the fact that preferences change very dynamically. Initially we thought that preferences change only for general situations (like work or home) and this seems to be wrong. A preference changes even when the user moves from one place to another, like from the office to the coffee machine. Currently this can be detected only when the user adds into the calendar this event, which is not very frequent. We are currently investigating how to detect such minor situations. Another major issue is the privacy concern, since the context of the users is logged. An efficient mechanism is needed to allow the algorithm to be incremental and thus to erase systematically the gathered context logs.

## 4 Conclusion and Future Work

We address in this paper the question of how to build a user profile model that can help better control our reachability in the context of an increasing exposure to unexpected interactions (spam messages, incoming calls that are not important in a situation etc.). We present a dynamic, situation-aware user model framework that enables real-time situation detection and a better control on the user's reachability. We propose mechanisms (extraction, classification algorithms, user feedback and services) that allow the model to adapt to the user. Future work include a real evaluation for the current implementation and improvement of the data mining algorithm. Also, gateways to social networking sites are considered to better automatize the categorization of contacts, leveraging that specific sites consider different spheres of the social network and that these systems are open for this kind of information retrieval.

## References

1. R. Harr, V. Kaptelinin. 2007. Unpacking the social dimension of external interruptions, GROUP '07: Proceedings of the 2007 international ACM conference on Conference on supporting group work, ACM Press, 2007, pp. 399-408.
2. M. Wiberg, S. Whittaker. 2005. Managing Availability: Supporting Lightweight Negotiations to Handle Interruptions. ACM T. HCI. 12, 4 (Dec. 2005), 365-387.
3. A. Toninelli, D. Khushraj, O. Lassila, R. Montanari. 2008. Towards Socially Aware Mobile Phones. 7th International Semantic Web Conference (ISWC2008).
4. D. Heckmann, B. Brandherm. 2005. Gumo - the general user model ontology. In Proceedings of User Modeling, Edinburgh, Scotland (2005)
5. M. Sutterer, O. Droegehorn, K. David. 2007. UPOS: User Profile Ontology with Situation-Dependent Preferences Support. Proceedings of the First International Conference on Advances in Computer-Human Interaction (2007) 230-235.
6. R. Agrawal, T. Imielinski, A. Swami. 1993. Mining Association Rules between Sets of Items in Large Databases. SIGMOD. June 1993, 22(2):207-16,
7. Khalil, A. and Connelly, K. 2006. Context-aware telephony: privacy preferences and sharing patterns. In Proceedings of the 2006 20th Anniversary Conference on

- Computer Supported Cooperative Work (Banff, Alberta, Canada, November 04 - 08, 2006). CSCW '06. ACM, New York, NY, 469-478.
8. Albrecht Schmidt, Antti Takaluoma and Jani Mntyjrv. Context-Aware Telephony over WAP. *Personal Technologies* Volume 4(4), September 2000. pp225-229.
  9. Hofte, G. H. 2007. Xensible interruptions from your mobile phone. In Proceedings of the 9th international Conference on Human Computer interaction with Mobile Devices and Services (Singapore, September 09 - 12, 2007). *MobileHCI '07*, vol. 309. ACM, New York, NY, 178-181.
  10. D. Cheng, H. Song, H. Cho, S. Jeong, S. Kalasapur, A. Messer, "Mobile Situation-Aware Task Recommendation Application," *Next Generation Mobile Applications, Services and Technologies, International Conference on*, pp. 228-233, 2008 *The Second International Conference on Next Generation Mobile Applications, Services, and Technologies*, 2008.