



HAL
open science

Collaborative Context Experience in a Phonebook

Michal Szczerbak, François Toutain, Ahmed Bouabdallah, Jean-Marie Bonnin

► **To cite this version:**

Michal Szczerbak, François Toutain, Ahmed Bouabdallah, Jean-Marie Bonnin. Collaborative Context Experience in a Phonebook. WAINA 2012: 26th International Conference on Advanced Information Networking and Applications Workshops, Mar 2012, Fukuoka, Japan. pp.1275-1281. hal-00725174

HAL Id: hal-00725174

<https://hal.science/hal-00725174>

Submitted on 24 Aug 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Collaborative Context Experience in a Phonebook

Michał Szczerbak, François Toutain

Orange Labs
France Telecom R&D
Lannion, France
{michal.szczerbak, francois.toutain}@orange.com

Ahmed Bouabdallah, Jean-Marie Bonnin

Institut Telecom
Telecom Bretagne
Cesson-Sévigné, France
{ahmed.bouabdallah, jm.bonnin}@telecom-bretagne.eu

Abstract— Starting from a context-aware phonebook application, which allows users to see their contacts’ situation, we introduce a collaborative context experience system. Users independently create notification rules, describing phonebook-based contextual situations for which they wish to receive notifications. The collaborative system gathers such rules from all users, and identifies important ones, that is, frequently declared ones. These rules are then processed to allow for generalization, and provided to users who can encounter a similar contextual situation. Similarities for this collaborative knowledge sharing mechanism are processed by applying semantic technology.

Keywords— context awareness; collaborative knowledge; semantic processing; phonebook application

I. INTRODUCTION

Context awareness is an emerging technology that enables services to become more personalized to users or for systems to become more automated. Context itself is an empowering piece of data that represents whole situations users or devices find themselves in [8]. It may concern some physical parameters, like temperature and humidity, or some higher level information, like currently performed activity. One might say that context is everything that accompanies or influences a situation. Humans use contextual reasoning upon those situations to perform their actions.

Therefore, there is a tendency to present users with contextual information in interactive services and applications so that they can use it to derive more conscious decisions. A context-aware phonebook is a very natural example for this and has become realistic thanks to availability of sensor-equipped smartphones. Its goal is to present availability or localization of contacts next to their names. In result, users are provided with additional information that is normally present in face-to-face interactions. Therefore, they may react to the provided context, for example they may want to wait until their callee’s meeting is over before contacting them, unless it is an emergency.

However, we believe there is more to empower users about than by just showing other people’s statuses and localizations, which is based on a simple data distribution mechanism. We extend this basic idea of a contextual phonebook to provide users with means to define complex situations concerning their contacts that they want to be

notified of. This enable us to introduce KRAMER (Kind of Reasoning that Abstracts Meta-situations for Empowering Recommendations), a system that enables sharing those notification rules between users.

The main motivation comes with an assumption that users are likely to define those notification triggering situations that are important for them. For example, one might want to be warned whenever his or her daughter is leaving school while his or her spouse is occupied at a meeting in work and cannot attend a child. Therefore, we design a system that shares this kind of knowledge among different anonymous users. In result, the notifications declared by ones will be received also by others if they (and their contacts) happen to find themselves in similar situations.

We have developed a mechanism to semantically process rules that are contributed to our system, in order to derive their significant abstractions. Those rules that are popular enough are then suggested to other users. A feedback between users receiving recommendations and the KRAMER system assures a constant re-evaluation of rules and promotes in a natural way those that are most important to users.

We believe that our work is the first approach of such a context-aware collaboration system, for which we provide an innovative semantic meta-situation generalization algorithm.

The rest of the paper is structured in the following manner. We give three scenarios for defining contextual rules and their further evolution in KRAMER in section 2. The state of the art of associated fields is presented in section 3. In section 4 we explain details on system mechanisms. We conclude in section 5 by giving further research axes.

II. SCENARIOS

In order for the system objectives to be clear and for the next sections to be understood more easily we present the following three scenarios. They cover several aspects of our system, which are discussed in details in section 4. Those scenarios are meant to be a reference for the next section’s state of the art evaluation with respect to what is missing today to make them possible.

EXEMPLARY DEFINITION OF A SITUATION USED IN SCENARIO 1

WHEN relation.daughter IS location.in_school AND availability.free
AND relation.wife IS availability.at_a_meeting

A. Scenario 1

I want to be notified of a situation when my daughter is leaving school and my wife is busy at a meeting. I define this contextual situation using my phonebook application. Thanks to the KRAMER system, this situation can be provided to other people. For instance, my friend receives the notification when his daughter is leaving school unattended. However, he has also a son at a similar age as our daughters so he decides to modify the rule to be more abstract and to cover children in general leaving school while one's wife is busy. In result, my boss, who has only sons, can receive the notification one day, because a concept of *child* applies also to a concept of *son*.

B. Scenario 2

After a couple of times everybody in a cinema heard my annoying ring tone for an incoming call, I decided to declare a rule concerning only me in a situation of being in a cinema to remind me to switch the ringing off. Apparently there are more day-dreamers like me out there, because similar reminders have been defined for theatres, opera houses and the rule has been automatically generalized to all "buildings providing culture and entertainment". In result, the day when hologram cinema facilities come into existence, the same notification will be given to its visitors.

C. Scenario 3

Having so little time for social meetings I decide to be notified whenever one of my friends has a "talk to me" availability status at lunch time. My wife found it a great idea and defined a similar rule for getting in touch with her female friends. In result, shortly after that, I receive a notification about my female friend who is lonely that day, as the rule became generalized to cover both genders of friends. Knowing how badly seen the idea of having lunch alone with another woman would be, I select for that rule to be rejected forever. In result, I keep the rule applying to male friends but not the generalized one.

III. BACKGROUND

Context awareness is a domain that is experiencing more and more attention in the new systems and services that are being designed. Telecom operators are equally interested in contextualizing their communication services. In [8] we envision context to make interpersonal communication more personalized, intuitive and user-empowering. Analysis of different services already enriched with information regarding user situation shows that, in spite of several important problems, the technology is on its way to introduce a whole new class of innovative services built around context. In this paper we focus on a contextual situation recommendation mechanism based on context-rich phonebooks and a mobile context distribution system, like in [6].

The domain of recommender systems is already benefiting from the power lying in contextual user data. Munoz *et al.* convince to employ space and time dimensions

to make recommendations in the Internet of Things environment [4]. They conclude that their method improves finding similar users and they describe the system to be collaborative. We can find another idea to combine context-awareness and recommendations in [2]. Here, the collaborative filtering techniques are enhanced with finding a similarity between situations a user is in. For her system, the author declares contexts to be similar for a given user, locally.

In contrast, we think of finding context patterns and exploit similarities globally. One of the recent works on reasoning about context and finding complex events is the ETALIS system [7]. The event patterns are built from a set of events satisfying both temporal and semantic relationships. ETALIS itself is based on static rules defined previously by a user. We would like, however, the system to learn about frequent complex events happening at run time, to derive the collaborative information, which can then be presented to users.

Such a system could be given a context taken from a group of users to perform its semantic generalization in order to derive frequent patterns [1]. It becomes possible to reason on abstract concepts should the semantic similarity measures be applied. For the complex situations, that we can call "meta-situations", it should be rather a method of comparing graphs of concepts like in [9]. However, Zhong *et al.* do not employ their algorithm for finding frequent generalized graphs. In general, we did not find any work concerning the problematic of finding contextual situations and exploiting their similarities.

IV. KRAMER

Our main goal in KRAMER is to prove that collaborative knowledge and experience is applicable to the context-rich situations domain. We are doing it on an example of contextualized phonebooks enabled to define and share meta-situations in a semantic form. This will lead to empowering users by a new type of information, and therefore possibly to motivating their anonymous cooperation.

A. Architecture overview

The solution discussed in this paper is based on a client-server architecture. We consider users to be equipped with a special contextual phonebook application on their mobile phones, which is a bridge connecting all of them with the core KRAMER system. We named this phonebook application COSMO (COLlaborative Situation MOdule). It is similar to the phonebook in [6] in a sense that next to the contact's name there are several pieces of information related to that contact's context. These could be, for example, location, availability, activity, mood, and application used. All of which is shared between contacts by a context distribution system. Every contact is further annotated with a concept relating him or her to the user (mother, son, friend, etc.)

Furthermore, COSMO enables declaring which context dimensions for which set of contacts should be stored in a local database. Those meta-situations are processed by rules triggering a notification each time the meta-situation reoccurs. However, COSMO does not necessarily have to relate to contact's id, name or number in order to constitute a rule regarding that contact. Instead, the data record may be provided with a semantic concept describing the contact's relationship with the user. Context model is addressed in subsection B of this section.

Even though users are defining those rules for personal use, this semantic representation of meta-situations makes them ready to be shared via the KRAMER system.

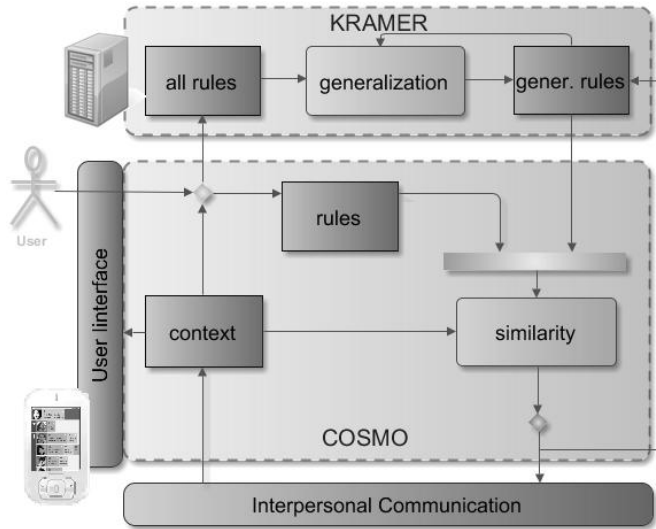


Figure 1. COSMO / KRAMER architecture

The main KRAMER server process is the generalization of all the atomic rules contributed by all users. In result, similar semantically described meta-situations are grouped together in order to select those that are most frequent. The more frequent a rule is found, the more probable is it being recommended to other KRAMER users. Furthermore, similar meta-situations, as determined by KRAMER, become more abstract thanks to a second process. This algorithm is addressed in subsection D of this section.

After on, both types of rules (those defined locally and those being suggested by KRAMER) are being compared to current situation in the same way. The condition is that a user could be interested by such meta-situations, that is: he has a set of contacts in a phonebook that matches concepts from those meta-situations. Once notified of a situation that other users have found important, a user may decide whether it is a rule he or she wants to accept to be stored locally or not. This choice influences this particular rule reputation. These mechanisms are addressed in subsection C of this section.

As it can be seen on the schematic Figure 1, there are two main functionalities that KRAMER introduces and that

we focus on: rules generalization and notifications triggered by a similar context. Both of those modules are represented with a lighter background fill. The darker background filled rectangles designate databases used in the system.

We argue that this simple mechanism provides a great value of cooperation with a positive stimulation towards system's auto-improvement, because users are likely to commit useful applications if they are the ones using them in the first place. Furthermore, we employ experiences of Roberts *et al.* [5] with their mobile recommendation system, and we design computationally complex tasks, *i.e.* generalization process, to be performed on a server. Computational complexity is addressed in subsection E of this section.

B. Context model

In order to provide real time notifications based on our current context and the situations patterns defined previously, we employ semantic technology. We believe that similarities between meta-situations can be exploited in taxonomies of already semantically rich concepts. Therefore, in KRAMER we perform their semantic generalization. We define meta-situations as a set of contacts and their context. We represent them in conceptual graphs. Each graph has a root node related to the user with edges to relationship concepts representing significant contacts. These nodes have further edges to nodes describing location and availability.

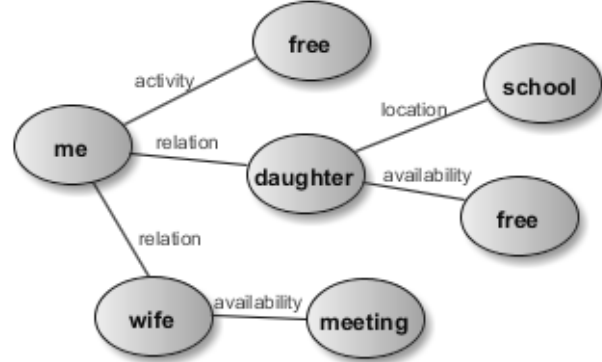


Figure 2. Conceptual graph for a meta-situation

Looking at scenario 1 from section II (represented in graph in Figure 2), the meta-situation is *daughter is leaving school while wife is occupied*. One might imagine that the user has selected, in the COSMO application, the contact fields of two very precise women: his wife and his daughter. However, storing only the concept of a *wife*, the concept of being *occupied at a meeting*, the concept of a *daughter* and the concept of a *school* define very well the meta-situation, and its actors, that he would be notified of. Even if the user has several daughters, the rule is likely to concern only those that go to school, which is even better.

All of the context dimensions are represented by taxonomies defining *is-a* relationships between concepts. This enables finding similar situations and deriving their

abstract forms. This fact is exploited in KRAMER’s generalization process in order to popularize similar rules, concerning in scenario 1, for example, all school-age children. In our prototype, we use exemplary taxonomies created just for testing purposes.

C. Suggestion mechanism

As stated before, users store their context rules locally but they should be able to learn new rules from each other. Users receive suggestions on important situations via their COSMO module (phonebook application). COSMO modules are operating separately from the server of the KRAMER system, in a mobile environment. Therefore, keeping this part of KRAMER as light (in computational terms) as possible is necessary. Once conditions triggering a given notification are met by a current situation, the user is notified and presented with details of that particular meta-situation (*e.g.* which friend contact is seeking companionship for the scenario 3 in section II).

Moreover, every notification suggested by KRAMER comes with a possibility of storing the rule locally or marking it as undesirable, according to one’s preferences. Furthermore, the system enables to delegate the notification to another contact. For example, one may imagine that a user in scenario 1 from section II may be occupied and unable to attend his daughter as well as his wife. However, given the contextual phonebook like COSMO, that user may notice that his mother is unoccupied and is located near his daughter’s school. The system enables him to delegate the whole meta-situation for her to take her granddaughter safely home.

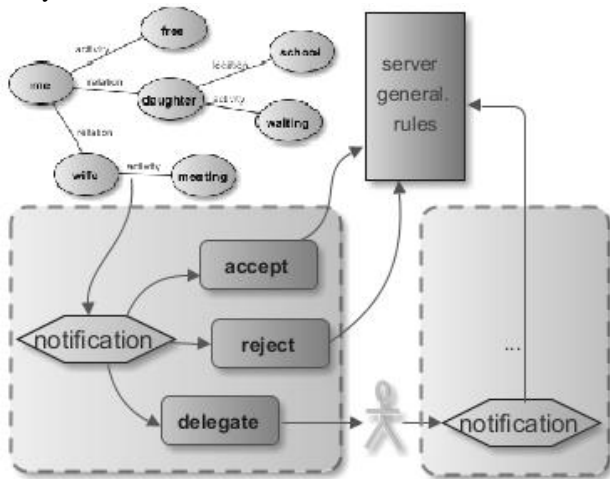


Figure 3. Suggestion process diagram

Not only that accepting, rejecting or delegating a notification provides different functionality related to user’s preferences. This decision helps, in addition, to evaluate KRAMER’s suggestions. Every meta-situation generalized on server is attributed a frequency rating. This rating *FR* measures the usefulness of a corresponding rule to users by its popularity. Its calculation is based in the beginning on a number of atomic meta-situations taken into the generalized

form. Later, it is constantly re-evaluated as a function of a feedback from users who received a suggested notification. In result, the probability of receiving any particular notification from the system is becoming smaller with descending frequency rating. Taking *N* as number of all rules in the system, w_x – weights, K_x – number of decisions: *a* – accept, *r* – reject, *d* – delegate, we derive a generic formula

$$FR = \frac{w_a K_a - w_r K_r + w_d K_d}{N} \quad (1)$$

Formula (1) requires assigning proper weights to calibrate the calculation of different decision types. One should notice that the generalization process itself also contributes to evaluating frequency of a meta-situation, for it counts atomic rules as accepted ones. In result, meta-situations are naturally becoming more or less popular in time with respect to the number of acceptances and rejections. Rules for the notifications are found frequent enough to be provided to users when they pass a defined threshold. The threshold may change in time in function of globally changing frequency ratings, which are limited by the *N* factor in the denominator.

Notifications suggested by KRAMER come as a product of a generalization process and therefore associated meta-situations may contain concepts on different levels of abstraction, *e.g.* *daughter*, *child*, *family member*, *etc.* They apply opportunistically to contacts’ concepts they find in a particular phonebook, *e.g.* *child* means both *daughter* and *son*. If a notification being suggested is accepted, it is stored in that abstract form. Users are also welcome to define their own rules on any level of abstraction as they are to refine them later in order to make them more or less specific.

All of the meta-situations (both self-defined and received from KRAMER) need to be compared in real time with constantly changing contextual situations in a contacts list. Keeping in mind that some notifications may be needed to be presented instantly, we design this process to be of minimal computational complexity. We are currently implementing the Rete algorithm [3] to enable real time situations matching. Rete network can be recomputed at runtime, which is important due to the rule-set that will be constantly changing. Nodes of that network are stimulated by a context change and notifications are triggered once the signal reaches a terminal node. For further details refer to Forgy [3].

D. Generalization algorithm

The main goal of KRAMER is to enable cooperative definition of important interpersonal meta-situations. Our recommender system suggests for a user to be notified of an important situation. The degree of this importance is evaluated by the frequency rating formula (1), which measures the popularity of a particular rule. We find, however, that such rules are very likely to be numerous

while in a lot of cases only slightly different. One may define a rule like in Figure 2, and another may just replace a *daughter* with a *son* concept (see Scenario 1 in section II). Such two rules are alike in human interpretation of a school-age child, whereas they would fight against each other as far as a frequency rating is concerned. Meanwhile, rules become more frequent should their semantic generalization be performed [1]. KRAMER helps finding abstractions of similar meta-situations.

There are two problems regarding semantic generalization. Firstly, the rules derived should not be too general to provide a useful functionality. Secondly, automated generalization is very sensitive to the number of elements in a rule. A meta-situation from Scenario 1 in section II with *a wife unable to take her daughter home at the moment* could be very well extended with an *occupied grandmother*. In either case, with or without the context of a *grandmother*, the meaning remains that the girl is left alone after classes and a user needs to react accordingly. However, if we eliminate the *daughter* element, the situation changes completely. Therefore, in order for two situations to be even considered similar the number of elements is required to be the same. In result, concept graph structures (Figure 2) need to be also exactly the same.

Therefore, the KRAMER generalization mechanism starts by grouping all those contributed meta-situations, whose conceptual graphs match. For this purpose we apply the idea of Zhong *et al.* [9]. Having context concepts defined as graph nodes and edges as context dimensions (Figure 2) the algorithm needs to check if graphs have the same edges structure. We take the formula from [9], which recursively measures the similarity of two graphs by employing distances between concepts and their relations, and we adapt it. We define S to be a set of sub-graphs obtained from eliminating a root node from a current level of recursion, n and e – nodes and edges of two compared graphs. Let n_1^s (resp. n_2^s) be the root node of the graph obtained by cutting the edge e_1^s (resp. e_2^s) from the graph which root node is n_1 (resp. n_2).

$$SoG(n_1, n_2) = \frac{1}{|S|+1} sim_n(n_1, n_2) + \max \left\{ \sum_{s \in S} \frac{1}{|S|+1} sim_e(e_1^s, e_2^s) \cdot SoG(n_1^{e_1^s}, n_2^{e_2^s}) \right\} \quad (2)$$

For every recursion the best sub-graphs' match (the max function of all possible sub-graphs combinations) is selected and normalized to the interval $\langle 0; 1 \rangle$. Edges are found either the same, sim_e function taking "1", or different, value "0". Two graphs are considered as a match if the SoG function of their root nodes has value "1". Therefore, the result of sim_n function is always "1" here. The actual concept similarity is taken into account but for choosing best sub-graphs match. The formula (2) calculates whether two

graphs are similar or not. It is applied in KRAMER to every new rule being grouped and single representatives of every already existing group, since all group members have *a priori* the same structure. Once a match is found, the new meta-situation joins the group. If all results are lower than "1" it means a mismatch and a need to create a new group for that particular meta-situation.

Grouped situations then become a subject of generalization itself. We introduce an innovative mechanism to distinguish further those meta-situations that are really similar and to get their instant abstractions. The algorithm is run for every group individually and processes all situations at once. It starts by constructing a graph with the same structure that those situations have. Each node is represented, however, as a taxonomy of concepts of a given type (taxonomy of relationships if the edge leads to the relationship type, *etc.*). Then every situation in a group marks its corresponding concept on this shared taxonomy-based tree, which we call "meta-tree".

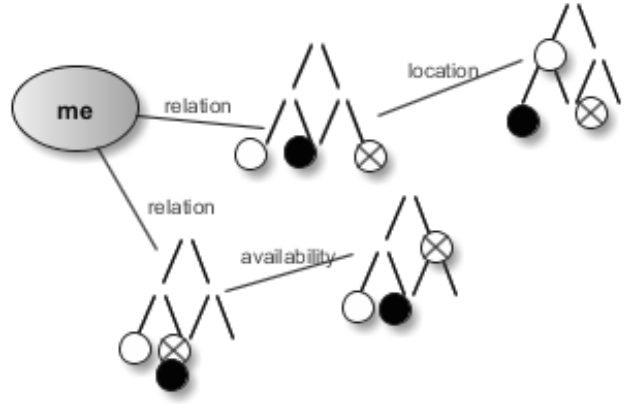


Figure 4. Meta-tree representing a group of meta-situations to be generalized

In Figure 4 three sample meta-situations (white, black and cross) marks their situation concepts on a common meta-tree structure. For further clarity, Table I presents exemplary data matching this figure.

TABLE I. SAMPLE META-SITUATIONS BEFORE GENERALIZATION

Figure 1 data	1 st contact		2 nd contact	
	relationship	location	relationship	activity
1. white	daughter	school	wife	meeting
2. black	son	school cafeteria	husband	performing surgery
3. cross	father	shop	husband	free

One may notice that situations denoted as black and white have their concepts often near on the taxonomies. And indeed those meta-situations feel similar if respective concepts in Table I are being compared. In order to promote this fact, the mechanism needs to separate those concepts on every taxonomy tree that are not close. Therefore, our generalization algorithm proceeds to successive cutting of

those branches of all trees that lead to empty (without any situation mark) leaf nodes.

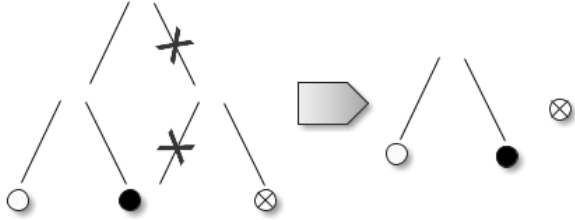


Figure 5. Empty branches cutting for meta-situations generalization

Taking the first relation concept tree from Figure 4, the algorithm finds a missing leaf concept and separates the cross mark from the black and white ones. In result, the whole cross denoted situation is found different from two others. Therefore, in our example the algorithm finds concepts of *daughter* and *son* to be similar, and different from the concept of a *father*.

Moreover, whenever a situation is separated on one taxonomy tree, it gets eliminated also from the others in a meta-tree, which makes the process dependant on all taxonomies simultaneously. Returning to our example, even though meta-situations 2 and 3 both relate to a husband contact, they are found different because of his availability and the other contact type differences.

This way we eliminate using similarity measures on concepts and declaring an *a priori* threshold for the formula (2). Instead, we are able to obtain the meta-situations generalized opportunistically by grouping graphs whose concepts have been found in common uncut sub-trees for all taxonomy trees. In result, concepts remaining in common sub-trees after the cutting process is finished become generalized into the lowest common ancestor. The simultaneous taxonomies cutting has also the benefit of limiting unnecessary abstracting the meta-situation. For the sample data in Table I and Figure 5, *son* and *daughter* concepts are generalized into a *child* concept, while *father* concept remains unchanged.

Admitting that white and black marks remain together on all the uncut trees for Figure 4, the corresponding two situations will be found similar, and all concepts shall become generalized. Therefore, there shall be two rules expected as an output, the cross one, and the generalized white and black one. Sample meta-situations from Table I becomes generalized by KRAMER in a form presented in Table II.

For the purpose of the frequency rating (1) the number of contributing atomic meta-situations is associated with each result. One should notice that, during the lifetime of the KRAMER system, many new users may contribute to the already existing generalized rules. For sake of computing time, the generalization mechanism processes both newly uploaded rules and those already generalized (loopback arrow in Figure 1). This is naturally supported by the algorithm and enables memorizing the frequency of rule usage from the very beginning for every rule.

TABLE II. SAMPLE META-SITUATIONS AFTER GENERALIZATION

Figure 1 data	1 st contact		2 nd contact	
	relationship	location	relationship	activity
1. white & 2. black	child	school	spouse	occupied
2. cross	father	shop	husband	free

E. Computational complexity

The worry of all comparison and pattern mining algorithms is their exponential computational complexity. Zhong *et al.* show, however, that the complexity of their method based on Bellman-Ford algorithm is polynomial. Our own algorithm is expected to be at most $O(G^2AA!)$, where G is the number of graphs to be matched, each containing approximately A edges. The specifics of the input data for KRAMER would suggest A to be rather small, fixed at 3-4 edges connected with “me” and 1-2 edges coming out from those actors as users are unlikely to define very complicated rules. Therefore, it is G that should dominate the complexity $O(G^2)$.

We have implemented the generalization module of the KRAMER in Java programming language. We have used a personal computer with two-core CPU, 2.80GHz each, 6GB RAM and 64-bit operating system. The test run on randomly generated rule graphs containing several nodes confirms the estimated computational complexity and shows that, in this environment, 5000 rules are processed to the generalized form in less than 1,5s.

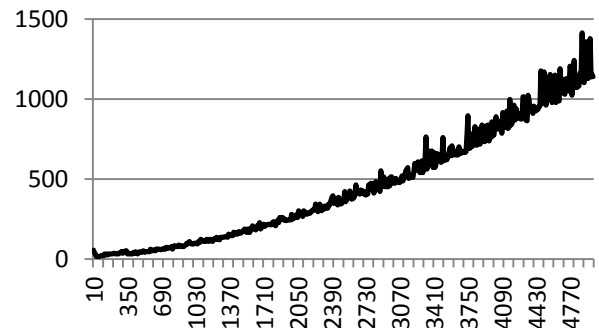


Figure 6. Time in [ms] to generalize sets of rules in KRAMER

The semantic comparison module is being currently implemented at Orange Labs and is designed to use a Rete network. Every change in the context environment shall propagate in the network in order to activate notifications upon satisfying all rule conditions. The computational complexity of this process is exactly $O(1)$. It is the network refining that consumes more resources. We imagine, however, processing big sets of generalized rules can be delegated to a server or local PC, depending on the type of connection between a COSMO module and the KRAMER server.

V. CONCLUSIONS

This paper introduces the KRAMER system designed to empower users with a collaborative knowledge derived from experiences and preferences of other KRAMER users. We apply it to notifications for certain context-rich phonebook contacts situations. Our work is focused on exploiting the semantic similarity of such complex situations, which we call “meta-situations”. For this reason, we introduce an algorithm finding groups of similar graphs representing people relationships, their localization and availability semantics. The prototype of this system is developed in Orange Labs. Still, there are several issues to be addressed in our research.

We shall most likely extend the KRAMER system with a notion of temporal relations between statuses and events. Frequently, it is important that one situation precedes another and we might find it important to add to the system. In result, there could be a need of refining the data representation or the comparing algorithm itself.

One of the open questions is whether KRAMER would need additional personalization mechanisms. For instance, different nationalities or genders could determine different sets of rules to be treated as important. For the time being, we consider the very fact of facing a similar situation as the one in a rule to be distinguishing types of users well enough (a user who has phonebook contacts of his wife and his child is similar to other family-men in a sense that he could be presented with rules using concepts of a *wife* and a *child*).

Furthermore, we design the system to be generic. This means that by changing peripheral blocks we shall maintain the semantic context generalization operational. In particular, we consider changing limited notifications into rich action suggestions. Such actions could be for example related to the telecommunication application domain, *e.g.* call management or supplementary services activation. Furthermore, we think that the very same core can be applied to machine-to-machine environments, where instead of statuses of contacts there would be states of ubiquitous nodes. The collaborative knowledge would be related to frequently used pervasive applications.

We are also considering enabling KRAMER to be provided with users’ frequent behaviour patterns mined from the raw context obtained from sensors. This approach might give a wider coverage of situations and actions the system would be recommending. Some associations and dependencies could be mined automatically even without any need of explicit data entry from users, like meta-situations defining in COSMO. However, for those types of data-mining mechanisms, huge databases of many different context dimensions should be provided for the system to work properly. The optimal pattern searching algorithms have exponential computation complexity and we cannot compromise a real-time system response to the changing situation. We fear that the system in such a form could be too intrusive and too chaotic in its recommendations.

Another concern could be related to privacy. We believe that our rules sharing mechanism is well designed in this respect, since the rule expression is anonymous. The identity of a user contributing a rule, as well as his or her contacts’ identities, are thus undisclosed. It remains to study how well this protection holds in case of an attack on the KRAMER system.

Finally, it is the users that we wish to empower in their everyday life and therefore, tests will be necessary for a feedback concerning the user experience with the system. Those tests are also important to evaluate the accuracy of the recommendations given by KRAMER. This parameter can be found mathematically by a metric that will depend on the number of both suggestions accepted and rejected by users. Even though the system should very well cope with starting from an empty rule database, we consider pre-filling it and giving users some scenario behaviours in order to observe more condensed results.

ACKNOWLEDGMENTS

The KRAMER system is being designed as a PhD thesis. It is co-funded by the CIFRE program, under subvention of the French government, as cooperation between Telecom Bretagne University and Orange Labs in France.

REFERENCES

- [1] E. Baralis, L. Cagliero, T. Cerquitelli, P. Garza, and M. Marchetti: Context-Aware User and Service Profiling by Means of Generalized Association Rules, *13th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 2009
- [2] A. Chen: Context-Aware Collaborative Filtering System: Predicting the User’s Preference in the Ubiquitous Computing Environment, *Location and Context-Awareness Lecture*, 2005
- [3] C.L. Forgy: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, *Artificial Intelligence*, vol. 19, pp. 17-37, 1982
- [4] M. Muñoz-Organero, G. A. Ramírez-González, P. J. Muñoz-Merino, and C. D. Kloos: A Collaborative Recommender System Based on Space-Time Similarities, *IEEE Pervasive Computing Magazine*, Vol. 9(3), pp 81-87, 2010
- [5] M. Roberts, N. Ducheneaut, B. Begole, K. Partridge, B. Price, V. Bellotti, A. Walendowski, and P. Rasmussen: Scalable Architecture for Context-Aware Activity-Detecting Mobile Recommendation Systems, *ADAMUS Workshop at IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, 2008
- [6] A. Schmidt, T. Stuhr, and H. Gellersen: Context-Phonebook – Extending Mobile Phone Applications with Context, *Third International Workshop on Human Computer Interaction with Mobile Devices*, 2001
- [7] N. Stojanovic, L. Stojanovic, D. Anicic, J. Ma, S. Sen, and R. Stuhmer: Semantic Complex Event Reasoning – Beyond Complex Event Processing, *Foundations for the Web of Information and Services*, Springer-Verlag Berlin Heidelberg, pp 253-279, 2011
- [8] F. Toutain, A. Bouabdallah, R. Zemek, and C. Daloz: Interpersonal Context-Aware Communication Services, *IEEE Communications Magazine*, Vol. 49(1), pp 68-74, 2011
- [9] J. Zhong, H. Zhu, J. Li, and Y. Yu: Conceptual Graph Matching for Semantic Search, *10th International Conference on Conceptual Structures: Integration and Interfaces*, 2002