The *IONWI* Algorithm: Learning when and when not to interrupt

Silvia Schiaffino and Analía Amandi
ISISTAN Research Institute – Facultad de Cs. Exactas – UNCPBA –
Campus Universitario, Paraje Arroyo Seco, Tandil, 7000, Bs As, Argentina
Also CONICET, Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina
{sschia,amandi}@exa.unicen.edu.ar

Abstract. One of the key issues for an interface agent to succeed at assisting a user is learning when and when not to interrupt him to provide him assistance. Unwanted or irrelevant interruptions hinder the user's work and make him dislike the agent because it is being intrusive and impolite. The *IONWI* algorithm enables interface agents to learn a user's preferences and priorities regarding interruptions. The resulting user profile is then used by the agent to personalize the modality of the assistance, that is, assisting the user with an interruption or without an interruption depending on the user's context. Experiments were conducted in the calendar management domain, obtaining promising results.

Keywords: intelligent agents, user profiling, human-computer interaction

1. Introduction

As intelligent agents take on more complexity, higher degrees of autonomy and more "intelligence", users start to expect them to play by the same rules of other complex, autonomous and intelligent entities in their experience, namely, other humans [16]. Our previous studies [19] demonstrated that the way in which an interface agent assists a user has an impact on the competence of this agent and it can make the interaction between user and agent a success or a failure. This is the concern of a recent research area within Human-Computer Interaction (HCI) that studies the “etiquette” of human-computer relationships [3; 17; 18]. We agree with the researchers in this area on that the ability to adapt to the way in which a user wants to interact with the agent is almost as important as the ability to learn the user's preferences in a particular domain.

As pointed out in [14], one of the problems with the interface agents developed thus far is their incorrect estimates of the user’s task priorities, which makes information to be introduced at inappropriate times and with unsuitable presentation choices. Although agents are well-intentioned, they do not consider the impact an interruption has on the user. Research has found that interruptions are harmful. They are disruptive to the primary computing task and they decrease users’ performance. However, interruptions are necessary in interface agent technology since agents need to communicate important and urgent information to users.

To solve this problem, when the agent detects a (problem) situation relevant to the user it has to correctly decide if it will send him a notification without interrupting the user’s work, or if it will interrupt him. On the one hand, the user can choose between paying attention to a notification or not, and he can continue to work in the latter case. On the other hand, he is forced to pay attention to what the agent wants to tell him if it interrupts him abruptly.

Not to disturb the user, the agent has to base its decision on various factors, such as: the relevance and the urgency the situation has for the user; the relationship between the situation to be notified or the assistance to be provided and the user's goals; the relevance the situation underlying the interruption has to the current user tasks; how tolerant the user is of interruptions; and when he does not want to be interrupted no matter how important the message is. In summary, the interface agent has to learn which situations are relevant and which are irrelevant so that no unwanted interruptions occur.

In this work we present a user profiling algorithm named *IONWI* that learns when a user can or should be interrupted by his agent depending on the user’s context. In this way, the agent can provide personalized assistance to the user without hindering his work.

This article is organized as follows. Section 2 presents our proposed profiling algorithm. Section 3 shows the results we have obtained when assisting users of a calendar management system. Section 4 describes some related works. Finally, Section 5 presents our conclusions and future work.

2. The *IONWI* Algorithm

In order to assist a user without hindering his work, an interface agent has to learn the user's interruption needs and preferences in different contexts. In this work we propose an algorithm, named *IONWI* (acronym for Interruption Or Notification Without Interruption), capable of learning when to interrupt a user and when not from the observation of the user's interaction with a computer application and with the agent.

The algorithm learns when a situation that may originate an interruption is relevant to the user's needs, preferences and goals, and when it is irrelevant. In addition, this algorithm also considers the relationship and relevance the situation originating the interaction has with the user's current task.

2.1 Algorithm inputs and outputs

The input for our learning algorithm is a set of user-agent interaction experiences. An interaction experience *Ex* is described by seven arguments <*Sit*, *Mod*, *Task*, *Rel*, *UF*, *E*, *date*>: a problem situation or situation of interest *Sit* is described by a set of features and the values these features take, $Sit=\{(feature_i,value_i)\}$; the modality *Mod* that indicates whether the agent interrupted the user or not to provide him assistance; the *Task* the user was executing when he was interrupted or notified, which is described by a set of features and the values these features take $Task=\{(feature_i,value_i)\}$; the relevance *Rel* the interruption has for the *Task*; the user feedback *UF* (regarding the assistance modality) obtained after assisting the user; an evaluation *E* of the assistance experience (success, failure or undefined); and the *date* when the interaction experience was recorded.

For example, consider that the user is scheduling a meeting with several participants and he is interrupted by his agent to remind him about a business meeting that will take place the next day. The user does not pay attention to the message being notified and presses a button to tell the agent not to interrupt him in these occasions. From this experience the agent learns that reminders of this kind of meetings are not relevant to the user, and it will send him a notification in the future without interrupting him. In this example, the different components of the assistance experience are:

Sit ={(type, event reminder), (event-type, business meeting), (organizer, boss), (participants, [Johnson, Taylor ,Dean]), (topic, project A evolution), (date, Friday), (time, 5p.m.), (place, user's office)}

Mod = interruption

Task = {(application, calendar management system),(task, new event), (event type, meeting), (priority, high), ...}

Rel = irrelevant, unrelated

UF = {(type, explicit), (action, do not interrupt)}

E = {(type, failure), (certainty, 1.00)} (interruption instead of notification)

Date = {(day, 18), (month, December), (year, 2005)}

The output of our algorithm is a set of facts representing the user's interruptions preferences. Each fact indicates whether the user needs an interruption or a notification when a given situation occurs in the system. Facts constitute part of the user profile. These facts may adopt one of the following forms: “in problem situation *Sit* the user should be interrupted”, “in situation *Sit* the user should not be interrupted”, “in situation *Sit* and if the user is performing the task *T*, he should not be interrupted”, “in situation *Sit* and if the user is performing the task *T*, the agent can interrupt him”. Each fact *F* is accompanied by a certainty degree *Cer(F)* that indicates how certain the agent is about this preference. Thus, when an interface agent has to decide whether to interrupt the user or not given a certain problem situation, the agent uses the knowledge it has acquired about a user's interruption preferences to choose the assistance modality it supposes the user expects in that particular instance of a given situation. Once the assistance has been provided, the agent obtains explicit and/or implicit user feedback. This new interaction is recorded as an assistance experience, which will be used in the future to incrementally update the knowledge the agent has about the user.

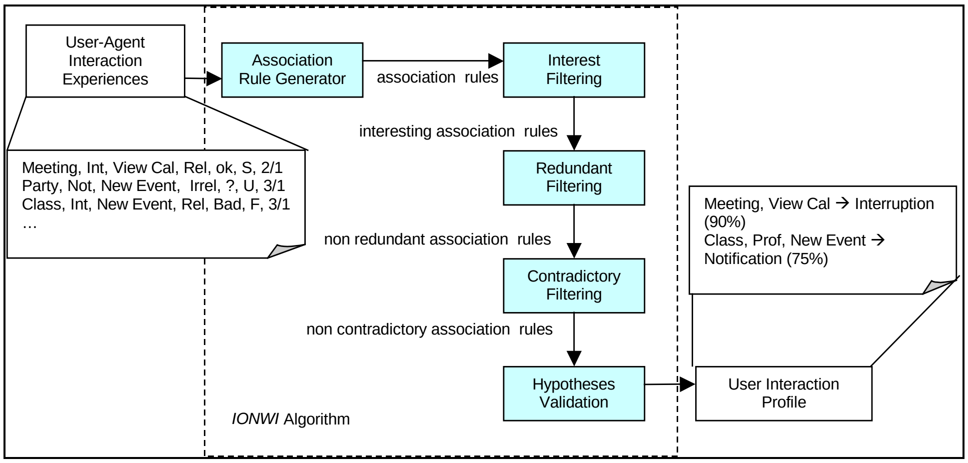

Fig. 1. ***IONWI*** Overview

2.2 *IONWI* Overview

The *IONWI* algorithm uses association rules to obtain the existing relationships between situations, current user tasks and assistance modalities. Classification techniques have been discarded since we cannot always label an interaction as a success or a failure, and we need a group of interactions to draw a conclusion about the user's preferences.

As shown in Figure 1, the first step of our algorithm is generating a set of association rules from the user-agent interaction experiences. Then, the association rules generated are automatically post-processed in order to derive the user profile from them. Post-processing steps include: detecting the most interesting rules according to our goals, eliminating redundant and insignificant rules, pruning out contradictory weak rules, and summarizing the information in order to formulate the hypotheses about a user's preferences more easily. Once a hypothesis is formulated, the algorithm looks for positive evidence supporting the hypothesis and negative evidence rejecting it in order to validate it. The certainty degree of the hypothesis is computed taking into account both the positive and the negative evidence. This calculus is done by using metrics from association rule discovery. Finally, facts are generated from the set of highly supported hypotheses; facts compose the user interaction profile.

The following subsections describe in detail each step of the algorithm.

2.3 Mining Association Rules from User-Agent Interaction Experiences

An association rule is a rule that implies certain association relationship among a set of objects in a database, such as occur together or one implies the other. Association discovery finds rules about items that appear together in an event (called transactions), such as a purchase transaction or a user-agent interaction experience. Association rule mining is commonly stated as follows [1]: Let $I=\{i_1,...,i_n\}$ be a set of items, and D be a set of data cases. Each data case consists of a subset of items in I. An association rule is an implication of the form $X\rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X\cap Y=\varnothing$. X is the antecedent of the rule and Y is the consequent. The support of a rule $X\rightarrow Y$ is the probability of attribute sets X and Y occurring together in the same transaction. The rule has support *s* in *D* if *s*% of the data case in *D* contains $X \cap Y$. If there are *n* total transactions in the database, and X and Y occur together in *m* of them, then the support of the rule $X\rightarrow Y$ is *m/n*. The rule $X\rightarrow Y$ holds in *D* with confidence *c* if *c*% of data cases in *D* that contain X also contain Y. The confidence of rule $X\rightarrow Y$ is defined as the probability of occurrence of X and Y together in all transactions in which X already occurs. If there are *s* transactions in which X occurs, and in exactly *t* of them X and Y occur together, the confidence of the rule is *t/s*.

Given a transaction database *D*, the problem of mining association rules is to find all association rules that satisfy: minimum support (called *minsup*) and minimum confidence (called *minconf*). There has been a lot of research in the area of association rules and, as a result, there are various algorithms to discover association rules in a database. The most popular is the Apriori algorithm [1], which is the one we use to find our association rules.

2.4 Filtering Out Uninteresting and Redundant Rules

In this work, we are interested in those association rules of the form "situation, modality, task → user feedback, evaluation"; "situation, modality → user feedback, evaluation"; "situation, modality, relevance → user feedback, evaluation" and "situation, modality, task, relevance → user feedback, evaluation", having appropriate support and confidence values. We are interested in these rules since they provide us information about the relationships between a situation or problem description and the modality of assistance the user prefers, which have received a positive (negative) evaluation. They also relate a situation and the current user task with an assistance modality, as well as a situation, the current user task and the relevance of the situation to the task with a certain assistance modality. To select these types of rules, we define templates [10] and we insert these templates as restrictions in the association mining algorithm. Thus, only interesting rules are generated (steps 1 and 2 in Figure 1 are then merged).

Once we have filtered out those rules that are not interesting for us, we will still have many rules to process, some of them redundant or insignificant. Many discovered associations are redundant or minor variations of others. Thus, those spurious and insignificant rules should be removed. We can then use a technique that removes those redundant and insignificant associations [13]. For example, consider the following rules:

R1: Sit{(Type, Event Reminder)(Event-Type = doctor))} (Task=View Calendar), (Mod =interruption) → (UF = do not interrupt), (Ev = failure) [sup: 0.4, conf: 0.82]

R2: Sit{(Type, Event Reminder)(Event-Type = doctor)}, (Task=View Calendar), (Event-Priority = high)), (Mod =interruption) → (UF= do not interrupt), (Ev = failure) [sup:0.4, conf: 0.77]

If we know R1, then R2 is insignificant because it gives little extra information. Its slightly higher confidence is more likely due to chance than to true correlation. It thus should be pruned. R1 is more general and simple.

In addition, we have to analyze certain combinations of attributes in order to determine if two rules are telling us the same thing. For example, a rule containing the pair "interruption, failure" and another containing the pair "notification, success" are redundant provided that they refer to the same problem situation and they have similar confidence values. As well as analyzing redundant rules, we have to check if there are any contradictory rules. We define that two rules are contradictory if for the same situation and, eventually for the same user task, they express that the user wants both an interruption and a notification without interruption.

2.5 Building Facts from Hypotheses

The association rules that have survived the pruning processes described above are those the *IONWI* algorithm uses to build hypotheses about a user's interruption preferences. A hypothesis is obtained from a set of association rules that are related because they refer to the same problem situation but are somewhat different: a “main” association rule; some redundant association rules with regards to the main rule, which could not be pruned out because they did not fulfill the similar confidence restriction; and some contradictory rules with regards to the main rule, which could be not pruned away because they did not meet the different confidence requirement. The main rule is chosen by selecting from the rule set the rule that has the greatest support value, whose antecedent is the most general, and whose consequent is the most specific.

$$Cer(H) = \alpha Sup(AR) + \beta \frac{\sum_{k=1}^{r} Sup(E+)}{\sum_{k=1}^{r+t} Sup(E)} - \gamma \frac{\sum_{k=1}^{t} Sup(E-)}{\sum_{k=1}^{r+t} Sup(E)}$$

Equation 1

Once the *IONWI* algorithm has formulated a set of hypotheses it has to validate them. The certainty degree of a hypothesis *H* is computed as a function of the supports of the rule originating the hypothesis and the rules considered as positive and negative evidence of *H*. The function we use to compute certainty degrees is shown in Equation 1, where: α, β and γ are the weights of the terms in the equation (we use α=0.8, β=0.1 and γ=0.1), *Sup(AR)* is the support of the rule originating *H*, $Sup(E^+)$ is the support of the rules being positive evidence, $Sup(E^-)$ is the support of the rules being negative evidence, *Sup(E)* is the support value of an association rule taken as evidence (positive or negative), *r* is the amount of positive evidence and *t* is the amount of negative evidence.

2.6 Incremental Learning

The database containing interaction experiences is not static, because updates are constantly being applied to it. On the one hand, new interaction experiences are added since the agent keeps observing a user's behaviour. On the other hand, old experiences are deleted because they become obsolete. In consequence, new hypotheses about a user's interruption preferences may appear and some of the learned hypotheses may become invalid.

We address this problem from the association rule point of view, that is, as the database changes new association rules may appear and at the same time, some existing association rules may become invalid. The incremental version of *IONWI* uses the FUP2 algorithm [5] to update the association rules and the DELI algorithm [12] to determine when it is necessary to update the rules. The DELI algorithm uses a sampling technique to estimate the difference between the old and new association rules. This estimate is used as an indicator for whether the FUP2 algorithm should be applied to the database to accurately find out the new association rules. If the estimated difference is large enough (with respect to some user specified threshold), the algorithm signals the need of an update operation, which can be accomplished by using the FUP2 algorithm. If the estimated difference is small, then we do not run FUP2 immediately and we can take the old rules as an approximation of the new rules. Hence, we wait until more changes are made to the database and then re-apply the DELI algorithm.

3. Experimental Results

We tested our algorithm with a set of 26 datasets[1] containing user-agent interactions in the calendar management domain. Each database is composed of the attributes that describe the problem situation or situation of interest originating the interaction, the primary user task, the modality of the assistance, the relationship between the situation and the user task, the user feedback, and the evaluation of the interaction experience. The sizes of the datasets vary from 30 to 120 interactions.

To evaluate the performance of an agent using our learning algorithm we used one of the metrics defined in [4]. The precision metric measures an interface agent’s ability to accurately provide assistance to a user. As shown in Equation 2, we can define our precision metric as the ratio of the number of correct interruption preferences to the total number of interruption preferences generated by *IONWI*. Similarly, as shown in Equation 3, we can define the recall metric (i.e. what the agent could not learn) as the ratio of the number of correct interruption preferences to the number of preferences indicated by the user.

[1] The datasets can be found at http://www.exa.unicen.edu.ar/~sschia

Figure 2 presents the results we have obtained. The graph in Figure 2(a) plots the percentage of interruption preferences correctly identified by the *IONWI* algorithm (with respect to the total number of preferences obtained); the number of incorrect interruption preferences; and the number of "hidden" preferences, that is those preferences that were not explicitly stated by the user but are correct. Each figure shows the percentage values obtained when averaging the results we got with the different users. The graph in Figure 2(b) shows the percentage of correct interruption preferences (with respect to the number of preferences specified by the user) and the percentage of missing interruption preferences, that is those that the algorithm could not detect. Each graphic shows the average percentage values of the results obtained with the different datasets.

$$IONWI_{precision} = \frac{number\ of\ correct\ preferences}{number\ of\ preferences}$$

Equation 2

$$IONWI_{recall} = \frac{number\ of\ correct\ preferences}{number\ of\ preferences\ for\ user}$$

Equation 3

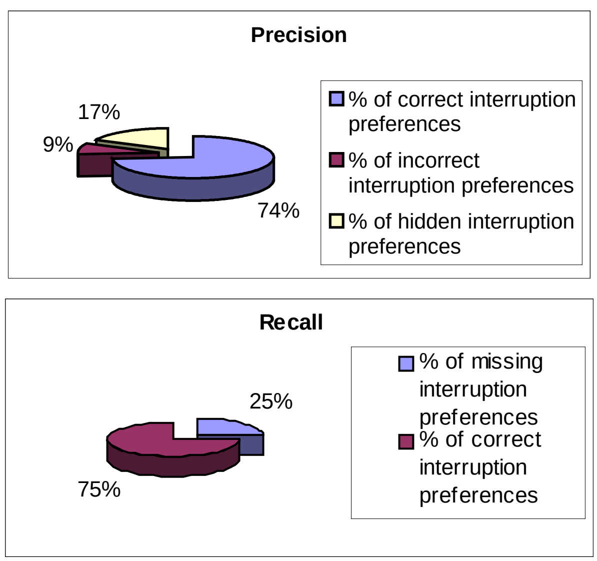

Fig. 2. *IONWI* Precision (a) and Recall (b)

We can observe in the figures that the percentage of incorrect interruption preferences is small (9% in average), and that the percentage of correct preferences plus the percentage of hidden preferences is considerably high. The percentage of correct interruption preferences plus the percentage of hidden preferences can be considered as the precision of the algorithm. This value is approximately 91%. Thus, we can state that the learning capability of the *IONWI* algorithm is good.

Regarding the algorithm recall, 25% of the interruption preferences specified by the user were not discovered by our algorithm. Although not observable in the graphic, this value was smaller for those datasets containing more than 50 records.

4. Related Work

Interruptions have been widely studied in the HCI area[2], but they have not been considered in personal agent development. These studies revealed that the disruptiveness of an interruption is related to several factors, including complexity of the primary task and/or interrupting task, similarity of the two tasks [8], whether the interruption is relevant to the primary task [6], stage of the primary task when the interruption occurs [7], management strategies for handling interruptions [15], and modalities of the primary task and the interruption [2, 11].

People at Microsoft Research have deeply studied the effects of instant messaging (IM) in users, mainly on ongoing computing tasks [6, 7, 9]. These authors found that IM that were relevant to ongoing tasks were less disruptive than those that were irrelevant. This influence of relevance was found to hold for both notifications viewing and task resumption times, suggesting that notifications that were unrelated to ongoing tasks took longer to process.

As we have already said, related studies on interruptions come from different research areas in which interface agents are not included. Nevertheless, the results of these studies can be taken into account by interface agents to provide assistance to users without affecting users' performance in a negative way and, thus, diminishing the disruptiveness of interruptions. None of the related works we have discussed has considered the relevance of interruptions to users, or the relevance the situation originating the interruption has for the user. This issue and the relevance of interruptions to user tasks are two aspects of interruptions that our learning algorithm considers.

5. Conclusions and Future Work

We have presented a profiling algorithm that learns when and when not to interrupt a user, in order to provide him assistance. We have evaluated our proposal in the calendar management domain and the results we have obtained are quite promising. Experiments with personal agents assisting users with our approach in other domains are currently being carried out.

As a future work, we are planning to enhance the representation of a user's context in order to take other aspects into account.

[2] Bibliography on this topic: http://www.interruptions.net/literature.htm

References

[1] Agrawal, R., Srikant, R. - Fast Algorithms for Mining Association Rules – In Proc. 20th Int. Conf. Very Large Data Bases (VLDB) – 487 – 499 – (1994)
[2] Arroyo, E., Selker, T, Stouffs, A. – Interruptions and multimodal outputs: Which are less disruptive? – In IEEE International Conference on Multimodal Interfaces ICMI 02 – 479 - 483 (2002)
[3] Bickmore, T. Unspoken rules of spoken interaction. Communications of the ACM, 47 (4): 38 – 44 – (2004)
[4] Brown, S. and Santos, E. - Using explicit requirements and metrics for interface agent user model correction. In Proc. 2nd International Conference on Autonomous Agents – (1998)
[5] Cheung, D., Lee, S., Kao, B. A general incremental technique for maintaining discovered association rules. In Proc.5th Int. Conf. on Database Systems for Advanced Applications – (1997).
[6] Czerwinski, M., Cutrell, E., Horvitz, E. Instant messaging and interruption: Influence of task type on performance. In Proc. OZCHI2000 – 2000.
[7] Czerwinski, M., Cutrell, E., Horvitz, E. Instant Messaging: Effects of Relevance and Timing. People and Computers XIV: Proceedings of HCI 2000 – 71 – 76 (2000)
[8] Gillie T., Broadbent, D. What Makes Interruptions Disruptive? A Study of Length, Similarity and Complexity - Psychological Research, Vol. 50, 243 – 250 (1989)
[9] Horvitz, E., Jacobs, A., Hovel, D. Attention-Sensitive Alerting - Proceedings of UAI 99, Conference on Uncertainty and Artificial Intelligence – 305 - 313 (1999)
[10] Klementinen, M., Mannila, H., Ronkainen, P., Toivonen, H., Verkamo, A. I. - Finding interesting rules from large sets of discovered association rules. In 3rd Int. Conf. on Information and Knowledge Management – (1994) 401 – 407
[11] Latorella, K. Effects of Modality on Interrupted Flight Deck Performance: Implications for Data Link - Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting (1998)
[12] Lee, S., Cheung, D. Maintenance of discovered association rules: When to update? In Proc. SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery – (1997)
[13] Liu, B., Hsu, W., Ma, Y. - Pruning and summarizing the discovered associations - In Proc. 5th ACM SIGKDD – (1999)
[14] Mc. Crickard, S., Chewar, C. Attuning notification design to user goals and attention costs. Communications of the ACM, 46 (3): 67 – 72 – (2003)
[15] Mc Farlane, D. Coordinating the interruption of people in human-computer interaction. INTERACT 99, 295 – 303 – (1999)
[16] Miller, C. Definitions and dimensions of etiquette. In Proc. AAAI Fall Symposium on Etiquette and Human-Computer Work – (2002)
[17] Miller, C. Human-computer etiquette: Managing expectations with intentional agents. Communications of the ACM, 47 (4): 31 – 34 – (2004)
[18] Nass, C. Etiquette equality: Exhibitions and expectations of computer politeness. Communications of the ACM, 47 (4): 35 – 37 – (2004)
[19] Schiaffino, S., Amandi, A. – User – Interface Agent Interaction: Personalization Issues – International Journal of Human – Computer Studies, 60 (1): 129 – 148 – (2004)

Formal Analysis of the Communication of Probabilistic Knowledge

João Carlos Gluz[1,2,3], Rosa M. Vicari[1], Cecília Flores[1], Louise Seixas[1]

1 Instituto de Informática, UFRGS,95501-970, Porto Alegre, RS, Brasil, {jcgluz,rosa,dflores}@inf.ufrgs.br, seixasl@terra.com.br

2 ESD, UERGS, Guaíba, RS, Brasil, joao-gluz@uergs.edu.br

3 FACENSA, Gravataí, RS, Brasil, jcgluz@facensa.com.br

Abstract. This paper discusses questions about communication of probabilistic knowledge in the light of current theories of agent communication. It will argue that there is a semantic gap between these theories and research areas related to probabilistic knowledge representation and communication, that creates very serious theoretical problems if agents that reason probabilistically try to use the communication framework provided by these theories. The paper proposes a new formal model, which generalizes current agent communication theories (at least the standard FIPA version of these theories) to handle probabilistic knowledge communication. We propose a new probabilistic logic as the basis for the model and new communication principles and communicative acts to support this kind of communication.

1 Introduction

This paper will present a theoretical study about which kind of meaning can be assigned to the communication of probabilistic knowledge between agents in Multiagent Systems (MAS), at least when current theories for agent communication are considered. The work starts in section 2, presenting several considerations showing that exists a semantic gap between current agent communication theories and research areas related to probabilistic knowledge representation and communication. This gap creates very serious theoretical problems if the designer of agents that reason probabilistically tries to use the communication framework provided by these theories to model and implement all agent's communication tasks.

To minimize this gap we propose a new formal model in section 3, which generalizes the formal model, used in FIPA agent communication standards [6], to handle probabilistic knowledge communication. We propose a new probabilistic logic, called *SLP*, as the basis for the new model. The *SLP* logic is compatible with the logic used as the foundation of FIPA standards (the *SL* logic) in the sense that all valid formulas (theories) of *SL* are also valid formulas of *SLP*. The axiomatic system of *SLP* is correct. It is also complete, if the axiomatic system of *SL* is complete.

Based on *SLP* logic we propose a minimum set of new communication principles in section 4 that are able to correlate probabilistic reasoning with communication related inference tasks. Two new communicative acts are proposed that would allow agents to communicate basic probabilistic propositions without having to agree previously on a probabilistic content format.

This is the most important result of the paper. To our knowledge, this is the first work that tries to integrate in a single probabilistic-logical framework two entirely different approaches to understand and model communication. What we have done, after have carefully isolated formal axiomatic agency and communication theories used by FIPA, was to define the minimum set of new axioms necessary and sufficient to support an probabilistic form of assertive and query communicative acts. We also maintain the principles, acts and axioms as simple as possible to be able to easily assess how much we were departing from classical Speech Act theory. We believe, that given the circumstances, albeit a conservative approach, this is the correct approach. The result was a clear and simple generalization of current FIPA axiomatic communication and agent theories that is able to handle basic probabilistic communication between agents.

A secondary, but interesting, result of the paper is the (relative) completeness of *SLP* logic. To our knowledge, there is no other axiomatization for an epistemic and temporal modal logic, which allow probabilities for first order modal sentences, and is proved complete.

2 Motivation

This work has started with a very practical and concrete problem, which was how to model (and implement) the communication tasks of all agents from a real MAS: the AMPLIA system [13,8]. We have decided to use only standard languages and protocols to model and implement these tasks in order to allow reusability of the agent's knowledge and to allow an easier interoperation of AMPLIA with others intelligent learning systems. To this purpose we decided to use FIPA standards based on two assumptions: (a) the standards are a good way to ensure MAS knowledge reusability and interoperability; (b) the formal basis of FIPA standards offer an abstract and architecture independent way to model all communication tasks of the system, allowing a high level description of the communication phenomena. However, we have found that it was impossible to meet even most basic communication requirements of AMPLIA using only FIPA standards. All AMPLIA's agents use and communicate probabilistic (bayesian) knowledge, but FIPA standards assigns no meaning to probabilistic knowledge representation or communication.

Of course it is possible to try to "hide" all probabilistic knowledge in a special new content format, allowing, for example, that Bayesian Networks (BN) should be "encoded" in this format and then embedded as contents of FIPA Agent Communication Language (ACL) communicative acts. The knowledge to be passed as contents of assertive acts like FIPA's **inform**, can be considered as a logical proposition that the agent believe it is true. In being so, it is possible to assume that, from a communication point of view, it is only necessary that the agent believe that the "hidden" probabilistic knowledge transported by the act be true. Any other meaning related the probabilistic knowledge do not need be "known" by the agent in respect to communication tasks or in any reasoning related to these tasks.

2.1 The Research Problem

The approach to “hide” probabilistic knowledge solves some basic implementation problems if theoretical or formal aspects of this kind of communication are not considered. However, when analyzed more carefully this approach does not seem to be very sound.

The first problem is related to the fact that formal semantics of FIPA ACL is based on axiomatic logical theories of intention and communication [4,5,11,12]. Besides particular pre and pos-conditions (expressed as logical axioms) for some act, these theories will define clearly when the act should be emitted, what are the intentions of the sender agents when it send the act, which effects this act should cause in the receiver agent and so on. The knowledge transported in these acts are only logical propositions, but these propositions are related to internal beliefs, intentions and choices of the agents and must be used in reasoning process that will decides when to emit some act or how the act received should be understood. This imply that even if you have some probabilistic knowledge "hidden" in the contents of a communicative act, then this knowledge *cannot be used* in any internal reasoning process related to communication tasks, because formal model and theories that fundament this reasoning (at least in FIPA standards) are purely logical and do not allow reasoning about probabilities. This generates a strange situation when you have an agent with probabilistic reasoning abilities: the agent can "think" probabilistic in all internal reasoning, but never can "think" probabilistically when talking, listening and trying to understand (i.e. communicating) other agents, at least when purely logical theories are used to fundament the communication. It has the additional consequence that an agent that reason only by probabilistic means cannot "use" FIPA acts, languages and protocols if it wants to keep theoretical consistency.

The second question arises from epistemological and linguistic considerations, when we take into account agents that can reason probabilistically. We will assume that the agent uses subjective (bayesian) reasoning and can assign probabilities to his beliefs, that is, the agent can reason with degrees of belief. Assuming only basic rationality for this kind of agent, then, if it has some probabilistic belief and needs to inform this belief to another agent it will need to be sure that the proper degree of belief be also correctly informed. For instance, if it strongly believes (90% of chance) that it will rain tomorrow and need to inform this belief to another agent to change his behavior (for example, to cancel some encounter), then it will need to convince the other agent to have the same strong belief about the possibility to rain tomorrow. Some appropriate locus for the transportation of this kind of probability needs to be found in current theories of communication. The problem is that the Speech Act Theory of Searle and Grice, which provides the epistemological and linguistic basis for formal communication theories, simply do not consider the possibility of agents to communicate knowledge of probabilistic nature because the most basic semantic "unit" of knowledge that is considered by the theory is a logical proposition. Consequently, all formal theories of communication (including, the Theory of Action, Intention and Communication of Cohen, Levesque [4,5] and Sadek [11,12]) have adopted this point of view and do not consider probabilistic knowledge communication as a real possibility.

Together both questions create a very interesting dilemma: if an agent use probabilistic reasoning and need to inform some probabilistic belief to another agent it will have serious problems to do this task, because current linguistic theories say that there is no means to accomplish it (according to these theories there is no *locus* to communicate probabilities). These theories, at least in their formal counterpart, say even more, stating that even if you can send this probabilistic knowledge there is no way to consider this knowledge when reasoning about communication tasks. This surely is not a good situation from a theoretical point of view, and our work will try to start to correct this problem, at least in the limited sense of FIPA formal agent communication model.

2.2 Related Work

The problems expressed in previous sub-section are not addressed in recent research literature about ACLs (see [3]). Research in this area and related areas of agent societies and social interaction is more focused in the study about logical aspects of social institutions, including trust relationship, intentional semantics for social interaction and similar concepts, but not in checking the role of probabilities in these concepts. A similar situation also occurs in the research area of probabilistic knowledge representation for MAS. Main papers in those areas are focused on the question of how to communicate and distribute BN probabilistic knowledge between agents [14], keeping the inference processes consistent, efficient and epistemologically sound. These pieces of research offer a separate form of knowledge representation and communication not related to ACL research. Our work intends to start to bridge this gap, by showing how probabilistic knowledge can be included in the FIPA communication framework in an integrated and uniform way.

Our approach to formalize the communication of probabilistic knowledge is based on the idea that the best way to do this, in a way that is integrated and compatible with current agent communication theories (at least in the FIPA case), is to use a modal logic that can handle probabilities, that is, to use a *probabilistic logic*. In terms of Artificial Intelligence research, probabilistic logics were first described by Nilsson [10], already using a possible-worlds model to define the semantic of his logic. The initial work of Nilsson was profoundly extended, in the beginnings of 1990, by the works of Halpern [9], Abadi [1] and Bacchus [2] mainly related to epistemic (or doxastic) probabilistic modal logics. Currently there is also an active line of research based on probabilistic extensions to the CTL* temporal logic from Emerson and Srinavan, like the PCTL logic of Segala. However, due to the nature of the theories of agent communication, that require BDI modal operators, we focused our research only on epistemic probabilistic modal logics.

3 SLP Probabilistic Logic

3.1 FIPA's *SL* Logic

The *SL* (*Semantic Language*) is a BDI-like modal logic with equality that fundaments FIPA communication standards. This logic was defined by Sadek's work [11,12], which attributes a model-based semantics for *SL* logic. In *SL,* there is no means of attributing any subjective probability (or degree of belief) to a particular belief of some agent, so it is not possible to represent or reason about probabilistic knowledge in this logic.

Besides the usual operators and quantifiers of the predicate logic with equality, *SL* contains modal operators to express the beliefs ($\boldsymbol{B}(a,\varphi)$), choices $\boldsymbol{C}(a,\varphi)$ and intentions ($\boldsymbol{I}(a,\varphi)$) of an agent a. *SL* also has a relatively obscure modal operator that defines an "absolute uncertainty" that an agent can have about some belief. The $\boldsymbol{U}(a,\varphi)$ operator, however, does not admit any kind of degree or uncertainty level. There is no clear connection between probability theory and $\boldsymbol{U}$ operator. It is also possible to build action expressions that can be connected in series $e_1;e_2;...;e_n$, in alternates $e_1|e_2$ or verified by an agent a $(a,e)?$. Temporal and possibility assertions can be made based on the fact that an action or event has happened (***Done***(e, φ)), on the possibility that an action or event may happen (***Feasible***(e, φ)) and on which agent is responsible for an action (***Agent***(a,e,φ)).

3.2 The *SLP* Logic

The extension of the *SL* logic is called *SLP*, for *Semantic Language with Probabilities*, and it is defined through the extension of the *SL* formal model. For such purpose, *SLP* will incorporate numerical operator, relations and expressions, and terms that denote probabilities expressing the subjective probability (degree of belief) of a given sentence or statement being true.

The probabilistic term ***BP***(a,φ) is specific for *SLP* and informs the probability of a proposition φ be true with respect to the beliefs of agent a, that is, it defines the subjective probability assigned to φ by a. For example, $\boldsymbol{BP}(a,\exists(x)(\mathrm{P}(x)) \leq \mathbf{1}$ express the fact that the subjective probability assigned by agent a to the possibility that some element of the domain satisfies $\mathrm{P}(x)$ is less than **1**.

The model-based semantics for formulas of *SLP* is defined over a set Φ of symbols for variables, functions, predicates, primitive actions, agents and constants through models M with the following structure:

$$M = \langle \mathcal{W}, \mathcal{A}gt, \mathcal{E}vt, \mathcal{O}bj, \mathcal{B}, \mathcal{C}, \mathcal{E}, \mathcal{AGT}, \sigma, \mathcal{RCF}, \mu \rangle$$

The elements $\mathcal{W}$, $\mathcal{A}gt$, $\mathcal{O}bj$, $\mathcal{E}vt$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{E}$, $\mathcal{AGT}$ and σ are part of the formal model originally defined for *SL* by Sadek [12]. They define the set of possible worlds ($\mathcal{W}$), agents ($\mathcal{A}gt$), primitive events ($\mathcal{E}vt$), objects ($\mathcal{O}bj$) and causative agent for primitive events ($\mathcal{AGT}$) of *SLP*. They also define the set of accessibility relations for beliefs ($\mathcal{B}$), choices ($\mathcal{C}$) and future worlds ($\mathcal{E}$) of *SLP*. The mapping σ denotes a standard first-order logic interpretation that attributes, for each possible world, function and predicate symbol in Φ a correspondent element in $\mathcal{A}gt \cup \mathcal{O}bj \cup \mathcal{E}vt$ (the logical domain of *SLP*).

The elements μ and $\mathcal{RCF}$ are new elements specifically defined to *SLP*. The set μ is a set of mappings that attributes to each agent a a discrete probability distribution function μ_a on the set of possible-worlds $\mathcal{W}$. The basic restriction to this set of mappings is that any mapping μ_a must respect the restrictions for any discrete probability function. The symbol $\mathcal{RCF}$ denotes the (up to isomorphism) closed field of real numbers. $\mathcal{RCF}$ it is the domain for the purely numerical formulas of *SLP* and includes addition and multiplication operations on real numbers, the neutral elements of these operations, the partial ordering $\leq_{rcf}$ and it satisfies all properties of real closed fields.

The formal semantics of *SLP* expressions, that are not probabilistic, are identical to the semantics given for *SL* in [12]. The presentation of the semantic for the entire *SLP* logics is out of the scope of present work (it is defined in [7]), however, here we will define the formal semantics of the basic belief relation ***B***(a,φ) and of the new probabilistic term ***BP***(a,φ), to show the correlation between these two constructions.

Definition 1. *The modal operator* $\boldsymbol{B}(a,\varphi)$ *expresses the fact that the agent a beliefs that the sentence* φ *is true in a model M, world w and evaluation function v if and only if* φ *is true in any world w' which can be reached from w using* $\mathcal{B}_a$ *the belief accessibility relation for the agent a*:

$M,w,v \models \boldsymbol{B}(a,\varphi)$ **iff** $M,w',v \models \varphi$, for all w' such that $w\,\mathcal{B}_a\,w'$.

Definition 2. *The semantic of the probabilistic term **BP(a,φ)** is the probability estimated by agent a that φ is true. This probability is calculated summing up the distribution function μ_a over the worlds where agent a believe that φ is true*:

$[\boldsymbol{BP}(a,\varphi)]_{M,w,v} = \mu_a(\{w' \mid w\mathcal{B}_a w' \text{ and } M,w',v \models \varphi\})$

Besides these definitions, we add two assumptions to the formal model of *SLP*.

Assumption 3. *The following equivalences are valid in SLP*:

$\boldsymbol{B}(a,\varphi) \Leftrightarrow \boldsymbol{BP}(a,\varphi)=\mathbf{1}$

$\boldsymbol{U}(a,\varphi) \Leftrightarrow \boldsymbol{BP}(a,\varphi)=\mathbf{0.5}$

This assumption states the basic relationship between probabilistic and non-probabilistic (i.e. purely logical) beliefs in *SLP* and between "absolute" uncertainties and probabilistic beliefs.

Assumption 4. *Any formula φ inside **BP(a,φ)** terms must be a sentence (a closed formula) of the logic. Numerical constants or variables cannot be used as arguments of logical predicates (and vice-versa)*.

The axiomatic system of *SLP* was built over the axiomatic system of *SL*. It incorporates all axioms and inference rule from *SL*. To support probabilities were added the axiomatic system for the real closed field of numbers and axioms and inference rules equivalent to Kolmogorov axioms for Probability Theory.

3.3 Properties of SLP Logic

The basic properties of SLP are enunciated in the following propositions.

Proposition 5. *Any valid formula of SL is also a valid formula of SLP and any purely logical valid formula of SLP is a valid formula of SL*.□

The proof of this proposition is not so simple because of assumption 3 which forces that every world with nonzero probability from a *M* model can be reached by any other world of this model through the $\mathcal{B}$ relation, something that is not required in *SL* (or in other epistemic modal logics). Even so, it was possible to prove in [7], that any valid model of *SL* is also a valid model of *SLP* and *vice-versa* and thus proves the proposition 5.

Proposition 6. *The axiomatic system of SLP is correct*.□

The new axioms and inference rules of *SLP* are derived from the axiomatic theory of probabilities from Kolmogorov and from the axiomatic theory of the real field, both proved correct axiomatic systems.

In our proposed extension to *SL*, we have taken special care to avoid the problem of undecidability of probabilistic logics described in [1]. We have found a very interesting result, showing that there is a simpler and intuitive set of restrictions, not so strong as the restrictions proposed by Halpern and Bacchus that keep the resulting axiomatic system complete.

Proposition 7. *The axiomatic system of SLP is complete if the axiomatic system of SL is also complete*.□

The basic insight that lead us to the (relative) completeness proof of *SLP* was based on the observation that the incompleteness proof for probabilistic logics made by Abadi and Halpern [1] relied on the fact that the same variables can be "shared" by terms inside probabilistic operator and logical formulas outside these operators, i.e., it is possible to have expressions like $P(x,y) \wedge \boldsymbol{BP}(Q(x))=r$, where the variable x is shared by $P(x,y)$ and $Q(x)$ inside the ***BP*** operator. The consequence is that if we not allow shared variables between probabilistic terms and logical formulas, then Abadi technique will not work. This is not the istic terms and logical formulas, then Abadi technique will not work. This is not the same to say that the corresponding axiomatic system is complete, but it shows that this should be possible. Indeed, if we do not allow this kind of sharing, as is the case of *SLP* because of assumption 4, it is possible to use proof techniques developed by Halpern [9] and separate the probabilistic and non-probabilistic parts of some formula. This is the basic method employed on the completeness proof of *SLP*. In [7] it was shown that the validity of any formula φ of *SLP* can be reduced to the validity of an equivalent formula $\psi \wedge \pi$, where ψ is a purely logical formula containing no numerical or probabilistic term and π is a purely numerical formula containing no logical predicate/term neither any probabilistic term.

In this case, the validity of formula ψ is entirely dependant on the original *SL* axiomatic system and the validity of π depends on the first order axiomatic theory of real closed fields that, by a well-known result of Tarski, is a decidable problem. This result was proved using a finitary generalization of the Halpern techniques presented in [9] to substitute probabilistic terms that contain closed first order modal formulas with universally quantified numerical variables.

4 Communication of Probabilistic Knowledge

4.1 Principles for Probabilistic Communication

The FIPA ACL semantic depends on several logical axioms that define principles for agency and communication theories (see [11,12] for details). The theory of agency employed by FIPA includes rationality, persistency and consistency principles for beliefs, choices and intentions of agents defined as *SL* axioms and theorems. The theory of communication is formed by several axioms that define communication principles like the belief adjustment, sincerity, pertinence and cooperation principles besides the 5 basic communication properties stated in FIPA ACL specification [6]. These principles are generally sufficient to handle reasoning needs for communication purposes in any rational BDI agent that is FIPA compliant (at least when the sender's agent centered semantics used by FIPA ACL is appropriate for the application or domain in question). In being so, our first principle can be stated as the following assumption.

Assumption 8 *Agents that need to communicate probabilistic knowledge and intend to use FIPA-ACL should also respect the theory of agency and the theory of communication proposed in FIPA standards*.

This assumption is perfectly reasonable because of compatibility between *SL* and *SLP* assured by proposition 5, that implies that any valid theory of *SL* is a valid theory of *SLP*. However, when agents use probabilistic reasoning and need to use this kind of knowledge for communication purposes, then the purely logical theories of agency and communication are not much useful. To handle these situations we propose that these theories be extended by two new principles that will be able to bridge the gap between purely logical considerations and probabilistic reasoning, in terms of agent's communication decisions. We will propose only a minimum set of new principles, strictly necessary to correlate probabilistic knowledge used by the agent to decision and inference processes related to communication tasks.

One fundamental property of FIPA theory is the principle that assures the agreement between the mental state of some agent and their beliefs [12]. Using this principle is possible to assert propositions like $\boldsymbol{B}(a, \varphi) \leftrightarrow \boldsymbol{B}(a, \boldsymbol{B}(a, \varphi))$ and $\boldsymbol{BP}(a,\varphi)=\mathbf{1} \leftrightarrow \boldsymbol{B}(a,\boldsymbol{BP}(a,\varphi)=\mathbf{1})$, if all propositions and predicate symbols in φ appears in the scope of a modal operator formalizing a mental attitude of agent *a*:

This is an interesting fact but is very limited in the case of probabilistic communication. The principles of FIPA's theory of communication assume that the agent must believe non-probabilistically in some fact, before the communication starts. Therefore, what we need is some principle that will allow us to correlate probabilistic beliefs with non-probabilistic beliefs. This is assured by the following proposition of *SLP*.

Proposition 9. *Principle of Probabilities and Beliefs Agreement: if some agent a assume that the probability of proposition φ is p, then this is equivalent to state that it also believe in this fact*:

$$\models \boldsymbol{BP}(a, \varphi)=p \leftrightarrow \boldsymbol{B}(a, \boldsymbol{BP}(a, \varphi)=p\,) \;\Box$$

This principle allows agents to put any probabilistic beliefs "inside" epistemic belief operators and then to use any other axioms and theorems of communication or agency theories to make communication related reasoning.

The proposition 9 is necessary but is not enough. We need some kind of reason to effectively start some new communicative act. In FIPA this is assured by the principle of belief adjustment [12] that states that if some agent *a* believe in *φ*, believe that is competent in this belief and thinks that another agent *b* do not believe in *φ*, then it adopts the intention to make *b* believe in *φ*.

$$\models \boldsymbol{B}(a, \varphi \wedge \boldsymbol{B}(b, \neg\varphi) \wedge \boldsymbol{Comp}(a, \varphi)) \rightarrow \boldsymbol{I}(a, \boldsymbol{B}(b, \varphi))$$

The predicate $\boldsymbol{Comp}(a,\phi)$ states the competence of agent *a* about *φ*.

The belief adjustment principle also falls in the same limiting situation of the mental state and belief agreement principle when applied to the probabilistic case. Therefore, we need another principle stated in the following proposition.

Proposition 10. *Principle of Probabilities Adjustment: if some agent a believe that the probability of proposition φ is p, believe that it is competent in this belief and also believe that another agent b have different estimation for the probability of φ, then it should adopt the intention to make agent b also believe that the probability of φ is p*:

$$\models \boldsymbol{BP}(a,\varphi)=p \wedge \boldsymbol{BP}(a,\boldsymbol{BP}(b,\varphi)=p)<\mathbf{1} \wedge \boldsymbol{B}(a,\boldsymbol{Comp}(a,\boldsymbol{BP}(a,\varphi)=p))) \rightarrow \boldsymbol{I}(a, \boldsymbol{BP}(b, \varphi)=p) \;\Box$$

This principle is derived from belief adjustment principle, using the proposition 9 stated before (see [7] for details). It will have the same function of belief adjustment principle for the probabilistic reasoning case, providing agents with intentions to solve perceived differences between probabilistic beliefs shared by several agents.

4.2 Communicative Acts for Probabilities

Like *SL*, *SLP* also can be used as a content representation language for FIPA-ACL communicative acts. This allows the representation and distribution of probabilistic knowledge like BN between agents using standard assertive (**inform**) acts. However, to do this is necessary to assume a particular structure in the contents of these acts. The assertive acts defined in Speech Act theory (and the equivalent **inform** FIPA-ACL acts) do not assume any particular internal structure in the propositions passed as contents of these acts. So, in the general case of probabilistic communication not seem reasonable to always assume a particular structure in the content of assertive act used to communicate probabilities. To handle this we propose that the strength (or weakness) of the assertive force of some speech act should be measured by a probability. In this way, any kind of propositions can be used as contents of these probabilistic assertive acts, because the (subjective) probability of the proposition will be transmitted as a graduation of the force. This graduation is a numerical coefficient that represents the subjective probability of the proposition (i. e., the graduation of the assertive force is directly related to the belief degree on the proposition). Two new probabilistic communicative acts were defined. They are considered extensions to the FIPA-ACL, creating the *Probabilistic Agent Communication Language* (PACL).

The acts **inform-bp** and **query-bp** acts are defined, respectively, to allow that the information about subjective probabilities of an agent to be shared with other agents and to allow that a given agent could query the degree of belief of another agent. Using the notation employed by FIPA-ACL [6] the **inform-bp** act is formalized as follows:

$<a, \textbf{inform-bp}\ (b, <\varphi, p>)>$

FP: $\boldsymbol{BP}(a,\varphi)=p \wedge \boldsymbol{BP}(a, \boldsymbol{BP}(b,\varphi)=p)<\mathbf{1}$

RE: $\boldsymbol{BP}(b,\varphi)=p$

This act informs the probability for some closed formula φ. The feasibility precondition of the act (*FP*) requires only that an agent to believe that the subjective probability of φ is p and that another agent b has the chance of not believing in this fact. In this case, if the other necessary conditions are fulfilled (see [6]), then the **inform-bp** act will be emitted. The rational effect (*RE*) that is expected with the act emission is that agent b also comes to believe that the probability of φ is p.

The **query-bp** act was also modeled after an analysis of the **query-if** act, which is its similar when dealing with truth-values. This directive act is used to retrieve the probabilistic information associated to a particular proposition.

4.3 Examples

The use of **inform-bp** acts is straightforward. Assume that some agent a believe that agent b have a different estimation of the probability of φ and also believe that his estimation is competent:

$$\boldsymbol{BP}(a,\varphi)=p \wedge \boldsymbol{B}(a, \boldsymbol{BP}(b,\varphi)\neq\boldsymbol{BP}(a,\varphi)) \wedge \boldsymbol{B}(a,\boldsymbol{Comp}(a,\boldsymbol{BP}(a,\varphi)=p))) \tag{1}$$

Using the axioms and inference rules of *SLP* it is possible to infer, from $\boldsymbol{B}(a, \boldsymbol{BP}(b,\varphi)\neq\boldsymbol{BP}(a,\varphi))$ and $\boldsymbol{BP}(a,\varphi)=p$, that $\neg\boldsymbol{B}(a, \boldsymbol{BP}(b,\varphi)=p)$. But this is equivalent to $\boldsymbol{BP}(a,\boldsymbol{BP}(b,\varphi)=p)<\mathbf{1}$, resulting:

$$\boldsymbol{BP}(a,\varphi)=p \wedge \boldsymbol{BP}(a,\boldsymbol{BP}(b,\varphi)=p)<\mathbf{1} \wedge \boldsymbol{B}(a,\boldsymbol{Comp}(a,\boldsymbol{BP}(a,\varphi)=p))) \tag{2}$$

Then, by (2) and proposition 10 the agent a need to assume the intention to inform b about the probability of φ. By the communication theory of FIPA this intention and beliefs stated in (2) are enough to cause the emission of the **inform-bp** act from a to b agent informing the probability of φ.

If we force that agents a and b use *SLP* as content language and require that agent a be completely unsure if agent b knows the probability of φ, then it is also possible to use the **inform** acts of FIPA. The principle stated in proposition 9 allows to infer, from $\boldsymbol{BP}(a,\varphi)=p$, that:

$$\boldsymbol{B}(a,\boldsymbol{BP}(a,\varphi)=p). \tag{3}$$

In FIPA **inform** act, the feasibility precondition (*FP*) also requires that agent a be completely unsure if the agent b knows some proposition ψ is stated as:

$$\neg\boldsymbol{B}(a, \boldsymbol{B}(b, \psi) \vee \boldsymbol{B}(b,\neg \psi) \vee \boldsymbol{U}(b, \psi) \vee \boldsymbol{U}(b,\neg \psi)) \tag{4}$$

Substituting ψ in (4) by $\boldsymbol{BP}(b,\varphi)=p$ we have:

$$\neg\boldsymbol{B}(a,\boldsymbol{B}(b,\boldsymbol{BP}(b,\varphi)=p) \vee \boldsymbol{B}(b,\neg\boldsymbol{BP}(b,\varphi)=p) \vee \boldsymbol{U}(b,\boldsymbol{BP}(b,\varphi)=p) \vee \boldsymbol{U}(b,\neg\boldsymbol{BP}(b,\varphi)=p)) \tag{5}$$

So, agent a believes in (3) and if it also believes in (5) it can emit an **inform** act to agent b, with the proposition $\boldsymbol{BP}(a,\varphi)=p$ as the content of the act.

5 Future Works

Several interesting developments can follow our work. A direct possibility it is to check the influence of probabilistic knowledge and reasoning in other types of communicative acts and interaction protocols. Particularly interesting and related to our ongoing research it is the application of probabilistic knowledge and reasoning to model formally negotiation protocols, mainly when these protocols are related to the pedagogical negotiation, which is a very complex form of interaction that occurs in intelligent learning environments (and classrooms) [8]. Another possibility is to use the logical representation schemes for BN (like the schemes presented in [2] and [7]) as a starting point for the research of shared ontologies for probabilistic knowledge. The considerable research work already done for logical based ontologies, can be applied to this new research.

7 Acknowledgments

The authors gratefully acknowledge the Brazilian agencies CAPES, CNPq and FAPERGS for the partial support to this research project

8 References

[1] M. Abadi and J. Halpern. Decidability and Expressiveness for First-Order Logics of Probability. In *Procs of IEEE Symp. on Foundations of Computer Science*, 30, 1989.

[2] F. Bacchus. Lp, a Logic for Representing and Reasoning with Statistical Knowledge. *Computational Intelligence*, 6:209-301, 1990.

[3] B. Chaib-Draa and F. Dignun. Trends in Agent Communication Language. *Computational Intelligence* v. 2, n. 5. Cambridge, MA: Blackwell Publ., 2002

[4] P. Cohen and H. Levesque. Rational Interaction as the Basis for Communication, In P. Cohen, J. Morgan and M. Pollack (Ed.). *Intentions in Communication*. Cambridge, MA: MIT Press, 1990.

[5] P. Cohen and H. Levesque. Communicative Actions for Artifical Agents. In *Procs of ICMAS-95*. San Francisco. Cambridge: MIT Press, 1995.

[6] FIPA. FIPA Communicative Act Library Specification, Std. SC00037J, FIPA, 2002.

[7] J. C. Gluz. Formalization of the Communication of Probabilistic Knowledge in Multiagent Systems: an approach based on Probabilistic Logic (In Portuguese). *PhD Thesis*. Instituto de Informática, UFRGS, Porto Alegre, 2005.

[8] J. C. Gluz, C. Flores, L. Seixas and R. Viccari, R. Formal Aspects of Pedagogical Negotiation in AMPLIA System. In: *Procs of TISE-2005*. Santiago, Chile, 2005.

[9] J. Y. Halpern. An Analysis of First-Order Logics of Probability. *Artificial Intelligence*, 46: 311-350, 1990.

[10] N. J. Nilsson. Probabilistic Logic. *Artificial Intelligence, Amsterdan,* 28: 71-87, 1986.

[11] M. D. Sadek. Dialogue Acts are Rational Plans. In: *Procs. of ESCA/ETRW Workshop on the Structure of Multimodal Dialogue*, Maratea, Italy, 1991.

[12] M. D. Sadek. A Study in the Logic of Intention. In: *Procs. of KR'92*, p. 462-473, Cambridge, USA, 1992.

[13] R. M. Vicari, C. D. Flores, L. Seixas, A. Silvestre, M. Ladeira, H. A. Coelho. Multi-Agent Intelligent Environment for Medical Knowledge. *Artificial Intelligence in Medicine*, 27(3): 335-366, March 2003.

[14] Y. Xiang, A probabilistic framework for cooperative multi-agent distributed interpretation and optimization of communication. *Artificial Intelligence*, 87: 295-342, 1996

Detecting and Repairing Anomalous Evolutions in Noisy Environments: Logic Programming Formalization and Complexity Results

Fabrizio Angiulli[1], Gianluigi Greco[2], and Luigi Palopoli[3]

[1] ICAR-CNR, Via P. Bucci 41C, 87030 Rende, Italy
angiulli@icar.cnr.it
[2] Dept. of Mathematics - Univ. della Calabria, Via P. Bucci 30B, 87030 Rende, Italy
ggreco@mat.unical.it
[3] DEIS - Univ. della Calabria, Via P. Bucci 41C, 87030 Rende, Italy
palopoli@deis.unical.it

Summary. In systems where agents are required to interact with a partially known and dynamic world, sensors can be used to obtain further knowledge about the environment. However, sensors may be unreliable, that is, they may deliver wrong information (due, e.g., to hardware or software malfunctioning) and, consequently, they may cause agents to take wrong decisions, which is a scenario that should be avoided. The paper considers the problem of reasoning in noisy environments in a setting where no (either certain or probabilistic) data is available in advance about the reliability of sensors. Therefore, assuming that each agent is equipped with a background theory (in our setting, an extended logic program) encoding its general knowledge about the world, we define a concept of detecting an anomaly perceived in sensor data and the related concept of agent recovering to a coherent status of information. In this context, the complexities of various anomaly detection and anomaly recovery problems are studied.

1 Introduction

Consider an agent operating in a dynamic environment according to an internal background theory (the agent's trustable knowledge) which is enriched, over time, through sensing the environment. Were sensors completely reliable, in a fully observable environment, the agent could gain a perfectly correct perception of environment evolution. However, in general, sensors maybe unreliable, in that they may deliver erroneous observations to the agent. Thus, the agent's perception about environment evolution might be erroneous and this, in turn, might cause that wrong decisions are taken.

In order to deal with the uncertainty that arises from noisy sensors, probabilistic approaches have been proposed see, e.g., [5, 6, 7, 14, 16, 21]) where evolutions are represented by means of dynamic systems in which transitions among possible states

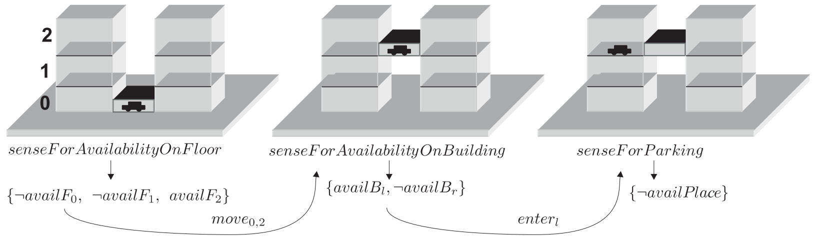

Fig. 1. Parking lot example.

are determined in terms of probability distributions. Other approaches refer to some *logic* formalization (see, e.g., modal logics, action languages, logic programming, and situation calculus [2, 11, 12, 20]) in which a logical theory is augmented to deal quantitatively and/or qualitatively with the reliability of the sensors.

In this paper we take a different perspective instead, by assuming that no information about reliabilities of sensors is available in advance. Therefore, in this context, neither probabilistic nor qualitative information can be exploited for reasoning with sensing. Nonetheless, it is in any case relevant to single out faulty sensor data in order for the agent to be able to correctly maintain a correct perception about the status of the world. To this aim, we introduce a formal framework good for reasoning about anomalies in agent's perception of environment evolutions, that relies on the identification of possible discrepancies between the observations gained through sensors and the internal trustable knowledge of the agent.

In order to make the framework clearer, we next introduce a running example.

1.1 Example of Faulty Sensors Identification

Consider an agent who is in charge of parking cars in a parking lot (see Figure 1). The parking lot consists of two buildings, each with several floors. The floors are reached via a single elevator which runs in the middle in between the two buildings (so, there is a building to *left* and one to the *right* of the elevator door). A number of sensors are used to inform the agent about parking place availability at different levels of the two buildings. In particular, the sensors tell the agent: (a) if there is any available parking place at some level in any of the two buildings (sensor s_1); (b) given the floor where the agent is currently located, if there is any available parking place in the left and/or the right building at that floor (sensor s_2); (c) given the floor and the building (left or right) where the agent is currently located, whether parking places are available at that floor in that building (sensor s_3) – let us assume that there are a total of n parking places at each level of each of the two buildings. Also, the agent uses a background theory that tells him that if he is at floor i of the building x and sensors s_1, when queried, signalled parking availability at level i and sensor s_2, when queried, signalled a parking availability in building x then there must be indeed at least one parking place available at his current position.

Now, assume that, in fact, the agent senses sensor s_3 and the sensor returns the information that no place is available at the current agent's position. This clearly disagrees with the internal state of the agent that tells that there should be indeed at least one place available in that position. Such disagreement implies that some anomalies came into play somehow.

In particular, the agent might doubt about the reliability of sensor s_3 (that is, there actually are available parking places at the agent position, but s_3 tells that none is available). Similarly, the agent might suspect that sensor s_1 is reliable while s_2 is not, thereby inferring that there is a place to park at the very floor where he is currently located, but on the opposite building.

1.2 Contribution and Organization

Within the framework outlined above, the contribution of the paper is as follows.

In Section 2, we introduce some preliminaries on extended logic programming which shall constitute the basic formalism exploited for modelling agents background knowledge. Then, in Section 3, we formally propose a concept of anomaly in state evolutions of a dynamic environment, as perceived by an agent sensing that environment through (possibly) noisy sensors. Moreover, we define a suitable concept of recovering the agent internal mental state from anomalies.

After that the framework has been introduced, we turn to the study of the computational complexity of some basic relevant problems related to state evolution anomaly detection and recovery. The results are proved and discussed in Section 4. We considered background knowledge bases modelled by means of not-free extended logic programs as well as general logic programs under both the brave and the cautious semantics. We point out here that, depending on the complexity of the agent background knowledge, anomaly checking may be characterized by a quite varied degree of difficulty, ranging from simple checking for the occurrence of complementary literals in sensor data and in the agent background knowledge (which is basically the case for our running example above) to quite complex tasks. The capability of characterizing computational complexity sources in knowledge representation frameworks is important both for gaining knowledge of the structure of the problems the framework comprises and, above all, to be able to realize effective rewriting and optimizations needed to efficiently implement them [10]. This justifies our interest in analyzing the complexity of anomaly detection and repair in agent evolutions, which will accounted for in the paper.

We believe that our investigation is a step towards providing capabilities for dynamic plan monitoring and repairing in noisy environment, where it can be useful for an agent that is trying to achieve its goals to be able to monitor, identify anomalies and fix a plan while evolving [3, 4, 8]. In this respect, it deserves of further work the possibility of prototypically implementing anomalies identification primitives for agent evolutions on top of some available answer set engine (e.g., [13, 17]), and subsequently made them available to conditional planning environments (e.g., [15, 19, 22]).

2 Preliminaries on Extended Logic Programs (ELPs)

We briefly recall here that a propositional ELP is a set of rules of the form $L_0 \leftarrow L_1, \ldots, L_m,\ not\ L_{m+1}, \ldots, not\ L_n\ (n \geq m \geq 0)$, where the symbol "$not$" denotes negation by default, and each L_i is a literal, i.e. an expression of the form p or $\neg p$ with p a propositional letter and the symbol "$\neg$" denotes classical negation. By $\mathbf{h}(r)$ we denote the head L_0 of the rule r, and by $\mathbf{b}(r)$ its body $L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n$. An ELP is *positive* if classical negation does not occur in the program.

In the following, we consider the *answer set* semantics for ELPs [9]. *Answer sets* of an ELP P are defined as follows. Let $Lit(P)$ denote the set of all the literals obtained using the propositional letters occurring in P. Let a *context* be any subset of $Lit(P)$. Let P be a *negation-by-default-free* ELP. Call a context S *closed under* P iff for each rule $L_0 \leftarrow L_1, \ldots, L_m$ in P, if $L_1, \ldots, L_m \in S$, then $L_0 \in S$. An *answer set* of P is any minimal context S such that (1) S is closed under P and (2) if S is inconsistent, that is if there exists a propositional letter p such that both $p \in S$ and $\neg p \in S$, then $S = Lit(P)$. An answer set of a general ELP is defined as follows. Let the *reduct of* P *w.r.t the context* S, denoted by $Red(P, S)$, be the ELP obtained from P by deleting (i) each rule that has $not\ L$ in its body for some $L \in S$, and (ii) all subformulae of the form $not\ L$ of the bodies of the remaining rules. Any context S which is an answer set of $Red(P, S)$ is an *answer set* of P. By $\mathrm{ANSW}(P)$ we denote the collection of all consistents answer sets of an ELP P. An ELP P is ANSW-consistent iff $\mathrm{ANSW}(P) \neq \emptyset$.

An ELP P *cautiously* entails a literal l, written $P \models_c l$, iff for each $S \in \mathrm{ANSW}(P)$, $l \in S$. An ELP P *bravely* entails a literal l, written $P \models_b l$, iff there exists $S \in \mathrm{ANSW}(P)$ such that $l \in S$.

3 Formal Framework

In this section, we introduce a simple framework to model environment state evolutions with sensing, and we formally state the problem of reasoning about possibly faulty sensors. In this respect, we present some techniques that an agent might exploit to identify 'anomalous' observations (and, hence, faulty sensors), and a 'repair approach' in execution monitoring accommodating the uncertainty on the outcome of the sensors.

3.1 Sensors, Agents, and Transitions

Let $\mathcal{F}$ be a set of propositional variables. We denote by $\neg f$ the negation of any $f \in \mathcal{F}$, and by $\mathcal{F}^{\neg}$ the set $\{\neg f \mid f \in \mathcal{F}\}$.

We distinguish two (disjoint) sets of variables: (i) *beliefs B*, denoting the agent beliefs about the status of the world; (ii) *observables O*, modelling the actual status of the world as returned by a set $\mathcal{S}$ of environment sensors. Specifically, for each sensor $s \in \mathcal{S}$, $\lambda(s) \subseteq \mathcal{O}$ denotes the set of propositional variables that are sensed by

$$\mathcal{B}:\begin{cases} floor_0,\ floor_1,\ floor_2\\ building_l,\ building_r\end{cases} \qquad \mathcal{S}:\begin{cases}\lambda(s_1)=\{availF_0,\ availF_1,\ availF_2\}\\ \lambda(s_2)=\{availB_l,\ availB_r\}\\ \lambda(s_3)=\{availPlace\}\end{cases}$$

$$\mathcal{O}:\begin{cases} availF_0,\ availF_1,\ availF_2\\ availB_l,\ availB_r\\ availPlace\end{cases} \qquad \mathcal{K}:\begin{cases} availPlace \leftarrow availF_i,\ availB_j,\\ \qquad\qquad floor_i,\ building_j.\end{cases}$$

$$\begin{cases} senseForAvailabilityOnFloor: & \langle floor_0,\ s_1\rangle\\ senseForAvailabilityOnBuilding: & \langle\emptyset,\ s_2\rangle\\ senseForParking: & \langle\emptyset,\ s_3\rangle\\ move_{i,j}: & \langle floor_i,\{\neg floor_i,\ floor_j\}\rangle\\ enter_j: & \langle\emptyset,\{building_j\}\rangle\end{cases}$$

Fig. 2. Formalization of the parking lot example.

s. Moreover, at any time instant, we assume that the value of the sensor is returned by a function $val : \mathcal{S} \mapsto \lambda(S) \times \lambda(S)^{\neg}$ producing a consistent set of observables (literals which the sensor evaluates to 'true').

Example 1. In the *parking lot* application, the sensors $\mathcal{S}$, the observables $\mathcal{O}$ and the beliefs $\mathcal{B}$ are reported in Figure 2, where, for instance, $availF_i$ means that there are parking places available at level i, $availB_l$ that at the current floor there are places available in the left building, and $building_r$ that the car is currently in the building on the right. $\triangleleft$

Each agent is characterized by a *background knowledge* $\mathcal{K}$ expressed as an extended logic program over $\mathcal{F}$, and, over time, by a current *state* represented as a pair of sets $S = \langle S_B, S_O\rangle$, where $S_B \subseteq B \times B^{\neg}$ and $S_O \subseteq O \times O^{\neg}$, such that both S_B and S_O are consistent. In the following, S_B (resp. S_O) will be denoted by $\mathcal{B}(S)$ (resp. $\mathcal{O}(S)$). In order to achieve its goals, the agent operates by executing *operators* that cause *transitions* between states, that is, they cause the environment to evolve together with the agent mental state. In particular, a transition usually changes the agent's beliefs; yet, it may change the observables if the corresponding operator includes a *sensing*.

Several ways to define transitions between states in the presence of sensing actions have been proposed in the literature, accounting, e.g., for non-deterministic effects, causal effects, probabilities, and so on (see, e.g., [20, 18, 14, 12, 21] and references therein). In the foregoing, we decided to refer to a particularly simple approach, since the results we are going to present are largely independent of the chosen formalization of transitions and associated operators. Hence, in order to make the exposition clearer we resume to a quite simple model which allows to specify preconditions, multiple-effects and *sensing actions*. So, in our context, an operator t for an agent A is simply a pair $\langle c, e\rangle$ such that c is a logic formula over the set $B \cup O$ denoting the preconditions, and e is the effect of the operator which can be either ***(i)*** a consistent subset e_B of $B \times B^{\neg}$, or ***(ii)*** a sensor $e_s \in \mathcal{S}$. We denote by $con(t)$ the precondition c, and by $\mathit{eff}(t)$ the effect e of t. See Figure 2 for the theory $\mathcal{K}$ and the set of operators in our parking lot application.

In the following, given a precondition c and an answer set S we will assume that the entailment $S \models c$ is polynomial time decidable. An ELP P *cautiously* entails a condition c, written $P \models_c c$, iff for each $S \in \mathrm{ANSW}(P)$, $S \models c$. An ELP P *bravely* entails a condition c, written $P \models_b c$, iff there exists $S \in \mathrm{ANSW}(P)$ such that $S \models c$.

We next define the semantics of applying operators to agent's states.

Definition 1. Let A be an agent and $\langle S_B, S_O\rangle$ be a state for it. An operator $t = \langle c, e\rangle$ is *applicable* in $\langle S_B, S_O\rangle$ if $(\mathcal{K} \cup S_B \cup S_O) \models c$, and the *result of its application* is the state $\langle S'_B, S'_O\rangle$ defined as:

- $\langle S_B, (S_O \setminus v^{\neg}) \cup v\rangle$, with $v = val(e_s)$, if $e = e_s \in \mathcal{S}$, and
- $\langle S_B \setminus e_B^{\neg} \cup e_B, S_O\rangle$, if $e = e_B \subseteq B \times B^{\neg}$.

In the case above, we also write $\langle S_B, S_O\rangle \rightarrow_t \langle S'_B, S'_O\rangle$. □

Example 2. Consider again Figure 2, and in particular, the set of operators reported in the bottom part of it: Given the state $\langle\{floor_0\}, \emptyset\rangle$, we can easily see that the operator $move_{1,2}$ is not applicable. Conversely, the agent might apply the operator $senseForAvailabilityOnFloor$ and a possible outcome is $\langle\{floor_0\}, \{\neg availF_0, \neg availF_1, availF_2\}\rangle$. Figure 1 shows an example of transitions between states exploiting such operators. ◁

3.2 Reasoning on Evolutions

The repeated application of operators define an evolution for the agent. Formally, an evolution H for A is a succession of states of the form $\langle S_B^0, S_O^0\rangle \rightarrow_{t_1} \langle S_B^1, S_O^1\rangle \rightarrow_{t_2} \ldots \rightarrow_{t_n} \langle S_B^n, S_O^n\rangle$, such that (i) each transition t_i is applicable in the state $\langle S_B^{i-1}, S_O^{i-1}\rangle$ and (ii) each state $\langle S_B^i, S_O^i\rangle$ is the result of the application of t_i in $\langle S_B^{i-1}, S_O^{i-1}\rangle$. Intuitively, H represents an actual plan that the agent is performing in order to achieve a given goal starting from the initial state $\langle S_B^0, S_O^0\rangle$.

In the following, $len(H)$ denotes the number of transitions occurring in the evolution H; $state_i(H)$ denotes the ith state of the evolution H; $state(H)$ denotes $state_{len(H)}(H)$; $tr_i(H)$ denotes the ith transition occurred in the evolution; $H[i]$ denotes the evolution $state_0(H) \rightarrow_{tr_1(H)} \ldots \rightarrow_{tr_i(H)} state_i(H)$.

As previously pointed out, while dealing with noisy sensors, there might be evolutions in which the agent finds some discrepancies between its mental beliefs (plus its trustable knowledge) and the observations at hand. The following definition formalizes such a notion of 'disagreement'.

Definition 2. Let H be an evolution for the agent A with knowledge $\mathcal{K}$. A set of observations $W \subseteq \mathcal{O}(state(H))$ is an *anomaly* for A in H if $\forall w \in W$, $th(A, H) \setminus W \models \neg w$, where $th(A, H)$ denotes the theory $\mathcal{K} \cup \mathcal{B}(state(H)) \cup \mathcal{O}(state(H))$. □

Example 3. Let H be the evolution: t_1 : $senseForAvailabilityOnFloor$; t_2 : $move_{0,2}$; t_3 : $senseForAvailabilityOnBuilding$; t_4 : $enter_l$; and t_5 : $senseFor Parking$.

Assume, now, that sensed values are such that $\mathcal{O}(state_1(H)) \supseteq \{\neg availF_0, \neg availF_1, availF_2\}$, $\mathcal{O}(state_3(H)) \supseteq \{availB_l, \neg availB_r\}$, and $\mathcal{O}(state_5(H)) \supseteq \{\neg availPlace\}$ – see, again, Figure 2. Intuitively, the agent is planning to park at the second floor in the left building after sensing s_1 and s_2. But, the result of sensing s_3 is anomalous, as it disagrees with its mental beliefs (in $\mathcal{K}$) according to which $availPlace$ should be true there. $\lhd$

The agent employed in our running example has a unique possible view of the world, being its knowledge a positive program. In general, an agent may have several possible worlds. Thus, in the following we will distinguish between the cautious and the brave semantics. In particular, while anomaly existence under the cautious semantics expresses that no possible world is consistent with the sensor readings, under the brave semantics a set of sensor readings is anomalous if each sensor reading is inconsistent with some possible world in which all sensors of the set are simultaneously kept quiet.

Given an anomaly, we are interested in finding possible fixes for it, i.e., "alternative" evolutions defined over the same set of transitions in which, however, the result of the sensing actions may differ from the evolution in which the anomaly has been singled out. This is formalized next with the notion of *repair* for an evolution.

Definition 3. An evolution H' for A is a *repair* for H w.r.t. an anomaly W if:
1. $len(H) = len(H')$,
2. $tr_i(H) = tr_i(H')$, for each $1 \leq i \leq len(H)$, and
3. $\forall w \in W \cap \mathcal{O}(state(H'))$, $th(A, H') \setminus W \not\models \neg w$.

Moreover, H' is *non trivial* if $W \cap \mathcal{O}(state(H'))$ is not empty. $\Box$

Example 4. For instance, a repair for our running example is obtained by replacing the value returned by sensor s_2 with $\{\neg availB_l, availB_r\}$ while keeping the values returned by s_1 and s_3. This represents the scenario in which the available place is in the opposite building of the same floor. $\lhd$

4 Reasoning with Noisy Sensors

Now that we have defined our formal framework for anomaly detection and repairing of an agent's mental state evolution, we turn to the problem of defining relevant agent's reasoning tasks. Moreover, as already stated in the Introduction, is it important to pinpoint the computational complexity characterizing such tasks, since this is a fundamental premise to devising effective and optimized implementations of our framework.

Specifically, we shall next consider the following relevant problems:

- ANOMALY-EXISTENCE: Given an agent A and an evolution H for it, does there exist an anomaly W for A in H?
- REPAIR-EXISTENCE: Given an agent A and an anomaly W for A in an evolution H, does there exist a (non trivial) repair H' for H w.r.t. W?

	not -free	cautious	brave
ANOMALY-EXISTENCE	P-c	Σ_2^P-c	NP-c
REPAIR-EXISTENCE	NP-c	Σ_2^P-c	Σ_2^P-c
REPAIR-CHECKING	NP-c	Σ_2^P-c	Σ_2^P-c
ANOMALY&REPAIR-CHECK.	P-c	D^P-c	D^P-c

Fig. 3. Complexity of Basic Problems.

- REPAIR-CHECKING: Given an agent A and evolutions H and H', is H' a repair for H w.r.t. some anomaly W for A?
- ANOMALY&REPAIR-CHECKING: Let A be an agent and H an evolution. Given an evolution H' and a set of observables $W \subseteq \mathcal{O}(state(H))$, is W an anomaly for A in H, and H' a repair for H w.r.t. W?

Complexity results concerning problems defined above are depicted in Figure 3. In the following, the complexity of the ANOMALY-EXISTENCE problem for a particular semantics of general logic programs is investigated.

Let T be a truth assignment of the set $\{x_1, \ldots, x_n\}$ of boolean variables. Then, we denote by Φ the boolean formula $\Phi = C_1 \wedge \ldots \wedge C_m$ in conjunctive normal form, with $C_j = t_{j,1} \vee t_{j,2} \vee t_{j,3}$, where each $t_{j,k}$ is a literal on the set of boolean variables $X = x_1, \ldots, x_n$. Recall that deciding the *satisfiability* of Φ is a well-known NP complete problem.

Theorem 1. ANOMALY-EXISTENCE *for ELPs under brave semantics is* NP*-complete.*

Proof:

(Membership) The problem can be solved by a polynomial time nondeterministic Turing machine that guesses a subset $W \subseteq \mathcal{O}(state(H))$ together with $n = |W|$ contexts $S_1, \ldots, S_n$ of $th(A, H) \setminus W$ such that $\neg w_i \in S_i$, and then checks in polynomial time that each S_i is an answer set of the reduct of $th(A, H) \setminus W$ w.r.t. S_i and, hence, of $th(A, H) \setminus W$.

(Hardness) Given the boolean formula Φ, consider the set of observables $\mathcal{O}(\Phi) = \{x_0, x_1, \ldots, x_n\}$, the sensor $s(\Phi)$ with $\lambda(s(\Phi)) = \mathcal{O}(\Phi)$, and the agent $A(\Phi)$ with knowledge $\mathcal{K}(\Phi)$:

$$
\begin{array}{ll}
r_0 : sat \leftarrow c_1, \ldots, c_m. & \\
r_{1,j} : c_j \leftarrow \sigma(t_{j,1}). & (1 \leq j \leq m) \\
r_{2,j} : c_j \leftarrow \sigma(t_{j,2}). & (1 \leq j \leq m) \\
r_{3,j} : c_j \leftarrow \sigma(t_{j,3}). & (1 \leq j \leq m) \\
r_{4,i} : \neg x_i \leftarrow not\ x_i, sat. & (0 \leq i \leq n)
\end{array}
$$

where $\sigma(x_i) = x_i$ and $\sigma(\neg x_i) = not\ x_i$, the operator $t(\Phi) = \langle \emptyset, s(\Phi) \rangle$, and the evolution $H(\Phi) = \langle \emptyset, \emptyset \rangle \rightarrow_{t(\Phi)} \langle \emptyset, \mathcal{O}(\Phi) \rangle$. Now we prove that there exists an anomaly $W \subseteq \mathcal{O}(\Phi)$ for $A(\Phi)$ in $H(\Phi)$ iff Φ is satisfiable.

($\Rightarrow$) Assume that there exists an anomaly W for $A(\Phi)$ in $H(\Phi)$. Then $th(A(\Phi), H(\Phi)) \setminus W \models \neg w$, $\forall w \in W$. As the negation of some observable x_i can be implied only by rule $r_{4,i}$, then it is the case that there exists an answer set M of $th(A(\Phi), H(\Phi))$ such that $sat \in M$. Consequently, $T(x_i) = \mathbf{true}$, $\forall x_i \in X \setminus W$, and $T(x_i) = \mathbf{false}$, $\forall x_i \in W$, is a truth assignment to the variables of Φ that makes the formula satisfied.

($\Leftarrow$) Assume that Φ is satisfiable, and let T_X be a truth value assignment to the variables in X that makes Φ true. Then $W = \{x_0\} \cup \{x_i \mid T_X(x_i) = \mathbf{false}\}$ is an anomaly for $A(\Phi)$ in $H(\Phi)$.

According to the theorem above, negation by default makes ANOMALY-EXISTENCE intractable. Moreover, under the cautious semantics the problem is more difficult than under the brave (see Figure 3), unlike most cases in which a kind of symmetry holds between the complexity of the two semantics (within the same level of the polynomial hierarchy).

5 Conclusions

In this paper we have defined a formal framework good for reasoning about agents' mental state evolution about environments sensed through possibly unreliable sensors. In our framework, no information (neither certain nor probabilistic) is assumed to be available in advance about the reliability of sensors. The agent's perception can however be maintained to encode a correct perception of the world through the identification and the resolution of discrepancies occurring between sensor delivered data and the agent's internal trustable knowledge, encoded in the form of an ELP under answer set semantics. After having defined the formal framework, in order to pinpoint main computational complexity sources implied in the implementation of the anomaly detection and repairing agent's mental state evolution, several reasoning problem have been considered and their complexity have been studied.

We note that the problem of *belief change* is only loosely related to work here done. Indeed, rather than being interested in revising the agent theory in order to entail the new information provided by the environment, we are interested in singling out environmental manifestations to be doubted about. The notion of minimal repair is indeed relevant in order to rank different possible repairs. We point out that several relations of preference between repairs can be embedded in the basic framework here introduced. Indeed, a natural form of preference relies in the number of agent observations the repair should change in order to recover its mental consistency, while it is also interesting to rank repairs depending on the number of anomalies they are able to fix. Furthermore, while sensors under consideration can only report binary states, the framework is not limited to the management of binary environmental measures, as many-valued discrete signals can be indeed simulated by sets of binary signals. Investigating the impact of enriching the framework with sensors delivering real-valued data is also of interest. Finally, it is interesting to explore how the presented framework could be embedded within a full-fledged conditional agent planning system. All those issues discusses above will be the topics of future investigation, while we are currently involved with the implementation of our system on top of the DLV system [13].

References

1. F. Angiulli, R. Ben-Eliyahu-Zohary, and L. Palopoli. Outlier detection using default logic. *in Proc of IJCAI'03*, pp. 833-838.
2. F. Bacchus, J.Y. Halpern, and H.J. Levesque. Reasoning about noisy sensors and effectors in the situation calculus. *Artificial Intelligence*, 111(1-2): 171-208, 1999.
3. M. Balduccini and M. Gelfond. Diagnostic reasoning with a-prolog. *TPLP*, 3(4-5): 425-461, 2003.
4. C. Baral, S.A. McIlraith, and T.C. Son. Formulating diagnostic problem solving using an action language with narratives and sensing, *in Proc of KR'02*, pp. 311-322.
5. C. Baral, N. Tran, and L.C. Tuan. Reasoning about actions in a probabilistic setting. *in Proc. of AAAI/IAAI'02*, pp. 507-512.
6. C. Boutilier, R. Dean, and S. Hanks. Planning under uncertainty: structural assumptions and computational leverage. *JAIR*, 11: 1-94, 1999.
7. C. Boutilier, R. Reiter, and B. Price. Symbolic dynamic programming for first-order MDPs. *IJCAI'01*, pp. 690-700.
8. M. Fichtner, A. Großmann, and M, Thielscher. Intelligent execution monitoring in dynamic environments, *Fundamenta Informaticae*, 57(2-4):371-392. 2003.
9. M. Gelfond and V. Lifschitz. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Comput.*, 9(3-4): 365-386, 1991.
10. G. Gottlob. Complexity and Expressive Power of Disjunctive Logic Programming. *in Proc of ILPS'94*, pp. 453-465.
11. E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, and H. Turner. Nonmonotonic Causal Theories. *Artificial Intelligence*, 153(1-2): 49-104, 2004.
12. L. Iocchi, T. Lukasiewicz, D. Nardi, and R. Rosati. Reasoning about Actions with Sensing under Qualitative and Probabilistic Uncertainty. *in Proc of ECAI'04*, pp. 818-822.
13. N. Leone, G. Pfeifer, W. Faber, F.Calimeri, T.Dell'Armi, T. Eiter, G. Gottlob, G.Ianni, G.Ielpa, C.Koch, S.Perri, and A.Polleres. The dlv system for knowledge representation and reasoning. *in Proc. of JELIA'02*, pp. 537-540.
14. M.L. Littman, J. Goldsmith, and M. Mundhenk. The Computational Complexity of Probabilistic Planning. *JAIR* 9:1-36, 1998.
15. J. Lobo. COPLAS: a COnditional PLannner with Sensing Actions. *FS-98-02*, AAAI, 1998.
16. S.M. Majercik and M.L. Littman. Contingent planning under uncertainty via stochastic satisfiability. *Artificial Intelligence*, 147(1-2):119-162, 2003.
17. I. Niemelä and P. Simons. Smodels: An implementation of the stable model and well-founded semantics for normal LP. *in Proc. of LPNMR'97*, pp. 420-429.
18. J. Rintanen. Expressive Equivalence of Formalisms for Planning with Sensing. *in Proc. of ICAPS'03*, pp. 185-194.
19. J. Rintanen. Constructing conditional plans by a theorem prover. *JAIR*, 10:323-352, 2000.
20. T.C. Son, P.H. Tu, and C. Baral. Planning with Sensing Actions and Incomplete Information Using Logic Programming. *in Proc of LPNMR'04*, pp. 261-274.
21. T.C. Son and C. Baral. Formalizing sensing actions A transition function based approach. *Artificial Intelligence*, 125(1-2):19-91, 2001.
22. M. Thielscher. Programming of Reasoning and Planning Agents with FLUX. *in Proc. of KR'02*.

Adding Semantic Web Services Matching and Discovery Support to the MoviLog Platform

Cristian Mateos Marco Crasso Alejandro Zunino Marcelo Campo

ISISTAN Research Institute - Also CONICET
Facultad de Cs. Exactas, Departamento de Computación y Sistemas, UNICEN
Campus Universitario - Paraje Arroyo Seco - (B7001BBO) - Bs. As., Argentina
email: azunino@exa.unicen.edu.ar

Summary. Semantic Web services are self describing programs that can be searched, understood and used by other programs. Despite the advantages Semantic Web services provide, specially for building agent based systems, there is a need for mechanisms to enable agents to discover Semantic Web services. This paper describes an extension of the MoviLog agent platform for searching Web services taking into account their semantic descriptions. Preliminary experiments showing encouraging results are also reported.

1 Introduction

Once a big repository of Web pages, images and others forms of static data, the Web is evolving into a worldwide network of *Web Services*, paving the way to the so-called Semantic Web [1]. A Web Service [2] is a distributed piece of functionality that can be published, located and accessed through standard Web protocols. The goal of Web services is to achieve automatic interoperability between Web applications by providing them with an infrastructure to use Web-accessible resources.

Several researchers agree that mobile agents will have a fundamental role to materialize this vision [3, 4]. A mobile agent is a computer program which is able to migrate between network sites to perform tasks and interact with resources. Mobile agents have good properties that make them suitable for exploiting the potential of the Web [5]: support for disconnected operations, robustness and scalability.

Despite the advantages mobile agents offer, many challenges remain to glue them with Web services. Most of these challenges are a result of the nature of the Web. From its beginnings the Web has been mainly designed for human use and interpretation. Hence, mobile agents cannot autonomously take advantage of Web resources, thus forcing developers to write hand-coded solutions that are difficult to extend, reuse and maintain. Besides, the inherent complexity of mobile code programming with respect to traditional non-mobile systems, has dwindled the massive adoption of mobile agent technology, limiting its usage to small applications and prototypes.

In this sense, we believe there is a need for a mobile agent development infrastructure that addresses these problems and, at the same time, preserve the key benefits of mobile agents for building distributed applications. To this end, we have developed MoviLog [6], a platform for building Prolog-based mobile agents on the WWW.

MoviLog encourages the usage of mobile agents by supporting a novel mechanism for handling mobility named RMF (Reactive Mobility by Failure). It allows programmers to easily build mobile agents on the Semantic Web without worrying about Web services location or access details. Furthermore, to take into account the semantics of services, we have extended MoviLog with support for semantic matching and discovery of Web services. The extension, called Apollo, enables an automatic interoperation between mobile agents and Web services with little development effort.

This paper is organized as follows. The next section introduces semantic Web services. Sect. 3 presents the MoviLog platform. Sect. 4 describes Apollo. Sect. 5 explains an example. Sect. 6 reports experimental results. Sect. 7 discusses the most relevant related work. Finally, Sect. 8 draws conclusions.

2 Semantic Web Services

Web services are a suitable model to allow systematic interactions of programs across the WWW. To hide the diversity of resources hosted by the WWW, Web services technologies mostly rely on XML, a structured language that extends and formalizes HTML. In this sense, the W3C Consortium has developed SOAP [1], a communication protocol based on XML. Besides, languages for describing Web services have been developed. An example is WSDL[2], an XML-based language for describing services as a set of operations over SOAP messages. From a WSDL document, a program can find out the specific services a Web site provides, and how to use and invoke these services.

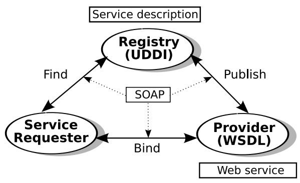

Fig. 1. Web services architecture

UDDI [3] defines mechanisms for searching and publishing Web services. By means of UDDI, Web service providers register information about the services they offer, thus making it available to potential clients. The information managed by UDDI ranges from WSDL files describing services to data for contacting providers.

Fig. 1 shows the conceptual architecture of Web services. A Web service is defined by a WSDL document describing a set of operations. A provider creates WSDLs for its services and publish them in an UDDI registry. A requester can browse registries to find services matching his needs. Then, the requesters can bind to the provider by invoking any of the operations defined by the WSDLs.

The weakest point of the architecture shown above is that it does not consider the semantics of services. To achieve an automatic interaction between agents and Web services, each service must be described in a nonambiguous and computer-understandable way. In this sense, some languages for Web services metadata annotation have emerged, such as RDF [4] and OWL [7], whose goal is to provide a formal

[1] SOAP (Simple Object Access Protocol): http://www.w3.org/TR/soap/
[2] WSDL (Web Service Description Language): http://www.w3.org/TR/wsdl
[3] UDDI (Universal Description, Discovery and Integration): http://www.uddi.org
[4] RDF (Resource Description Framework): http://www.w3.org/RDF/

model for describing the concepts involved in services. In this way, agents can *understand* and reason about the functionality a Web service performs, thus enabling the automatization of Web applications. Finally, a step towards the creation of a standard ontology of services is OWL-S [8]. The next section introduces MoviLog.

3 MoviLog

MoviLog [6] is a platform for programming mobile agents. The execution units of MoviLog are Prolog-based mobile agents named *Brainlets*. MoviLog uses strong mobility, where Brainlets execution state is transferred transparently on migration. Besides providing basic mobility primitives, the most interesting aspect of MoviLog is the notion of Reactive Mobility by Failure (RMF), a novel mobility model that reduces the effort for developing mobile agents by automating decisions such as when or where to migrate upon a *failure*. A failure is defined as the impossibility of an executing agent to obtain some required resource at the current site.

Roughly, each Brainlet possess Prolog code that is organized in two sections: *predicates* and *protocols*. The first section defines the agent behavior and data. The second section declares rules that are used by RMF for managing mobility. RMF states that when a predicate declared in the protocols section of an agent fails, MoviLog moves the Brainlet along with its execution state to another site that contains definitions for the predicate. Indeed, not all failures trigger mobility, but only failures caused by predicates declared in the protocols section. The idea is that normal predicates are evaluated with the regular Prolog semantics, but predicates for which a protocol exists are treated by RMF so that their failure may cause migration. The next example presents a simple Brainlet whose goal is to solve an SQL query given by a user on a certain database:

```
PROTOCOLS
  protocol(dataBase, [name(X), user(U), passwd(P)]).
CLAUSES
  doQuery(DBName, Query, Res):-
    dataBase([name(DBName), user('default'), passwd('')], Conn),
    doQuery(Conn, Query, Res), closeConnection(Conn).
  ?-sqlQuery(DBName, Query, Res):- doQuery(DBName, Query, Res).
```

PROTOCOLS section declares a protocol stating that the evaluation of *database(...)* predicate must be handled by RMF. In other words, the RMF mechanism will act whenever an attempt of connecting to the given database with the supplied username and password fails at the current site. As a result, RMF will transfer the agent to a site containing a database named *DBName*. After connecting to the database, the Brainlet will execute the query, and then return to its origin. Note that the protocol does not specify any particular value of the properties of the requested connection, which means that all unsuccessful attempts to access locally *any* database with *any* username-password combination will trigger reactive mobility.

Despite the advantages RMF has shown, it is not adequate for developing Web-enabled applications because it lacks support for interacting with Web resources. To overcome this limitation, RMF and its runtime support have been adapted to provide a tight integration with Web Services [9]. Also, to take advantage of services semantics, an infrastructure for managing and reasoning about Web services metadata named Apollo has been built. The rest of the paper focuses on Apollo.

4 Semantic Matching in MoviLog

Semantic matching allows agents to take advantage of ontologies by using inference capabilities. An ontology represents the meaning of terms in vocabularies and the relationships between these terms [1]. Reasoners are often used to infer knowledge from ontologies. We have developed a Prolog-based reasoner as a set of rules and facts for describing and manipulating ontologies. In addition, the reasoner includes matchmaking rules to determine semantic similarity between any pair of concepts.

4.1 Representing ontologies in Prolog

We have developed a reasoner on top of the OWL-Lite language [7]. Unfortunately, OWL-Lite only supports classification hierarchy and simple constraints, thus offering less expressiveness than other languages belonging to the OWL family. However, OWL-Lite ensures inference completeness and decidability.

Table 1. OWL to Prolog correspondence

OWL-Lite	Prolog	Description
Class	class(X)	X is a class.
rdfs:subClassOf	subClassOf(X,Y)	X is a subclass of class Y.
rdf:Property	property(X)	X is a property.
rdfs:subPropertyOf	subPropertyOf(X,Y)	X is a subproperty of property Y.
Individual	individualOf(X,Y).	X is an instance of class Y.
inverseOf	inverseOf(X,Y)	X is inverse to property Y.
equivalentProperty	equivalentProperty(X,Y)	X is equivalent to property Y.
equivalentClass	equivalentClass(X,Y)	X is equivalent to class Y.
Properties	triple(X,Y,Z).	X is related to Z by property Y.

Interestingly, OWL-Lite can be translated to first order logic [10]. Table 1 shows the Prolog counterpart for some of the OWL-Lite sentences supported by our reasoner. OWL-Lite classes and properties are represented as simple facts; relationships are expressed as RDF triples. An RDF triple is a structure with the form *triple(subject, property, object)* which indicates that *subject* is related by *property* to *object* value. OWL-Lite features such as cardinality, range and domain constraints over properties are represented as triples. For example, *triple(author, range, person)* states that property *author* must be an instance of the class *person.* In addition, equality, inequality and transitive sentences of OWL-Lite may indirectly relate a concept to another. Our reasoner defines the following set of rules for dealing with these relationships:

```
triple(X,E,Y):- equivalentProperty(P,E), triple(X,P,Y).
triple(Y,O,X):- inverseOf(P,O), triple(X,P,Y).
triple(X,T,Z):- transitive(T), triple(X,T,Y), triple(Y,T,Z).
```

The first rule states that X is related to Y by property E, if E is equivalent to P and X is related to Y by P. For example, if *author* and *writer* were equivalent properties, then *triple(article, writer, person)* holds. The second rule states that Y is related to X by property O whenever *inverseOf(P,O)* is true and X is related to Y by P. For example, if *hasPublication* and *author* were inverse properties, then *triple(person, hasPublication, article)* holds. The last rule handles transitive relationships between concepts: if *John* is *Paul*'s *advisor*, and *Paul* is *George*'s *advisor,* then *John* is *George*'s *advisor.*

Fig. 2 shows an ontology for documents. It defines that a *thesis* and an *article* are *documents*, both having one or more authors. A *thesis* has an *advisor*. Both *author* and *advisor* are properties with range *person*. A document has a title, a language and some *sections*. Finally, a section has a content. In the rules two new concepts appear: *Thing* and *owl:string*. Thing is the parent class of all OWL classes. Also, OWL includes some built-in datatypes.

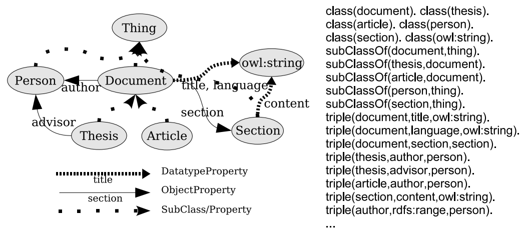

Fig. 2. An ontology for generic documents

4.2 Matching concepts

Ontologies can be used to describe data and services in a machine-understandable way. Automated data migration systems use ontologies to semantically describe their data structures. A process may then migrate a record from a source database to a sufficiently similar record in a target database. In automated Web services discovery systems, agents usually try to locate a sufficiently similar service to accomplish their current goal. Indeed, the problem to define what "sufficiently similar" means.

The degree of match between two concepts depends on their distance in a *taxonomy tree*. A taxonomy may refer to either a hierarchical classification of things or the principles underlying the classification. Almost anything can be classified according to some taxonomic scheme. Mathematically, a taxonomy is a tree-like structure that categorizes a given set of objects. We have defined four degrees of matching according to [11]. The rational to compute the similarity between two concepts X and Y is:

- **exact** if X and Y are individuals belonging to the same or equivalent classes, we label similarity as **exact**.
- **subsumes** if X is a subclass of Y we label similarity as **subsumes**.
- **plug-in** if Y is a subclass of X we label similarity as **plug-in**.
- **fail** occurs when none of the previous labels could be stated.

We have enhanced this scheme by considering the distance between any pair of concepts in a taxonomy tree (see Fig. 3). From the diagram, it can be clearly stated that *c2* is more similar to *b1* than *a1*: their similarity has been labeled as **plug-in**, but *c2* is hierarchically closer to *b1* than *a1*.

The matchmaking algorithm consists of a set of Prolog rules for calculating the distance between concepts within a taxonomy. The rule *match(C0, C1, Label, Dist)* returns the distance between *C0* and *C1* under *Label*. For example, the rule for equivalent classes is:

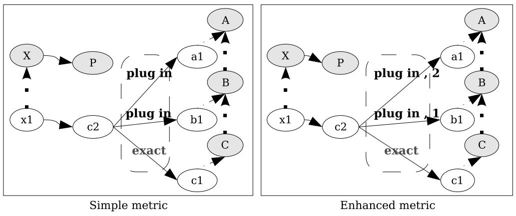

Fig. 3. Enhanced degree of match

```
match(X,Y,exact,0):- equivalentClass(X,Y).
```

The distance between two concepts is defined recursively as:

```
isSubClassOf(X,Y,1):- subClassOf(X,Y).
isSubClassOf(X,Y,N):- subClassOf(X,Z), isSubClassOf(Z,Y,T), N is T+1.
```

Applying the previous rules with X=article produces: *isSubClassOf(article,document,1)* and *isSubClassOf(article,thing,2)*. Matching rules for subsumes and plug-in labels use *isSubClassOf(X,Y,Z)* to compute distance as shown below:

```
match(X,Y,subsumes,N):- isSubClassOf(X,Y,N).
match(X,Y,plugin,N):- isSubClassOf(Y,X,N).
```

For space reasons, matchmaking support for properties is omitted. Nevertheless, the scheme previously discussed applies when computing distance between properties.

4.3 Semantic Web Services Discovery

In order to perform a semantic search of a Web service instead of a less effective keyword based search, an agent needs computer processable descriptions of services. Ontologies can be used for representing such descriptions. In this sense, OWL-S [8] aims at creating a standard service ontology. OWL-S consist of a set of predefined classes and properties for representing services. However, OWL-S is intended to describe services and how they must be invoked, but not how to semantically locate them. We combined OWL-S descriptions with UDDI registries to build a semantic Web services discovery system called Apollo. Fig. 4 shows its architecture.

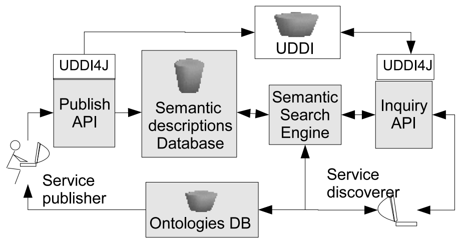

Fig. 4. The Apollo System

Apollo allows a Web service publisher to annotate services by using concepts from a shared OWL-S ontology database. Apollo is based on an OWL-S subset named Service Profile, which offers support for semantic description of services functionality, arguments, preconditions and effects. In this way, a publisher can describe services and its parameters in terms of concepts from the shared database. WSDL documents are stored in UDDI nodes by using UDDI4J[5]. Finally, each WSDL document and its concepts are associated through the *Semantic Descriptions Database*.

A search request contains a concept describing the desired service functionality, and two sets of concepts for in/out parameters. To perform a more effective search, service requests are forwarded both to UDDI registries and to the Semantic Search Engine. Data resulting from an UDDI search are transformed to concepts from the *Ontology Database* by a component that extends the UDDI Inquiry API.

The main component of the *Semantic Search Engine* is the semantic reasoner. It uses a matchmaking scheme and a simple algorithm for sorting the results of a service search according to the degree of match. The algorithm first tries to contact a Web service that semantically matches the requested conceptual output. If there are more than one Web service with the same degree of match for their output, the algorithm examines inputs to check that the requester is able to invoke the service. The pseudo code for the Web service rating algorithm is:

```
exact = 2; subsumes = 1; plug_in= 0;
MatchResult compare(MatchResult mr0, MatchResult mr1) {
    if (mr0.output.label > mr1.output.label) return mr0;
    else if (mr0.output.label < mr1.output.label) return mr1;
    else { if (mr0.output.distance < mr1.output.distance) return mr0;
           else if (mr0.output.distance > mr1.output.distance) return mr1;
    }
    /* Outputs match... Now compare input parameters. */
}
```

5 A sample scenario

Suppose we are deploying a network composed of sites that accepts Brainlets for execution. Some of these sites offers Web services for translating different types of documents (articles, forms, theses, etc.) to a target language. Every time a client wishes to translate a document, an agent is asked to find the service that best adapts to the kind of document being processed. In order to add semantics features to the model, all sites publish and search for Web services by using Apollo, and services are annotated with concepts from the ontology presented in Sect. 4.1 (see Fig. 2).

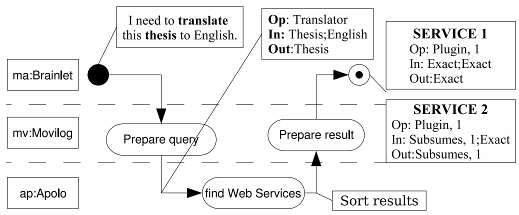

Fig. 5. A Brainlet for thesis translation

We assume the existence of different instances of Web services for handling the translation of a specific type of document. For example, translating a plain document may differ from translating a thesis, because a smarter translation can be done in this latter case: a service can take advantage of a thesis' keywords to perform a context-aware translation. Nevertheless, note that a thesis could be also translated by a Web service which expects a Document concept as an input argument, since Thesis concept specializes Document according to our ontology.

When a Brainlet gets a new document for translation, it prepares a semantic query. In this case, the agent needs to translate a thesis to English. Fig. 5 shows the activities

[5] UDDI for JAVA: http://www-124.ibm.com/developerworks/oss/uddi4j/

performed by each actor involved in the translation process. Before sending the service query, the Brainlet sets the service desired output as a Thesis. Also, the Brainlet sets the target language as english and the source document kind as Thesis, and then the semantic search process begins. Apollo uses semantic matching capabilities to find all existing Translation services. Let us suppose two services are obtained: a service for translating theses (*s1*) and a second service (*s2*) for translating any document.

After finding a proper list of translation Web services, Apollo sorts them according to the degree of match computed between the semantic query and services descriptions, and returns this new list back to the client. In the example, the degree of match for *s1* is greater than for *s2*, because *s1* outputs a Thesis (***exact***) while *s2* was labeled as ***subsumes*** with distance one.

```
PROTOCOLS
  protocol(webService, [ name(translate), input([thesis,english]), output(thesis) ]).
CLAUSES
  % The Prolog structure representing some thesis
  thesis([title('A title'), author('An author'), language(spanish),
         advisor('An advisor'), sections([...])}).
  ?-translate(TargetLang, Res):-
    webService([ name(translate), input([thesis,TargetLang]),
        output(thesis) ], WSProxy), thesis(Th), executeService(WSProxy,
        [Th, TargetLang], Res).
```

The previous code shows the implementation of the Brainlet discussed so far. As explained before, when the *webService(...)* predicate is executed, RMF contacts Apollo to find candidate services that semantically match the Brainlet's request. The evaluation of the predicate returns a proxy to the resulting service, which is used to effectively access it. The way the service is actually contacted (i.e. migrate to the service location or remotely invoke it) depends on access policies based on current execution conditions (network load, agent size, etc.) managed by the underlying platform.

6 Experimental results

In this section we report some experimental results. Particularly, we evaluated the performance of Apollo with regard to the number of published Web services. We generated a Semantic Web services database in an automatic fashion and we published it into Apollo. Both Apollo and all test applications were deployed on a Pentium 4 2.26 GHz with 512 MB of RAM, running Java 1.4.2 on Linux.

The Semantic Web services database was created by using two ontologies: a stock management domain and a car selling domain. Each service description was composed of five properties: input, output, category, preconditions and effects. Therefore, its input would be instantiated as a *cs:sportcar* concept, its output as a *cs:quote* concept, and finally its functionality as a *cs:car quoting* concept. Furthermore, another Web service can do the same for a "Sedan" car. In this case, since both *cs:sport* car and *cs:sedan* are *cs:vehicles*, service input would be instantiated as *cs:sedan*. Finally, searches have been simulated by using randomly generated conditions and expected results.

The resulting average response time for 600 random searches were: 2.37 ms (100 services), 12.65 ms (1000 services) and 149.33 ms. (10000 services). From this we can conclude that Apollo performance is good. Note that the overall response time is less than 200 ms for 10000 Web services descriptions.

Fig. 6 shows the relationship between database size and the time for processing 200 different searches. It can be seen that the worst response time is less than 600 ms. Note that the peaks of the curves are caused by the JAVA garbage collector.

7 Related work

Some related approaches are [12, 13, 14]. Most of them describe services by means of ontologies and a discrete scale of semantic similarity based on [11]. One limitation of these approaches is that their matching scheme do not consider the distance between concepts within a taxonomy tree. Hence, similarity related to different specializations of the same concept are wrongfully computed as being equal.

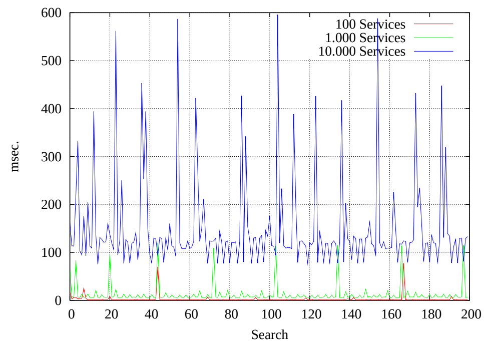

Fig. 6. # of searches vs. response time

The OWL-S Matchmaker [8] is a semantic Web service discovery and publication system. It includes a semantic matching algorithm based on service functionality and data transformation descriptions written in OWL-S. Data transformation descriptions are made in terms of service input and output arguments. Moreover, service search requests are enriched with concepts for describing the list of services that match a required data transformation. The OWL-S Matchmaker does not support taxonomic distance between concepts either.

In [15], a Web service is described by an OWL-S Service Profile instance or an extension of an existing profile. Semantic similarity between two services is computed by comparing their profiles' metadata instead of input/output concepts. A service request must contain the class associated to the *ideal* service profile (i.e. the one preferred by the requester), which is matched against published profiles. The drawback of this approach is its lack of support for finding available service profiles extensions.

Some interesting advances towards the integration of agents and Web services are ConGolog [16] and IG-JADE-PKSLib [17]. However, these approaches present the following problems: bad performance/scalability (IG-JADE-PKSLib), no/limited mobility (IG-JADE-PKSLib, ConGolog). In addition, none of the previous platforms provide support for semantic matching and discovery of Web services.

8 Conclusion and future work

This paper introduced Apollo, an infrastructure for semantic matching and discovery of Web services. Unlike previous work, Apollo defines a more precise semantic matching algorithm, implemented on top of a Prolog reasoner which offers inference capabilities over OWL-Lite to a semantic Web services search engine. In addition, the integration of MoviLog with Apollo enables the development of mobile agents that interact with Web-accessible functionality. This leads to the creation of an environment where sites can publish their capabilities as Semantic Web services, so that agents can use them.

In the context of Apollo, some issues remain to be solved. First, OWL-Lite needs to be replaced by a more powerful and expressive language, such as OWL DL or OWL Full. Second, the Ontologies Database content must be enhanced in order to provide a framework to describe, publish and discover other types of semantically-annotated Web resources (pages, blogs or agents), and not just Web services. Thereby an agent would be able to autonomously interact with Web services or Web content.

References

1. T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic Web," *Scientific American*, vol. 284, pp. 34–43, May 2001.
2. S. J. Vaughan-Nichols, "Web services: Beyond the hype," *Computer*, vol. 35, Feb. 2002.
3. J. Hendler, "Agents and the semantic web," *IEEE Intelligent Systems*, vol. 16, Mar. 2001.
4. M. N. Huhns, "Software agents: The future of Web services," in *Agent Technology Workshops 2002*, vol. 2592 of *Lecture Notes in Artificial Intelligence*, pp. 1–18, 2003.
5. D. B. Lange and M. Oshima, "Seven good reasons for mobile agents," *Communications of the ACM*, vol. 42, pp. 88–89, Mar. 1999.
6. A. Zunino, C. Mateos, and M. Campo, "Reactive mobility by failure: When fail means move," *Information Systems Frontiers*, vol. 7, no. 2, pp. 141–154, 2005. ISSN 1387-3326.
7. G. Antoniou and F. van Harmelen, "Web Ontology Language: OWL," in *Handbook on Ontologies in Information Systems* (S. Staab and R. Studer, eds.), Springer-Verlag, 2003.
8. M. Paolucci and K. Sycara, "Autonomous semantic Web services," *IEEE Internet Computing*, vol. 7, no. 5, pp. 34–41, 2003.
9. C. Mateos, A. Zunino, and M. Campo, "Integrating intelligent mobile agents with Web services," *International Journal of Web Services Research*, vol. 2, no. 2, pp. 85–103, 2005.
10. J. de Bruijn, A. Polleres, and D. Fensel, "Deliverable D20v0.1 OWL Lite, WSML Working Draft." `http://www.wsmo.org/2004/d20/v0.1/`, June 2004.
11. M. Paolucci, T. Kawamura, T. Payne, and K. Sycara, "Semantic matching of Web services capabilities," in *First International Semantic Web Conference*, vol. 2342, Springer, 2002.
12. K. Sivashanmugam, K. Verma, A. P. Sheth, and J. A. Miller, "Adding semantics to Web services standards," in *IEEE International Conference on Web Services*, June 2003.
13. I. Horrocks and P. F. Patel-Schneider, "A proposal for an owl rules language," in *The 13th international conference on World Wide Web*, pp. 723–731, ACM Press, Jan. 01 2004.
14. L. C. Chiat, L. Huang, and J. Xie, "Matchmaking for semantic Web services," in *IEEE International Conference on Services Computing (SCC'04)*, pp. 455–458, IEEE, 2004.
15. L. Li and I. Horrocks, "A software framework for matchmaking based on semantic Web technology," *International Journal of Electronic Commerce*, vol. 8, no. 4, pp. 39–60, 2004.
16. S. A. McIlraith and T. C. Son, "Adapting golog for programming the semantic Web," in *Fifth Symposium on Logical Formalizations of Commonsense Reasoning)*, May 20–22 2001.
17. E. Martínez and Y. Lespérance, "IG-JADE-PKSlib: An Agent-Based Framework for Advanced Web Service Composition and Provisioning," in *Workshop on Web Services and Agent-Based Engineering*, pp. 2–10, July 19–23 2004.

Learning Browsing Patterns for Context-Aware Recommendation

Daniela Godoy and Analía Amandi

ISISTAN Research Institute, UNICEN University
Campus Universitario, CP 7000, Tandil, Argentina
Also at CONICET, Argentina
{dgodoy,amandi}@exa.unicen.edu.ar

Abstract. The success of personal information agents depends on their capacity to both identify relevant information for users and proactively recommend context-relevant information. In this paper, we propose an approach to enable proactive context-aware recommendation based on the knowledge of both user interests and browsing patterns. The proposed approach analyzes the browsing behavior of users to derive a semantically enhanced context that points out the information which is likely to be relevant for a user according to its current activities.

1 Introduction

The main goal of personal information agents is to present relevant information to users based on the knowledge of their interests. In order to enable the adaptation and personalization of information delivered to users, personal agents learn and represent long-term interests into user profiles. Thus, user profiling addresses the issue of modeling interests to determine the relevance of a new, previously unseen piece of information.

In addition to merely choosing the right information, personal agents should also be aware of the user context in order to provide information in the time and in the place it is more relevant to users. User profiles are frequently seen as a way to disambiguate search topics. Even though this use of profiles supports interactive context-aware information retrieval in which relevant documents are gathered upon a direct user request, because of the lack of knowledge about the active goals, it fails at supporting proactive context-aware retrieval in which relevant documents are presented to users according to their activities [3].

In order to enable proactive context-aware information retrieval and recommendation, user behavior patterns as regards interests have to be explicitly modeled into profiles. The extraction of such patterns is fostered by semantically enriched profiles which provide a hierarchical organized view of the concepts a user is interested in. Either ontology-based profiling [8] or conceptual clustering [7] allow agents to obtain these hierarchies starting from examples.

In this paper, we present an approach to augment a hierarchical representation of user interests obtained by conceptual clustering with user behavior patterns extracted from observing the browsing activity. This enables proactive and adaptable behavior of personal agents which become able to predict and anticipate user information needs. Section 2 describes this approach. Experimental results are summarized in Section 3. Section 4 compares this work with related ones. Finally, concluding remarks are stated in Section 5.

2 Learning and Using Browsing Patterns

The browsing behavior of users is an important resource for inferring contextual information. It can be seen as a sequence of activities that are related to one another not only through evolving information interests that can be described at conceptual level, but also through proximity in time. By activity it is understood a page visit which takes place during the course of browsing, while groups of these activities can be referred to as sessions.

Information agents can take advantage of the knowledge gained from observing user browsing in conjunction with long-term user interests to retrieve context-relevant information. If an agent detects the user is browsing through certain interest categories, it can anticipate the categories in the same session the user is likely to be interested in. The goal of activity-awareness is, therefore, to proactively retrieve Web pages matching the user interests and compute a set of recommendations for the current or active user session.

To accomplish this goal, browsing patterns referring to categories in the user profile that are usually accessed together serve as the basis for recommendation and are mined starting from observation of frequent associations among browsing activities. For extracting navigational patterns, the existence of a conceptual hierarchy constituting the user profile is assumed, so that it can be used to characterize Web pages, i.e. to describe pages in terms of interest categories.

A conceptual clustering algorithm that carries out incremental, unsupervised concept learning over Web documents was used in this work to obtain such hierarchical descriptions of user interests. However, other approaches can be applied within this framework. Hierarchies of concepts produced by this algorithm, named *WebDCC* [7], are classification trees in which internal nodes represent concepts and leaf nodes represent clusters of examples.

In user profiles, browsing habits are represented by association of the form $A \Rightarrow B$, where A and B are groups of categories and the association indicates that, if the user current activities include visiting pages about the categories in A, the next activities are likely to include visiting pages about B.

2.1 Client-Side Sessionization

A browsing session is a set of page references that takes place during one logical period, e.g. the sequence of page accesses that takes place from a log in to a log out of the browser. By identifying the session boundaries, it is ensured that the information collected from one session is within the same context, which provides a good foundation for inferring and applying context in recommendation.

In contrast to Web usage mining, which focuses on extracting patterns of multiple users within server logs, a more accurate and complete picture of a user Web activity can be obtained from client side data. User actions can be recorded in an activity log by applications monitoring Web browsers. Thus, the content of Web pages, the access time, the time spent on each page and other information is available for analysis. Furthermore, actions such as opening or closing the browser can be used to start and finish browsing sessions.

From client-side observation, it is possible to reliably recognize sessions in the user activity log to evaluate the user interests as well as to understand user frequent browsing patterns. A session S_j is a list of pages a user accessed to ordered by time-stamp as follows:

$$S_j = \{(p_1, time_1), (p_2, time_2), \ldots, (p_n, time_n)\}$$

where $time_i$ is the time the user accessed the page p_i such that $time_i \leq time_j, \forall i \leq j$. Then, the user browsing activities are partitioned into a set of sessions $S = \{S_1, S_2, \ldots, S_k\}$ containing individual page references.

The process of segmenting the activity of a user into sessions is performed using a time-oriented heuristic in which a time-out establishes a period of inactivity that is interpreted as a signal that the session has ended. If the user did not request any page for a period longer than *max_time* (30 min. is used as default time-out) subsequent requests are considered to be in another session. In addition, the active session is finished when the browser is closed and a new session is started when the browser is re-opened.

2.2 Transaction Identification

The notion of session can be further abstracted by selecting a subset of pages that are significant or relevant for analysis. Each semantically meaningful subset of pages belonging to a user session is referred to as a transaction. Transaction identification assumes that user sessions have already been identified. Hence, the input to this process consists in the page references for a given user session. In Web usage mining there is no convenient method of clustering page references into transactions smaller than an entire user session [6].

To identify semantically meaningful transactions, content pages are considered as those belonging to one or more categories in the profile, unlike content pages in other approaches which are identified simply based on the time spent on a page or on backtracking during the user navigation [5]. Pages not belonging to any category in the profile are considered irrelevant for usage mining since they do no entail information about the user habits regarding interests. Then, a content-only transaction is formed by all the content pages in a session. Figure 1 illustrates the formation of these transactions.

The resulting transactions are further divided using the time window approach, which divides each transaction into time intervals no longer than a specified threshold. This approach assumes that meaningful transactions have an overall average length associated with them. For a large enough specified time window, each transaction will contain an entire user session. If W is the length of the time window, then two pages p_i and p_j are in the same session if:

$$p_i.time - p_j.time \leq W$$

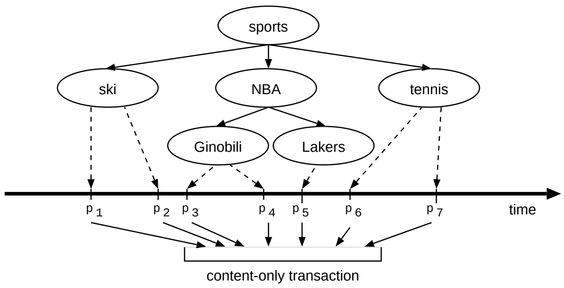

Fig. 1. Example of a content-only transaction

In this way, the set of pages $P = \{p_1, p_2, \ldots, p_n\}$, each with its associated $time_i$, appearing in the set of sessions S are partitioned into a set of m user transactions $T = \{t_1, t_2, \ldots, t_m\}$ where each $t_i \in T$ is a subset of P. The problem of mining association rules is defined over these collection of subsets from the item space where an item refers to an individual page reference.

To incorporate the knowledge of the user interests in pattern extraction, further processing of user activities is needed to map individual Web page references to one or more user interest categories. The enriched version of transactions leads to set of rules that includes categories. Thus, recommendations can be broaden to include any Web page belonging to the involved categories. To integrate content and usage data, each page p_i in a transaction t_j is considered to have an associated set of categories it belongs to, denoted $C_i = \{c_1, c_2, \ldots, c_p\}$, where C_i is extensionally defined by all the categories c_j in the path from the root of the hierarchy to the leaf cluster in which the page p_i was classified into.

If only the cluster a page belongs to is used to describe sessions, the discovered association rules will relate clusters but not categories. Instead, the inclusion of the ancestors in the path from the cluster the page was classified into until the root, makes it possible to find rules at different levels. The result of replacing the elements of the transactions in T by categories in the user profile is a set of transactions $T' = \{t'_1, t'_2, \ldots, t'_m\}$ where each $t'_i \in T'$ is a subset of C. The algorithm for transaction identification can be outlined as follows:

1. For each session $S_i \in S$, create a new transaction t_i in T
2. For each page $p_j \in S_i$, find the set C_j by classifying the page into the current user interest hierarchy
3. If $C_j \neq \emptyset$, add p_j to the transaction t_i since the page is a content page
4. Repeat steps 2 and 3 until all page references have been either added to the transaction or discarded
5. Repeat steps 1 to 4 until all sessions in S have been processed
6. Use the time window approach to partition each $t_i \in T$ into transactions smaller than W
7. For each resulting transaction $t_i \in T$, create the transaction t'_i in T' replacing each page $p_j \in t_i$ by the corresponding C_j

2.3 Mining Association Rules

The association rule mining problem was stated in [1]. Let $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$ be a set of literals called items, a subset $X \subseteq \mathcal{I}$ is called an itemset and a k-itemset is an itemset that contains k items. Let $\mathcal{D}$ be a database of transactions, where each transaction T is a set of items such that $T \subseteq \mathcal{I}$. Each itemset has a certain statistical significance called support such that an itemset has *support* s in the transaction set $\mathcal{D}$ if s% of the transactions in $\mathcal{D}$ contain X. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset \mathcal{I}$, $Y \subset \mathcal{I}$, and $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ holds in the transaction set $\mathcal{D}$ with *confidence* c if c% of the transactions in $\mathcal{D}$ that contain X also contain Y.

The problem of mining association rules in $\mathcal{D}$ consists in finding all rules $X \Rightarrow Y$ that have support greater than a user-specified minimum support, called *minsup*, and confidence, called *minconf*. For each rule, the support threshold describes the minimum percentage of transactions containing all items that appear in the rule, whereas the confidence threshold specifies the minimum probability for the consequent to be true if the antecedent is true.

In a hierarchical description of user interests, associations or access patterns may contain interesting regularities at different levels of abstraction including categories or clusters that are related according to the user habits. The problem of mining multiple-level or generalized association rules assumes a hierarchy or taxonomy $\mathcal{T}$ on the items instead of a flat itemset $\mathcal{I}$. A generalized association rule is an implication of the form $X \Rightarrow Y$, where $X \subset \mathcal{I}$, $Y \subset \mathcal{I}$, $X \cap Y = \emptyset$ and no item in Y is an ancestor of any item in X as this would be a trivially valid association. The rule $X \Rightarrow Y$ holds in the transaction set $\mathcal{D}$ with confidence c and support s if c% of the transactions in $\mathcal{D}$ that support X also support Y and s% of transactions in $\mathcal{D}$ support $X \cup Y$. These rules are called generalized association rules because both X and Y can contain items from any level of the taxonomy $\mathcal{T}$.

If the problem of determining if a transaction T support an itemset X is considered, for each item $x \in X$ it is necessary to check whether x or some descendant of x is present in the transaction. To simplify this task, all the ancestors of each item in T are added to this transaction to form an extended transaction T'. A straightforward method to find generalized association rules is to run any association rule algorithm on the extended transactions since T supports X if and only if T' is a superset of X. For empirical evaluation of the proposed approach, we used the *Apriori* algorithm over the set of extended transactions obtained as pages are classified in the concept hierarchy.

2.4 Activity-Based Recommendation

From user browsing sessions, patterns representing the user navigational behavior are extracted in the form of association rules, which relate sets of categories or concepts in the user profile. Information agents, therefore, become able to proactively retrieve relevant information to generate recommendations for a user by matching the current user activity against the discovered patterns.

To gather a set of possible recommendations, an agent can perform a Web search to retrieve pages belonging to the concepts the user is interested in. For example, the agent can retrieve pages from some fixed sites (e.g. a newspaper Web site) or find the nearest neighbors of a page in the profile used as query.

A fixed-size sliding window is used over the active session to capture the current user activity. For a sliding window of size n, the active session ensures that only the last n visited pages influence recommendation. The use of a window is important in discovering context since most users go back and forth while browsing to find the desired information so that earlier portions of the browsing history may refer to no longer valid information needs.

In the recommendation phase, the active session is compared with the discovered rules. If the active session matches the antecedent of an association rule, recommendations are finding by retrieving Web pages belonging to the categories in the rule consequent.

3 Experimental Results

To evaluate the activity-base recommendation approach, a client-side log of visited Web pages in a number of topics a user is interested in and the location of the pages on the user interest hierarchy are needed. Unfortunately, available datasets belong to individual Web sites and record the accesses of several users.

In the absence of client-side data, the content and logs of the *Music Machines*[1] Web site were used for experimentation. In these logs users are anonymized with respect to originating machine, i.e. all hits from one machine on a particular day have the same label. Thus, the browsing behavior of individual users can be interpreted respecting their interest categories within the site. *Music Machines* contains 4582 distinct pages about various kind of electronic musical equipment grouped by manufacturers.

Each access log consists of the user label, request method, accessed URL, data transmission protocol, access time and browser used to access the site. The server logs were filtered to remove those entries that are irrelevant for analysis and those referring to pages that do not exist in the available site copy.

From all users who entered the site after 20/8/98, when the copy of the site was made, the five users having the longest sessions were selected. Then, the experimental procedure simulates users browsing the *Music Machines* site and obtaining recommendations. For each user a profile was built based on both the content of Web pages from the site and the user behavior regarding interest categories. Experiments for each user proceed as follows:

1. Identify the user entries in the log files
2. Extract the URLs of the visited pages and run *WebDCC* algorithm over these pages using the available copy of the *Music Machines* site
3. Identify user sessions in the logs using *max_time*=30 minutes
4. Partition user sessions into transactions and mapping Web page references to categories in the profile

[1] http://www.cs.washington.edu/ai/adaptive-data/

ID	*duration*	*# entries*	*# filtered entries*	*# sessions*	*# pages*	*# clusters*	*# filtered clusters*
1	11:55:29	31404	938	9	1669	115	100
2	23:19:57	3639	2511	23	2427	229	145
3	13:18:47	3511	589	11	1663	129	91
4	21:28:38	3347	1087	10	1782	176	97
5	12:22:52	2862	2114	14	2851	312	176

Table 1. Summary of user data and experimental results

5. Divide the resulting set of transactions into a training (approx. 70%) and a testing set (approx. 30%) for experiments
6. Use the training set to mine association rules regarding categories
7. Use the testing set to simulate active session windows and recommend pages
8. Evaluate the recommendations in terms of precision and coverage

To assess quantitative values of recommendation performance, we used the adaptations of precision and coverage measures proposed by [9]. Given a transaction t and a set of recommendations R produced using a window w such that $w \subseteq t$, the precision and coverage of R with respect to t are defined as:

$$precision(R, t) = \frac{|R \cap (t-w)|}{|R|} \qquad coverage(R, t) = \frac{|R \cap (t-w)|}{|t-w|}$$

Thus, precision measures the degree to which recommendations are accurate for the active session and coverage measures its ability to recommend all the items that are likely to be visited by the user in the active session.

For a given transaction t in the testing set and an active session window of size n, we randomly chose $|t| - n + 1$ groups of items, each having size n, from the transaction as the surrogate active session windows. For each of these active sessions, a set of recommendations are produced based on the extracted rules. The recommendations are compared to the remaining items in the transactions, i.e. $t - w$, to compute performance measures. For each measure, the final score of the transaction t is the average over all of the $|t| - n + 1$ surrogate active sessions associated with this transaction.

The entries remaining after cleaning the logs were used to extract the documents each user accessed in the site. Table 1 summarizes the number of entries, sessions and unique pages accessed in the site. *WebDCC* algorithm was run over the documents each user accessed to identify the interest categories. These documents were partitioned into several clusters although no concepts were extracted by the clustering algorithm. This was mainly due to the site content and structure. It contains few pages referred to many manufacturers, so that different clusters are created for each of them and no generalization is possible.

From the total number of clusters resulting from running the algorithm, meaningless clusters containing a single instance were filtered out before association rule mining. Then, the pages in each session were partitioned into

W	# transactions	# items	# rules	# recom.	precision	coverage
3	120.20 ± 92.96	13.85 ± 3.62	247.20 ± 199.11	2.34 ± 2.06	75.79 ± 17.31	6.84 ± 6.85
			15423.00 ± 17533.67	3.87 ± 3.61	68.94 ± 20.90	10.13 ± 8.71
5	101.80 ± 98.05	19.73 ± 4.60	486.40 ± 409.05	4.87 ± 5.24	66.15 ± 21.81	8.67 ± 8.20
			16135.80 ± 15442.15	6.10 ± 5.61	59.73 ± 17.71	10.51 ± 8.49
10	45.60 ± 31.19	32.97 ± 6.41	920.00 ± 567.92	5.84 ± 5.48	63.97 ± 18.36	16.02 ± 13.07
			37223.60 ± 39060.43	7.81 ± 5.35	57.83 ± 13.69	19.34 ± 12.42
15	32.00 ± 20.02	46.73 ± 9.38	1255.60 ± 424.38	9.32 ± 4.04	62.32 ± 15.54	17.11 ± 6.90
			47948.40 ± 8319.91	11.44 ± 2.71	55.96 ± 9.32	17.89 ± 4.08

Table 2. Effect of time window in recommendation

transactions and, in turn, the pages in each transaction were mapped into clusters obtaining a set of content-enhance transactions.

There are several parameters influencing the results of recommendation, including the size of the time window W, the size of the sliding window n, the confidence threshold and the size of the itemsets.

The length of the time window W affects primarily the number of transactions obtained from a given session and, consequently, the number of rules and the quality of recommendations. In the first experiment, we investigated the impact of different values of W on the number of rules and the quality of the resulting recommendations for the five users by testing the values 3, 5, 10 and 15 minutes. In association rule mining, all rules having a support greater than 2% were extracted. For recommendation, we used a fixed active window of size 3 and a minimum confidence of 90%. Table 2 summarizes the average and standard deviation of values obtained for the five users in: number of transactions, items per transaction, extracted rules, recommendations, precision and coverage of recommendations. For each value of W in the table, the the results for rules having 1-itemsets and 2-itemsets are shown.

The higher the value W, the lower the number of transactions and the longer its size in terms of the average number of items. Fewer transactions lead to more rules since they are supported by the data, but the quality of these rules is inferior to the quality of rules extracted when more information is available. Indeed, the precision in recommendation decreases from $W = 3$ to $W = 15$. For further experiments we set $W = 3$ min. since the dropping in precision for the immediately next value $W = 5$ is significant (approx. 9%), but the improvement in coverage is rather small (approx. 2%).

The results of 1-itemsets and 2-itemsets, on the other hand, show the same relationship between precision and coverage. The values of precision diminish when 2-itemsets are considered, increasing the coverage of recommendations. In this case, not only the improvement in coverage can be considered small given the loss in precision, but also the number of rules rises drastically. The on-line analysis of such high number of rules becomes too expensive.

To investigate the effect of window size, the portion of the active window session used to produce recommendations, experiments were performed using

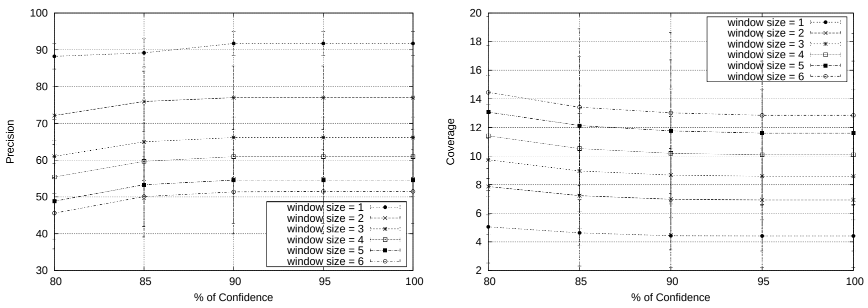

Fig. 2. Impact of window size on precision and coverage of recommendations

window sizes from 1 to 6. Figure 2 shows the impact of window size on precision and coverage of recommendations. In the figures, the results summarize the average and standard deviation of the scores achieved for the five users involved in the experiments varying the confidence threshold.

From both figures, it can be concluded that the smaller the size of the time window, the higher the precision of recommendation and the lower its coverage. Indeed, the best precision was obtained with a window of size one, but the coverage of recommendations was the poorer too. As the size of the window is enlarged, there are more rules matching the pages inside the window, increasing the number of recommendations and, consequently, their coverage.

There is a trade-off between enlarging enough the window size to recommend most of the pages that are relevant to the current activity context but not enlarge it too much to start recommending pages that were relevant to the previous context of the user in the browsing session. However, precision can be sacrificed in this decision for the sake of an increase in coverage since a lower precision means that the agent is recommending pages which are not contextually relevant but are still content relevant to the user interests.

4 Related Work

Efforts in building user profiles representing user interests have been frequently seen as a method of gathering contextual information since the knowledge contained in a profile persists across retrieval sessions and can be automatically added to queries. However, user profiles by themselves have not means to anticipate information needs given the user current activities and, therefore, do not support proactive context-aware recommendation. In our approach, the current activities act as trigger for retrieval of information matching long-term interests.

WordSieve [2] and *Watson* [4] are systems that use context for information seeking. *WordSieve* is an algorithm to build context profiles which distinguish sets of documents that users tend to access in groups. *Watson* observes the use of standard software tools and generates queries to seek context-relevant information. Instead of retrieving documents based solely on words extracted from recently consulted documents, our approach extracts information about how users tend to access documents regarding long-term interests to determine what kind of documents are likely to be interesting in a certain context.

The proposed approach differs from Web usage mining techniques in two aspects. First, server-side usage mining provides information about a specific Web site based on correlations among the pages that multiple users have visited. By contrast, our approach extracts rules from the observation of a single user browsing the Web. Second, most Web usage mining approaches obtain association rules that relate single Web pages. By capturing navigational patterns at conceptual level our approach provides more flexibility in recommendation.

5 Conclusions

The user context is an important aspect to take into account in recommendation which, however, has received little attention in personal agents. Most agents are concerned with estimating the interest of new pieces of information, instead of trying to place the relevant information in the right contexts. In this paper, we have described an approach to consider the activity context during profiling to enable context-aware recommendation. Experimental results showed that the extraction of association rules describing browsing patterns at conceptual level helps to predict part of the interests which are relevant to the user in a session.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.
2. T. Bauer and D. Leake. WordSieve: A method for real-time context extraction. In *Proceedings of the Third International and Interdisciplinary Conference on Modeling and Using Context*, pages 30–44, 2001.
3. P. Brown and G. Jones. Context-aware retrieval: Exploring a new environment for information retrieval and information filtering. *Personal and Ubiquitous Computing*, 5(4):253–263, 2001.
4. J. Budzik, K. Hammond, and L. Birnbaum. Information access in context. *Knowledge based systems*, 14(1-2):37–53, 2001.
5. M-S. Chen, J. S. Park, and P. Yu. Efficient data mining for path traversal patterns. *IEEE Transactions on Knowledge and Data Engineering*, 10(2):209–221, 1998.
6. R. Cooley, B. Mobasher, and J. Srivastava. Grouping web page references into transactions for mining world wide web browsing patterns. In *Proceedings of the IEEE Knowledge and Data Engineering Exchange Workshop*, pages 2–9, 1997.
7. D. Godoy and A. Amandi. Modeling user interests by conceptual clustering. *Information Systems*, 31(4-5):247–265, 2006.
8. S. Middleton, N. Shadbolt, and D. Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, 22(1):54–88, 2004.
9. B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Discovery and evaluation of aggregate usage profiles for Web personalization. *Data Mining and Knowledge Discovery*, 6(1):61–82, 2002.

Applying Collaborative Filtering to Reputation Domain: a model for more precise reputation estimates in case of changing behavior by rated participants

Alexandre Lopes, Ana Cristina Bicharra Garcia
Instituto de Computação - Universidade Federal Fluminense (UFF)
Rua Passo da Pátria, 156 Bloco E Sala 350 - 24.210-240 – Niterói – RJ.
alexcflopes@gmail.com, bicharra@ic.uff.br

Abstract. Automated Collaborative Filtering (CF) techniques have been successfully applied on Recommendation domains. Dellarocas [1] proposes their use on reputation domains to provide more reliable and personalized reputation estimates. Despite being solved by recommendation field researches (e.g. significance weighting [2]), the problem of selecting low-trusted neighborhoods finds new roots in the reputation domain, mostly related to different behavior by the evaluated participants. It can turn evaluators with similar tastes into distant ones, contributing to poor reputation rates. A Reputation Model is proposed to minimize those problems. It uses CF techniques adjusted with the following improvements: 1) information of evaluators taste profiles is added to the user evaluation history; 2) transformations are applied on user evaluation history based on the similarities between the taste profiles of the active user and of the other evaluators to identify more reliable neighborhoods. An experiment is implemented through a simulated electronic marketplace where buyers choose sellers based on reputation estimates generated by the proposed reputation model and by a model that uses traditional CF. The goal is to compare the proposed model performance with the traditional one through comparative analysis of the data that is created. The results are explained at the end of the paper.

1 Introduction

The goal of online reputation reporting systems and models applied in e-commerce systems [1, 3] is to restrain the participation of agents who have a poor-quality service history in electronic marketplaces. These models can combine both direct and indirect [4] information sources to better estimate their participants' reputation. The direct sources contain information on past encounters between the client and the rated supplier. The indirect one contain information indirectly acquired through other clients' witnesses [1, 3, 5] or through the analysis of their social relationship network, which is kept by the suppliers [4]. Even though reputation models based on direct sources are considered to be the most reliable way of estimating the supplier's reputation, there is a higher amount of available information through indirect sources, which should be used in case there's little probability of any two participants having a history of past encounters [4]. The problem with models based on indirect sources is the possibility that the reputation estimate will not be as reliable as desired, because it is difficult to measure precisely subjective aspects like quality of service. The reputation estimate is based on aggregates that do not reflect the differences in the client's taste or the context of the interaction [1, 3]. If it is poorly calculated, the reputation estimate can cause clients to interact with suppliers who they wouldn't choose to transact with otherwise.

1.1 Automated Collaborative Filtering applied to Reputation Models

For more reliable calculation of reputation estimates, Dellarocas [1], proposes the incorporation of the Automated Collaborative Filtering (CF) technique to the reputation model. The CF has been used in Recommendation and Information Filtering Systems [6]. It identifies similarities between an active user and other users based on the similarities in past ratings on common items, and uses this similarity to generate recommendations about items not yet rated by the active user. The neighbor users are the ones who have higher similarity factor with the active user.

The goal of incorporating the CF to reputation domains is to estimate the reputation of a supplier in a personalized way, calculating it based on the ratings of clients that have similar tastes with those of the active client.

1.2 The "False Good Neighbor" Problem

This problem is described in the Recommendation Systems literature as being a situation in which clients calculated as having greatest similarity with the active client are, in fact, not that similar [2, 7]. It can happen because of coincident ratings and because of a low number of common ratings between the clients. The very work of Herlocker [2] already proposes solutions for this issue, however, there are other factors in the reputation domain that can contribute to situations of false neighborhood, such as changes in the supplier's behavior from one encounter to another and rating manipulation by clients with bad intentions. In this paper we intend to explore the changes in the supplier's behavior. Suppliers who change their behavior from one encounter (and rating) to another may cause deviations in the similarity detection between clients, causing clients with similar preferences not to be considered neighbors, or inversely, causing clients with different preferences to be considered neighbors. It is important to notice that these problems do not happen in conventional Recommendation Systems because the rated items are products and not rational entities (humans or computers). Products don't have behavior, their characteristics and looks are normally maintained after each rating, while a supplier has goals, which influence his actions and behavior [4]. Apart from timely aspects, if two clients disagree on a product's rating, it is safe to say that it's because they have different tastes. In the case of supplier's rating, if they disagree on a rating, there are no guarantees that the suppliers maintain the same behavior with each client.

2 Proposed Model

This paper proposes a way of minimizing this problem adjusting the CF technique to calculate more precisely the similarity between users, considering, besides the ratings they enter, their preferences, which are presented through reputation rating issues (price, quality, etc.), that have been used as base for the rating. The model also aims to be applied on application domains as Electronic Marketplaces and partially decentralized P2P information sharing systems.

2.1 Automated Collaborative Filtering applied to Reputation Models

An example of a matrix of ratings in the reputation domain is presented in table 1.

Table 1. Matrix of ratings in a reputation system

Item rated / Rater	Client 1	Client 2
Supplier X	3.8	3.8
Supplier Y	1.1	3

To minimize the negative effects of a supplier's behavior change on the reliability of the recommendations, we propose a rating history adjustment based on taste similarity between the clients, which general scheme is presented in the following algorithm.

2.2 Proposed model's high level algorithm:

1. The active client chooses one of two operational modes [8]: the Prediction Mode – in which you estimate the reputation value of a supplier with whom to interact; or the Recommendation Mode – in which you generate a list of recommendations sorted by the highest reputation values of the estimated suppliers; in the first case, a premise is that the client can use any mediation resource available to locate suppliers and negotiate with them. In the second case, choosing the supplier is made based on the recommendation list generated.
2. The Reputation Service recovers all common rating history between the active client and other clients (neighbors), creating an Active Client Matrix of Ratings;
3. Before calculating the similarity with each neighbor, and to minimize the problems of changing behavior, the Reputation Service adjusts the Active Client Matrix of Ratings, with each cell being recalculated as described in the "Matrix of Ratings Transformation" section;

4. It is applied the traditional CF algorithm over the transformed matrix, which will generate the chosen supplier's predictive reputation value, or a recommendation list.
5. The client decides if he/she will start a transaction according to the recommendations or predictions generated.
6. At the end of the transaction, the client writes a testimonial on the supplier's reputation. Besides the general reputation, every testimonial must have the reputation values rated for each rating issue and the preferences of the client (more details in the following section).

2.3 Matrix of Ratings Transformation and reputation calculation

Extended Matrix of Ratings

In order for the transformation indicated in step 3 to take place, it is necessary to work with an Extended Matrix of Ratings (table 3) that contemplates, besides the General Reputation Ratings, the client's tastes and the reputation values given by rating issue (contract clause). The client's tastes are represented in the model as "Reputation Preferences", which are data structures inspired in the Behavioural Aspects **φ** and Ontological Structures defined in the ReGreT [9] model. The conceptual representation of the Extended Matrix of Ratings, as well as of the Reputation Preferences, can be seen in the class diagram, Fig.1, and described below.

Each and every Reputation Rating is associated to a Contract and to the Client's current Reputation Preference. The Reputation Preferences change as time goes by and are used in the supplier's reputation rating task. They are composed of Rating Issues. Each contract clause is related to a Rating Issue that has a weight and a rating formula. The Rating Issues indicate how deviations of the final values in relation to the agreed values influence (negatively or positively) the Behavioural Aspect. In this sense, they have similar function to the Ground Relations defined in the ReGreT model. Such influence must be calculated through a domain-dependent formula that is described in the Expression attribute of the RatingFormula class. The proposed model shares the ReGreT's premise that reputation is a complex concept (Complex Behavioural Aspects) rated through the combination of various simpler rating dimensions (Simple Behavioural Aspects), table 2. Thus, every Reputation Preference is made up of various Rating Issues, which, combined with its weight, would determine the general reputation value.

Table 2. Example of Rating Issues combinated into a Reputation Preference

Reputation Preferences (Complex Behavioural Aspect)	**Rating Issue** (Simple Behavioural Aspect)	**Issue**	**Weight**	**Influence**
Good-seller	Offers_High_Price	Price	0,6	Negative
	Offers_Good_Quality	Quality	0,4	Positive

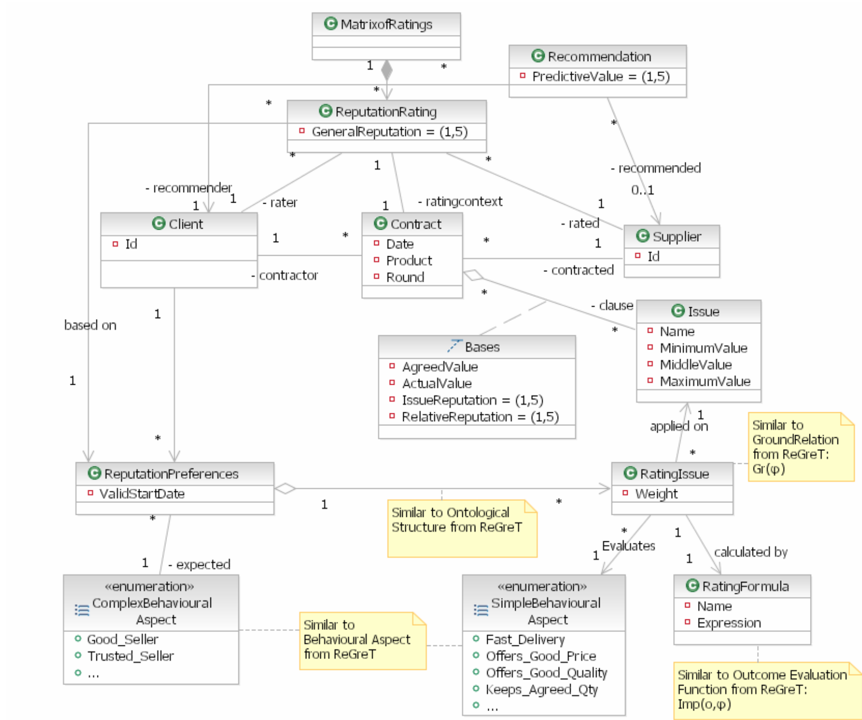

Fig. 1. Complete conceptual model (UML 2.0 notation)

Table 3 is an Extended Matrix of Ratings from table 1. The reputation preferences are in the “weight” line, while the reputation values given by rating issues are in the “Rating” line. The relative rating is the product of the weight of the issue and the reputation rating of the issue.

Table 3. Example of Extended Matrix of Ratings

	Rater	Client 1				Client 2			
Rated	**Issue**	**Quality**	**Price**	**Date**	**General**	**Quality**	**Price**	**Date**	**General**
Supplier X	Weight	0.3	0.2	0.5		0.2	0.3	0.5	
	Rating	5	4	3		4	5	3	
	Relative Rating	1.5	0.8	1.5	3.8	0.8	1.5	1.5	3.8
Supplier Y	Weight	0.3	0.2	0.5		0.3	0.2	0.5	
	Rating	3	1	0		4	4	2	
	Relative Rating	0.9	0.2	0	1.1	1.2	0.8	1	3

2.4 How to perform the adjustments over the Extended Matrix?

The equations (1) (2) and (3) implement the necessary calculations for reputation rating as well as the Matrix of Ratings adjustments shown in step 3 of the proposed model’s high level algorithm.

(1) $$Ar_{neighbor,supplier} = R_{activeclient,supplier} * sf_{activeclient,neighbor,supplier}$$

(2) $$R_{activeclient,supplier} = \frac{\sum_{issue} (rt_{supplier,issue} * w_{activeclient,issue})}{\sum_{issue} (w_{activeclient,issue})}$$

(3) $$sf_{activeclient,neighbor,supplier} = \text{cosine}\,(\text{preferences}_{activeclient,supplier}, \text{preferences}_{neighbor,supplier})$$

Where,

$Ar_{neighbor,supplier}$ – is the reputation value of each Matrix of Ratings cell, adjusted accordingly to the similarity factor between the Active Client and the Neighbor. The examples in tables 4 and 5 illustrate how the Matrix of Ratings transformation takes place.

$R_{activeclient,supplier}$ - is the supplier's reputation according to the perspective of the active client. It is the pondered average of the active client's reputation rating on every issue of the contract and not in only one reputation value, as it happens in other systems like eBay (www.ebay.com). The calculation formula is independent of the application domain and the quantity of issues of the negotiated contracts, and always result of growing scale values of real numbers between 1 and 5.

$rt_{supplier,issue}$ - is the supplier's reputation accordingly to a determined issue of the contract. The calculation format depends on the application domain and the rating of the Behavioural Aspect (examples described in the "Experiment" section), but it must produce values between 1 and 5 so as to not compromise the supplier's reputation calculation (eq. 2).

$w_{activeclient,issue}$ – is the weight and the importance given by the active client to the issue. The weight is a real number between 0 and 1 given by the client, which may vary as time goes by. The sum of the issue's weights must always total 1.

$sf_{activeclient,neighbor,supplier}$ - represents the similarity factor between the active client and a determined neighbor. It is determined through a cosine function, which calculates the distance between the rating issues' weight vectors, and so, identifying similarities in the reputation preferences applied by the active client and its neighbors when rating a common supplier's reputation.

2.5 Example of the Matrix of Ratings transformation

Considering Client 1 as being the active client, and applying the equations (1) (2) and (3) on a Matrix of Ratings from table 3 (and simplified table 4), we have the results in the Adjusted Matrix from table 5. This Matrix is used as input to step 4 of proposed model's high level algorithm.

Table 4. Active client 1's Matrix of Ratings

	Rater	
Rated	**Client 1**	**Client 2**
Supplier X	**3.8 (R)**	3.8
Supplier Y	**1.1 (R)**	3

Table 5. Adjusted Matrix of Ratings

	Rater	
Rated	**Client 1**	**Client 2**
Supplier X	3.8	**3.7 (Ar)**
Supplier Y	1.1	**1.1 (Ar)**

3 Experiment

The experiment's goal is to prove that the proposed model minimizes the effects on the seller's behavior change. It was assembled so it could be possible to compare the performance of the proposed model accordingly to a model based on traditional CF technique, through a system that simulates an e-commerce product marketplace, and that registers the effects of the rated seller's changing behavior. The experiment was inspired in the work conducted in [10], incorporating its organization and way of measuring the performance of the tested models. However, the system's architecture is different because the proposed reputation model is based exclusively on witnesses recovery by collaborative filtering mechanisms.

The simulation is made of 16 buyers and 64 sellers. Half of the buyers receive seller's recommendations based on the proposed CF algorithm, and the other half, based on the traditional algorithm. The transactions occur in 64 rounds, being 51 training rounds and 13 test rounds. The goal of the training rounds is to prepare the reputation database and to prevent the low dispersion of ratings from damaging the tested CF algorithm performance. During these rounds, the reputation module does not provide recommendation (sellers are randomly selected), it is only fed by the buyer's ratings. The test rounds are for monitoring buyer performance and for complementing the comparative analysis between the tested reputation models. Buyers select sellers with the highest reputation prediction value in the recommendation lists generated by the reputation module. In each round, it is possible to occur up to 16 transactions, totaling 1024 transactions per simulation. The maximum number each buyer can close per round is one transaction. Each seller can participate in one or more transactions per round.

The system has modules that simulate buyers and sellers, and a reputation service capable of generating recommendation based in the traditional CF technique and in the proposed technique. The reputation service implements two CF algorithms with user-to-user correlation developed from the algorithm originally proposed by Resnick in [6]. They share common configurations, like similarity calculation between neighbors through Pearson's coefficient, and the usage of a neighborhood selection method by a maximum amount of neighbors (best-n-neighbors) [2, 7] (configured to 30 neighbors).

When starting a transaction, both buyer and seller agree on the price and the quality of the product. The initial agreement is established based on the middle values (30,00 for price and 3 for quality). During the transaction, the seller can change the agreed values with the buyer according to his/her behavior, what will influence in the outcome of the transaction. There are three types of behavior:

Bad– During training rounds, in 60% of the transactions they increase the initially agreed price in ¼ and decrease quality in ¼. In the rest of the transactions, they present a similar behavior than the Good one. The originally defined percentage in the experiment with ReGreT was 75%, however, in this case, it makes more sense to define it as 60% so as to make the behavior changes more frequent. During test rounds, they increase price and decrease quality in 100% of the rounds. The sellers are configured like this to reproduce the false neighborhood situations in which other buyers with similar preferences are not considered neighbors, and vice-versa.
Good - Along all rounds, this type of seller increases the quality of product in ¼, and decreases its price in ¼. They are configured this way so as to benefit buyers who effectively receive the best recommendations, and then, represent a counterpoint relative to the disappointment of a buyer in case he/she receives a bad recommendation.
Neutral– There are no changes on the agreement of the contract in any of the rounds.

When closing a transaction, the buyers rate the seller's reputation, and update their cash. The seller's reputation is rated in a growing scale of real numbers which go from 1 to 5, being calculated as the ratings' pondered average of the issues of the product, as shown in the simplified equation (1):

$$(1)\ R_{b,s} = \frac{(rt_{s,qual} * w_{b,qual}) + (rt_{s,price} * w_{b,price})}{(w_{b,qual} + w_{b,price})}$$

The weight w is chosen randomly to each buyer in the beginning of the simulation, and remains unchanged during the whole simulation.

The Rating Issue formula specific in this domain are:
Quality Issue: (4) $rt_{s,qual} = fv_{s,qual}$ **Price Issue:** (5) $rt_{s,price} = 6 - (fv_{s,price} / 10)$

where $\mathbf{fv}_{s,price}$ represents the final selling price, and $\mathbf{fv}_{s,qual}$ the final quality value.

Each buyer initiates the simulation with 5000,00 in cash, which are updated at the end of each transaction, as shown in equations 6 and 7:

$$(6)\ cb_t = cb_{t-1} - fv_{s,price} + rp$$

where ***cb*** represents the buyer's cash (cb_t the current round cash and cb_{t-1} the previous round cash) and ***rp*** the resale price. It was defined that, in test rounds, the buyers should resell the acquired products, with the resale price being determined by the quality of the acquired product: (7) $rp = fv_{s,qual} * 10$

The resale is lucrative every time the buyer transacts with a seller who has good behavior, and prejudicial every time he transacts with a seller who has bad behavior. With this premise, there is a performance comparison between the two tested reputation models. The buyers are separated in two groups: the ones who select sellers based on the given recommendations according to the proposed reputation model (ACF1); and the ones who select their partners based on the recommendations generated by the traditional CF technique (ACF). At the end of each round the average cash value is collected between the buyers by group, and after the closing of the simulation, the distribution of averages by round can be analyzed so as to verify if the proposed method is better than the traditional one and if this difference is statistically significant.

3.1 Performed tests

Several test scenarios were anticipated in the simulation, representing different proportions of buyers according to their behavior. We applied statistical analysis in all of the scenarios (the “t” test, the unilateral, for independent samples, using significance level α of 5%). Each sample refers to one of the buyer’s groups, and is formed by the distribution of averages of cash per test round. We tested the following hypothesis:

H_0 – The performance of the proposed reputation model (called ACF1) is like the one in the model that uses the traditional CF technique (called ACF) in scenarios where buyers will change their behavior from one rating to another, noted as:

$$H_0: \mu_{acf1} = \mu_{acf}$$

H_1 - The performance of the proposed recommendation method tends to be better than the originally proposed CF method, in scenarios where the buyers will change their behavior from one rating to another:

$$H_1: \mu_{acf1} > \mu_{acf}$$

The performance of the proposed model (ACF1) was superior to the traditional model (CF) in every test, being statistically significant in 5 of the 7 tested scenarios. The average of buyer’s cash who used ACF1 was superior after the execution rounds on every test scenario executed. We present a summary of the results in table 6 and in Fig. 2.

Table 6. Summary of test results

	Summary of results						
	Scenario I	Scenario II	Scenario III	Scenario IV	Scenario V	Scenario VI	Scenario VII
Bad	40%	50%	50%	55%	60%	65%	75%
Good	50%	40%	50%	35%	30%	25%	25%
Neutral	10%	10%	0%	10%	10%	10%	0%
Best performance	ACF1	ACF1	ACF1	ACF1	ACF1	ACF1	ACF1
Significance probability	0.005	0.004	0.007	0.008	0.018	0.057	0.139

As the number of Bad buyers increase, the significance probability decreases. Even with the ACF1 model keeping better performance in relation to ACF, the difference between both is no longer statistically significant as the proportion of Bad Buyers is higher or equal to 65%.

Due to space constraints, we show only one of the comparative performance graphics of the test scenarios.

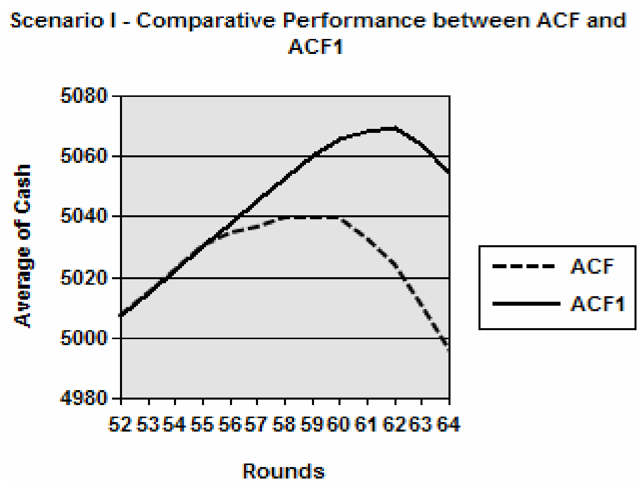

Fig. 2. 40% Bad, 50% Good and 10% Neutral Scenario

4 Conclusions

In this paper, we propose a model so that personalized reputation ratings based on CF can work adequately in case of changing behavior from the rated participants. The good results obtained in described simulations allow us to continue this work, performing field experiments so as to ratify the preliminarily results obtained.

5 References

1. C. Dellarocas, Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior, in: Proceedings of the Conference on Electronic Commerce, (Minneapolis, Minnesota, 2000), pp. 150-157.
2. J.L. Herlocker, J.A. Konstan, A. Borchers and J. Riedl, An algorithm framework for performing collaborative filtering, in: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, (1999).
3. P. Resnick, R. Zeckhauser, E. Friedman and K. Kuwabara, Reputation Systems, *COMMUNICATIONS OF THE ACM*, **43**(12), 45-48 (2000).
4. J. Sabater and C. Sierra, Reputation and Social Network Analysis in Multi-Agent Systems, in: AAMAS´02, (Bologna, Italy, 2002), pp. 475-482.
5. T.D. Huynh, N.R. Jennings and N.R. Shadbolt, On Handling Inaccurate Witness Reports, in: 8th Int. Workshop on Trust in Agent Societies, (Utrecht, The Netherlands, 2005), pp. 63-77.
6. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, GroupLens: An Open Architecture for Collaborative Filtering of Netnews, in: CSCW, (1994), pp. 175-186.
7. R. Torres, *Personalização na Internet* (Novatec Editora, São Paulo, 2004).
8. S.T.K. Lam and J. Riedl, Shilling Recommender Systems for Fun and Profit, in: Proceedings of the 13th international conference on World Wide Web, (2004), pp. 393-402.
9. J. Sabater. Trust and reputation for agent societies *Institut d'Investigació en Intel.ligència Artificial*, Consell Superior d'Investigacions Científiques, Catalonia, Spain. (2003).
10. J. Sabater, Evaluating the ReGreT system, Applied Artificial Intelligence, **18**(9/10), 797-813 (2004).