Situation Awareness: Dealing with Vague Context

Christos B. Anagnostopoulos, Yiorgos Ntarladimas, Stathes Hadjiefthymiades
Pervasive Computing Research Group, Communication Networks Laboratory,
Department of Informatics and Telecommunications,
University of Athens, Panepistimiopolis, Ilissia, 15784, Athens, Greece,
{bleu, yiorgos, shadj}@di.uoa.gr

Abstract—**Situation awareness is considered as an approach to model user context in ubiquitous information services. Such services can intelligently infer knowledge about user situation. Moreover, situational reasoning is attained taking into consideration similarity-based approaches. This paper proposes approximate decision-making about situational similarity using conceptual modelling, ontological representation, and, fuzzy logic inference.**

Keywords—**Situation awareness, conceptual modeling, ontological engineering, context reasoning, contextual similarity, fuzzy inference.**

I. Introduction

Context awareness is claimed to manipulate context related to certain entities. Moreover, situation awareness is considered as the particular favor of context awareness, where situations are viewed as logically interrelated contexts. Furthermore, modeling and describing situations as concepts in science-oriented ontologies, as well as, reasoning about situational context may lead to a specific kind of situation classification – *subsumption*. Specifically, the most similar user situation, of a situations ontology, with that of the current situation could approximately be interpreted as the most promising (i.e., relevant) in terms of subsumption.

This type of classification infers those situations that an entity is involved in and the degree of such *involvement*. What is proposed is an approximate reasoning procedure in order to infer knowledge about rather similar and compatible situations. Actually, the proposed system attempts to extensively compare a given situation with the situations of a specific ontology. Such system selects the most relevant situations. In addition, we refer to how the proposed system takes into consideration the degree of the user's situational involvement in a Pervasive Computing Environment (PCE). Actually, the system attempts to react to the current situation that a user may be involved in, by firing several rules that are predefined into the corresponding user profile. Such rules trigger certain actions given the current user situation.

It is known that many real context aware applications require support for managing imprecise context. Specifically, in a PCE, contextual information is rather vague and cannot always be retrieved. However, such vague information may lead to an inexact contextual reasoning. Several methods have been proposed and applied to deal with vague contextual information [1]. Moreover, vague contextual information implies vague situation modeling and, hence, approximate reasoning over such situations. In addition, such kind of reasoning produces approximate knowledge about the user situational involvement. The proposed system deals with such vague knowledge through fuzzy inference rules. Such rules not only cope with the imprecise knowledge about situational involvement, but also, with the user behavior and his/her historical context. The latter contexts support the proposed system with specific semantics in order to defuzzify the knowledge about the user situation involvements.

The rest of this paper is organized as follows: In Section II, we describe a specific scenario depicting the approach of the situation awareness in a PCE. Section III studies how to use conceptual modeling representations in order to model situational context, and in Section IV, we represent contextual similarity measures that are aware of context semantics. Section V focuses on reasoning about contextual similarity based on semantics and specific relations among diverse situational contexts. Section VI focuses on several fuzzy inference rules, which allow the proposed system to make decisions with respect to the user situations. In Section VII, we evaluate the proposed system. Section VIII discusses related work on that research area and, finally, Section IX concludes the article.

II. Situation Awareness

Situation aware applications [2] describe a new class of context-aware applications that are capable of recognizing a user situation. Moreover, they exploit the situational user context in order to provide him/her with appropriate information, or to perform certain tasks, in a pervasive way. Such kind of applications adapt themselves to a current, and maybe future, user situation context, hence, they react in a situation dependent manner.

In order to deal with situational context, we distinguish two types of rules that the proposed system reasons about situations; the *situation determination* rules and the *action determination* rules. The first type of rules refers to the conclusion (i.e., subsumption) about the class of compatible situations a PCE user may be involved in. Specifically, such rules combine contextual information from several knowledge resources about the user context, like user location, time, agenda entries, mobile terminal context, and the proximity to other people. The form of such rules could be the expressed as the follows:

$$\wedge_i context(x_i, user) \rightarrow isInvolvedIn(user, situation) \qquad \text{Rule.1}$$

Such expression denotes that, whenever the logical conjunction of the user contextual information x_i (i.e., *context*(x_i, *user*)) holds, then, it is implied that he/she be involved in a certain situation. Such contextual information may be imprecise due to limited resources or inexact contexts. Hence, we can approximately reason about the current user situation. Specifically, the more contextual information the system manages, the more accurate the results of the situational reasoning process will be.

Consider, for instance, that Alice and Bob are both company executives, located in a room of the company building (e.g., possibly not a meeting room) during the time of the meeting. Moreover, Bob's electronic agenda application, which is running on his PDA, contains a record related to an internal partner meeting at the same time. Obviously, we can conclude that they have a formal meeting, even though there is no exact information about the room they are located in, and of what kind of other persons are possibly in the same location (e.g., business executive, manager). Actually, the reasoner could not infer that they have a formal meeting, because it is not certain that Bob and Alice are the only company executives in that location. In addition, there is no information about the number and kind of other people that are located in this space at the same time. Furthermore, the reasoner could not even conclude that they have a meeting because the room is not necessarily a meeting room.

The uncertainty, which arises from the logical comparison of the current situations of Bob and Alice with those of the reasoner, can be interpreted as a similarity measure between them. Specifically, the situations could be modeled as concepts in taxonomies. Hence, one can calculate their similarity, in a quantitative way, and can reason about it.

The second type of rules associates each situation with a set of certain actions that can take place in a ubiquitous manner. Specifically, a user could assign to a context-aware application in a PCE a task related to his/her current situation. The form of such rules could be expressed through the following statement.

$$\begin{aligned} &isInvolved\ In(user, situation) \rightarrow \\ &do(withOption\ (task, option), situation) \end{aligned} \qquad \text{Rule. 2}$$

where, *task* is the corresponding task with the specific *situation* and *option* is the task application mode, as it will be described in the following paragraphs. One could denote that Rule 2 is rather similar with the expression defined in Situation Calculus [3]. Whenever the context-aware application infers (with a degree of belief) that the user is actually involved in a certain situation and his/her profile defines a certain task for that situation, then, the application performs that task. The degree of such belief leads to diverse options of the task execution. Suppose that Bob is about to attend a very formal meeting (i.e., a meeting with only business executives and managers). In such case, he should not be interrupted by insignificant e-mails during the meeting. Instead, he desires to be informed only on urgent matters. We assume that all senders in his address book have been marked beforehand with an 'S', which stands for significant, or, 'I', which, respectively, stands for insignificant senders, as depicted in Table I. Hence, according to this scenario, the most important e-mails are forwarded to Bob's PDA, for instance, those which are related to his work (e.g., the minutes of the meeting, or other crucial files). In this case, the addresses of all business executives have been marked with an 'S'. Whenever a situation reasoner infers Bob's situation, i.e., attendance of a formal meeting, then it can fire the specific action determination rule related to the '`Forward significant e-mail`' task. On the other hand, whenever such reasoner is not quite certain about Bob's situation, then, it has to deal with that uncertainty. This could be achieved either by notifying Bob how to act, or by taking no action. Finally, if the reasoner is sure that Bob is not involved in such a situation, then, it actually takes no action.

There are three options related to the process, which is managing Bob's e-mails during a meeting: '`take no action`', '`take action`', and '`notify`'. The first one does not proceed with any activity if a given event occurs (e.g., a less important email just arrived). According to the second option, once a specific incident occurs (e.g., a business executive's e-mail arrives), the process has to perform the certain task '`Forward significant e-mail`'. Such activity denotes that the system pervasively performs the desired task without further user interruption. The third option just reports that a specific event occurred. For instance, an e-mail from someone, which is not included in the address book, arrives, or the reasoner is not certain about Bob's situation. In this case, Bob's situation is rather ambiguous and, hence, the system cannot infer. What is proposed is notify the user about such incident, for instance, through a message display. This policy ensures that Bob will not miss an important message, if the e-mail is actually urgent and he will be less annoying, in case of another meaningless interruption.

TABLE I. SYSTEM DECISIONS

E-mail address marked as	System Option	Bob's reaction	System Revision
'I'	Take no action	–	–
'I'	Take no action	Take action	Penalty
'S'	Take action – Forward significant e-mail	–	–
'Unknown'	Notification	Accept Notification	–
'Unknown'	Notification	Reject Notification (Bob is interrupted)	Penalty

The main purpose is how Bob's reaction and behaviour could be exploited to render the system more accurate and pervasive with respect to its actions. Consider the fact that an e-mail arrives at Bob's PDA but Bob decides to reject it. Despite the fact that the e-mail was regarded by the system as important, it turned out to be insignificant for Bob. In this case, the system has to mark this e-mail as 'I' and never forward a future e-mail of the same sender, while Bob is still involved in the previous situation. Table I depicts the system options and future actions (revisions) related to Bob's situational context and behaviour.

The proposed system deals with vague context knowledge related to user situations, and attempts to reason about uncertainty whenever contextual information is imprecise. Specifically, we define the user situational involvement as a degree of involvement, namely d_{INV}. Such degree denotes the level of a user involvement in a certain situation. It partially affects the reasoner in order to determine which one of the three options ('`take no action`', '`notification`', '`take action`') is most suitable for the current user situation. Additionally, whenever the reasoner is certain about a user situation, then, it triggers the system to take, the corresponding with that situation, action in a pervasive way. Lower degree of certainty (i.e., lower value of d_{INV}) leads the system to either notify the user about taking the corresponding action, or performing no action at all. We define degree of pervasiveness, d_{PER}, as the measure that denotes whether the system is capable of reasoning about user situation in order to take actions with, at least, the minimum number of user notifications, or interruptions in the worst case. In addition, such measure refers to user historical context (i.e., reaction). Throughout this paper, we experiment with reasoning about vague context over two system versions. The first one (S_1) is only aware of the vague situational context (i.e., takes into account only the d_{INV} measure) in order to determine the most suitable option for the task application; whilst, the second version of the system (S_2) takes into account both the d_{INV} and d_{PER} degrees.

III. SITUATION MODELING

Situation awareness [4, 5] may be regarded as a special kind of context awareness, which models situation logically interrelated concepts. However, conceptual modelling explores the syntactic perspective of concepts but also the semantic dimension. Taking into account the similarity among concepts, in taxonomy, produces a more spherical building of such taxonomy including the essential semantics. There is a variety of conceptual modeling techniques that provide different similarity measures and determine whether two concepts are semantically *similar*. In fact, the more semantic information a conceptual schema disposes, the more precise the similarity measurement becomes. Undoubtedly, semantics is the key factor for reasoning about the qualitative as well as the quantitative *similarity*, and *compatibility* of two concepts.

Specifically, we model situation context as set of concepts related to different ontologies. Such ontologies describe and interpret the specific contextual information associated with the user situation. Hence, the similarity measure between two situations, declared as concepts, implies the similarity measure among those concepts that describe such situations. Actually, the d_{INV} measure is defined as a function over the similarity measure between the current user situation, *Q*, and with an asserted situation definition in situation ontology, as that in Fig.1. Moreover, conceptual modeling not only deals with modeling contextual information as interrelated concepts, but, also, with defining certain semantics among such concepts. For instance, the *Meeting* situation in Fig.1 is considered as a more general concept (super-concept) than that of the *Formal Meeting* situation (sub-concept). Specifically, the latter defines that the *Formal Meeting* could only take place with persons that are either business executives or managers. It has to be noted that *taxonomy* is a set of concepts organised by the *is-a* relation (e.g., *formal meeting* is-a *meeting*). Hence, *Meeting* is told to subsume *Formal Meeting* forming situation taxonomy.

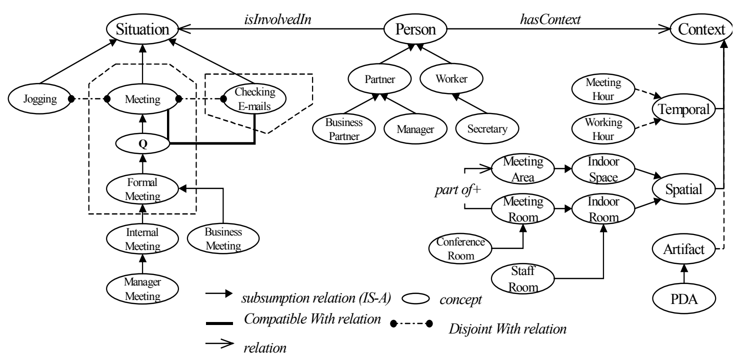

Fig. 1. Situations Ontology

It is well-known that RDF(S) [6] offers more expressivity in terms of semantic information modeling than RDF [6]. According to the latter knowledge representation, concepts and the relations between them are described using the *Subject–Predicate–Object* (*SPO*) scheme. On the contrary, RDF(S) gives the opportunity for richer conceptual representation, since it allows more semantics to be added in the conceptual taxonomy. Since we have to reason about situational context we need a more expressive language in terms of semantics and capable of conceptual subsumption. We make use of the OWL-DL [7] ontology language in order to build the situation ontology, which supports concept classification and reasoning through the SWRL language [8]. Apparently, given two concepts that belong in taxonomy, one can calculate the similarity between them, taking into account their relative taxonomic position. Apart from the classic taxonomy, OWL-DL can support a set of relations among concepts. For instance, a *Formal Meeting* is taking place at a specific meeting room (corresponds to a location relation), on a certain day and for predefined duration (corresponds to temporal relations). Moreover, the attendee of such kind of meeting could be company executives and managers (corresponds to personal relations). Actually, conceptual modeling deals not only with conceptual taxonomies and relations but, also, with more special features, such as mereotopological (e.g., part-of relations) or temporal relations (e.g., discernible intervals). Such kinds of semantics are not further explained throughout this paper, but, are taken into consideration in order to compute the situational similarity.

From an ontological modeling perspective, situation can be modeled as interrelated concepts of diverse ontologies. Such ontologies refer to specific contextual information representing the situation context. We call such specific ontologies *local contexts*. Local contexts describe certain parts of the user situation context like, time, location, personal, and mobile device context. Specifically, such ontologies describe the following local contexts:

Spatial context: It maintains information about the place, in which a person acts (e.g., office, meeting room, staff room), the number of present persons that the place contains (e.g., alone, crowded), what personal contexts such persons have (e.g., room only for company members) and, sets of part-whole regional axioms derived from upper mereology ontologies [9, 10].

Temporal context: It refers to a person's relative time, such as meeting time, working time, lunch time, and to his/her absolute time, such as morning, evening, night.

Artifact context: It represents the context of the user's computational entity, application, or mobile device such as PDA, Communicator. Such context could be the type of the running application (e.g., e-mails reader), its situation (e.g., idle, downloading files), or the profile of his/her mobile device (e.g., multimedia capabilities, memory capacity.

Personal context: It denotes the roles that a person may have in a certain situation. Specifically, such roles could be worker, manager, company or business executive, and secretary. Moreover, personal context refers to personal electronic agenda entries, such as scheduled meetings, dates, as well as, user preferences.

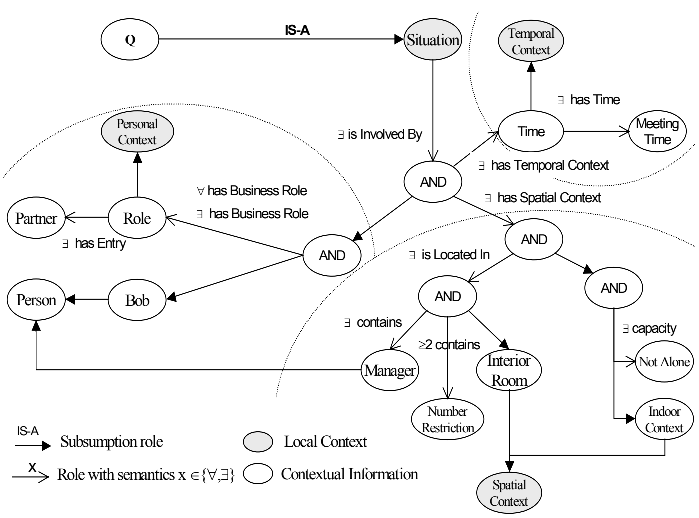

Fig. 2. Semantic Graph representation for the Q situation

TABLE II. DL SYNTAX FOR Q AND FORMAL MEETING SITUATIONS

Q ⊆ **Situation** ⊓ (∃ is Involved By. (Bob ⊓ ∃ has Time. Meeting Hour ⊓ ∃ is Located In. (Interior Room ⊓ ∃ contains. Manager) ⊓ ∃ has Business Role. Partner ⊓ ∀ has Business Role. Business Partner))
Formal Meeting ⊆ **Meeting** ⊓ (∃ is Involved By. (Partner ⊓ ∃ has Time. Meeting Hour ⊓ ∃ is Located In. (Meeting Room ⊓ ∃ contains. Manager ⊓ ∃ contains. Business Partner) ⊓ ∃ has Business Role. Partner ⊓ ∀ has Business Role. Business Partner))

In Table II, we model the current situation of Bob (Q) and the *Formal Meeting* situation using the Description Logic (DL) syntax. One could denote the diverse semantics over such situation descriptions, such as the subsumption (IS-A) relation (⊆), and the existential (∃) and quantificational (∀) restrictions over the relations. Moreover, situation could be envisaged as a semantic graph which is related to concepts, as depicted in Fig.2. In this figure, Q situation is a concept related to diverse local contexts. Two situations are similar whenever their corresponding local contexts appear similar, as we discuss in the following section.

IV. MEASURING SITUATIONAL SIMILARITY

The similarity between concepts could be defined as the function *sim*(. , .) that maps two concepts to the interval [0, 1]. The value of 0 denotes that the two concepts-arguments are strictly not similar, whilst, the value of 1 indicates that both concepts are equivalent. Hence, one can consider that, similarity between situations, is a weighted sum of the similarity of their local contexts as parts (Level 0). Furthermore, local contexts can be further described as concepts that contain even more specific local contexts (Level 1), hence, more specific knowledge (e.g., a meeting room - *spatial context* - that contains only business executives - *personal context*). Those kind of more specific contexts may, also, include other much more specific local contexts (Level 2), and so on. Consequently, similarity of two situational contexts of Level n+1 depends on the similarity of their aggregated local contexts of Level n. In our case, Q situation is modeled as a concept that consists of four local contexts, namely the: Spatial (Level 2), Temporal (Level 0), Artifact (Level 1), and Personal context (Level 1). Let the similarity measure of situations Q and S of level n, be *sim*(n, Q, S), then, it is recursively calculated as:

$$sim(n+1,Q,S)=\sum_{j} w_j sim(n,Q_j,S_j) \qquad \text{Eq.1}$$

where, w_j is the corresponding weight for the local contexts Q_j and S_j, such that $w_j \in [0,1]$.

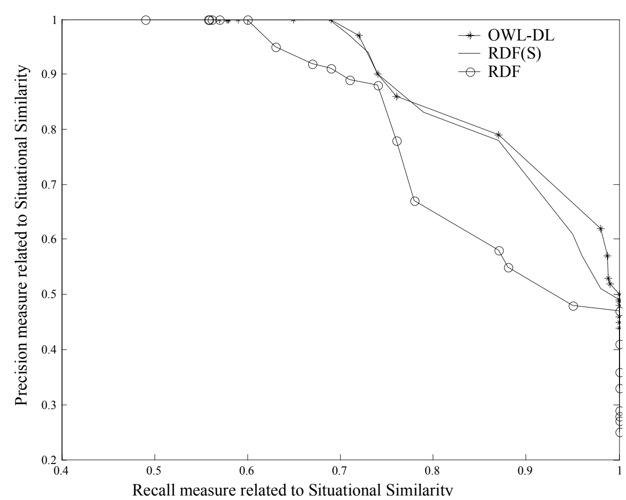

Fig. 3. Precision vs. Recall related to Situational Similarity Measure

There are a lot of techniques that calculate such types of similarity [11-14]. We use the similarity measure among ontological concepts as proposed in [12], which exploits the Tversky methods [13]. In order to evaluate contextual similarity we use the standard metrics of Precision and Recall [15], as depicted in Figure 3. Specifically, Recall is defined as the percentage of the retrieved and relevant situational contexts over the relevant situational context instances in the ontology. Precision is defined as the percentage of the retrieved and relevant situational context instances over the retrieved ones. Note that the similarity measure based on more expressive conceptual representations (e.g., OWL-DL) increases the value of Precision. On the other hand, RDF and RDF(S) representation schemes assume lower values of Precision.

V. Reasoning About Situational Similarity

In this section we describe how the proposed system selects the most relevant situations with the current user situation (denoted as Q). We assert every situation as *disjoint* with all others (i.e., they share no common features), and some are asserted *compatible* with others. The latter relation, that of compatibility between situations, means that a user can be involved in more than one situations at the same time. For instance, Bob can be involved in a *Meeting* and *Checking E-mails* situation concurrently, but not in undertaking physical exercise, like *Jogging*, which is regarded as incompatible with the two former situations. Moreover, Figure 1 depicts, also, the position of the classified situation Q among others, after the classification of the DL reasoner RACER [16]. Such reasoner deduces that Q is only a *Meeting* situation, and, thus, incompatible with the *Jogging* situation. However, there is no information about the degree of subsumption because, the reasoner provides only crisp results that either match or not. The result of the similarity measure is the set K containing <S_i ,*value*> tuples. The *value* is the degree of similarity, i.e., $sim(\text{n}, Q, S_i)$[1] and S_i is the candidate relevant and compatible situation with Q. In our case, K = {<*Meeting*, 0.6106>,<*Formal Meeting*, 0.8590>,<*Business Meeting*, 0.6200>}. One can conclude that, the most relevant situation is *Formal Meeting*, which is a meeting, something that verifies the results of the proposed mechanism.

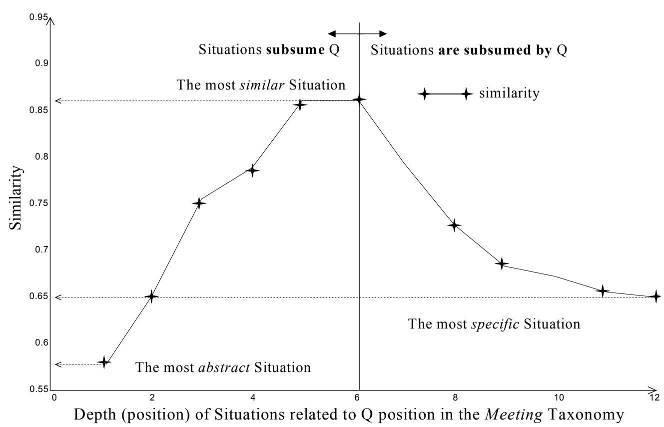

Fig. 4. Similarity Measure vs. Position of situations in the taxonomy

The similarity value related to the *Meeting* situation is not so high as any of its subsumed meeting situations. Specifically, Q is more relevant to *Formal Meeting* rather than *Meeting*. The latter situation is a more general description, thus, provides less information. In addition, Q is less relevant to the most specific situation, *Business Meeting*. This means that Q situation description refers to a more general concept than that of the *Business Meeting* situation. After simulations, the similarity value of any super-concept situation of Q, is lower than that of any sub-concept situation of Q. Figure 4 depicts how the similarity value of the classified Q is distributed along the depth of Q (i.e., the position in the taxonomy). Actually, the similarity value of the most specific situation, i.e., the deepest situation in the taxonomy, experimentally, never drops below the minimum similarity value, which is that of the most general situation (e.g., *Meeting* situation).

```
Procedure Reasoning about Compatible / Incompatible Situations
Input : Q, current situation
        K, asserted situations in the situation ontology
Output : W, set of compatible and relevant situations
Begin
  Set s_MAX the most relevant situation compared with Q
  /* that with the highest d_INV value*/
  Set H_MAX the situation taxonomy that contains s_MAX
  Set t_MAX the most general situation of H_MAX
  /* there is no other situation in H_MAX that subsumes t_MAX */
  Let K be the list of compatible and/or incompatible situations with s_MAX
  Sort (K), W= ∅
  /* K sorting is done with respect to the d_INV value */
  For each incompatible situation s_INC ∈ K Do
    If there exists s_INC such that:
     (t_MAX.d_INV ≤ s_INC. d_INV ≤ s_MAX . d_INV) Then
      K = K \ {s_INC}
    End If
  End For
      Notice: the compatible situations s_C : s_C ∈ H_MAX ∧ s_C ∈ K
              are considered necessary to be relevant,
      Hence: W = W ∪ {s_C}
   For each compatible situation s_C ∉ H_MAX Do
    If (t_MAX.d_INV ≤ s_C.d_INV ≤ s_MAX .d_INV) Then
      W = W ∪ {s_C}
    End If
   End For
  W=Sort(W)
  RETURN W
End.
```

Fig. 5. Similarity-based reasoning process for compatible situations

Such behavior appears in each situation that is compatible with Q. The reasoning process, shown in Figure 5, returns the most compatible and relevant situations related to Q. Incompatible situations with those that subsume Q, are considered irrelevant. This is attributed to the fact that, our proposed reasoning process returns only the situations that are relevant to Q. Such reasoning process is based on the Open World Assumption [29], not elaborated in this paper. After the application of the reasoning process, the set of the most compatible and relevant situations is W = {*Meeting*, *Formal Meeting*, *Checking E-Mails*}. The two former situations derived from the meeting taxonomy, and the later is a compatible with *Meeting* situation. Such situations are confined by the dash-lines in Figure 1.

A. Degree Of Situation Involvement

Let $a(\text{n}, Q, S_i)$ be the proportion of the similarity value of Q with S_i over the mean similarity values of Q with any situation S_j in the situations ontology, such that S_j belongs to W (see Reasoning Process in Section V) and is different from S_i, with n≥0. Such metric, defined in Equation 2, assumes values in

[1] Throughout the paper the n=2, denoting the maximum level of the situation local context.

[0,∝]. High values denote high involvement of a user in the situation S_i with respect to the rest of compatible and relevant situations. d_{INV} and a are defined in the following equations:

$$a(n,Q,S_i) = \frac{sim(n,Q,S_i)}{\frac{1}{|W|}\sum_{S_i \neq S_j \wedge S_j \in W} sim(n,Q,S_j)} \quad \text{Eq.2}$$

$$d_{INV}(n,Q,S_i) = \frac{a(n,Q,S_i)}{1 + a(n,Q,S_i)} \quad \text{Eq.3}$$

A zero value of d_{INV} means that, the reasoner is strictly not certain about the user involvement in S_i, while, a value of 1 denotes that the reasoner is certain about the user involvement in S_i. Hence, the situation determination rule could be written as:

$$\underset{i}{\wedge} context(x_i, user) \rightarrow isInvolvedIn(user, situation) \texttt{ with } d_{INV}$$

where, x_i stands for the contextual information related to the i[th] local context instance, such as spatial and temporal context.

According to the similarity-based reasoning, the situation decision rules for the situation, of Bob are expressed as follows:

$$\underset{i}{\wedge} context(x_i, Bob) \rightarrow isInvolvedIn(Bob, M) \texttt{ with } d_{INV} = 0.6106$$
$$\underset{i}{\wedge} context(x_i, Bob) \rightarrow isInvolvedIn(Bob, FM) \texttt{ with } d_{INV} = 0.8590$$
$$\underset{i}{\wedge} context(x_i, Bob) \rightarrow isInvolvedIn(Bob, CeM) \texttt{ with } d_{INV} = 0.5743$$

where *M*, *FM*, and *CeM* stand for *Meeting*, *Formal Meeting* and *Checking E-Mails* situations, respectively. According to the crisp reasoning and due to vague knowledge about Bob's context, the situation determination rule for the same case is as follows:

$$\underset{i}{\wedge} context(x_i, Bob) \rightarrow isInvolvedIn(Bob, M)$$

The crisp reasoning process either accepts that Bob is involved in a *Meeting* situation or not. Instead, the similarity-based reasoning defines a ratio that denotes in what level the system is certain about Bob's involvement in compatible situations.

The second type of rules deals with how the system can trigger certain tasks with respect three specific options (i.e., `'take no action'`, `'take action'`, `'notification'`).

One could define several thresholds for the above three options (e.g., similarly to Figure 6). According to the set of thresholds, if the d_{INV} is below the value of 0.65 then the task `'Forward significant e-mail'` is not performed. Moreover, if d_{INV} lies between the value of 0.65 and 0.85, then, the corresponding task is not carried out until a positive answer is given by the user. Finally, the corresponding task is performed whenever the system is rather certain (i.e., any d_{INV} value over the value of 0.85 signals such certainty).

$$isInvolvedIn(user, situation) \rightarrow$$
$$do(withOption(task, \text{'}Take\ No\ Action\text{'}), situation) \texttt{ if } d_{INV} \in [0,\ 0.65]$$

$$sInvolvedIn(user, situation) \rightarrow$$
$$do(withOption(task, \text{'}Notification\text{'}), situation) \texttt{ if } d_{INV} \in [0.65,\ 0.85]$$

$$sInvolvedIn(user, situation) \rightarrow$$
$$do(withOption(task, \text{'}Take\ Action\text{'}), situation) \texttt{ if } d_{INV} \in [0.85,\ 1.0]$$
$$\texttt{where } task = \texttt{'Forward siginficant e-mail'}$$

Fig. 6. The mode of the task execution is based on certain thresholds

However, the S_1 system is implemented to make decisions about the situational context based on such set of thresholds. Such system is not aware of the user behavior and responses whenever the d_{IND} lies between the 'notification' option boundaries (e.g., from 0.65 to 0.85). This means that, whenever the reasoner is not adequately certain about the user's situation, then, it performs the corresponding task after having notified the user. This is not so acceptable or convenient for the user, especially in a PCE, since even though S_1 is aware of the user situational context, it disregards the user reactions. Consider the fact that, the user is always notified for a task application, because the context-aware application could not accurately capture the user context, hence, it is more uncertain about the user's situation. Moreover, such crisp thresholds do not reflect the nature of the situation certainty. If the value of the d_{INV} is 0.84999, then the user has to be notified, even though it seems more reasonable for the system to perform the corresponding task. Fuzzy logic deals with the type of uncertainty, which arises when the boundaries of a class of objects are not sharply defined (e.g., the diverse options of the task application). The proposed system introduces Fuzzy Logic mechanisms in order to reason about situation similarity, as we discuss in following sections.

B. Degree Of Pervasiveness

The system has to be aware of the user situation context and the user reactions related to the `'notification'` option. Actually, in order for the system to act pervasively, it has to take decisions about performing a task with a specific option, in case the user is highly involved in the corresponding situation. In addition, such task performance could be done without the user notification or interruption, if such thing has been specified in his/her profile. The *uncertain* decision is raised whenever the system has to decide between the `'notification'` boundaries. Specifically, the system is supposed to act pervasively in high value of d_{INV} and to take no action in low value of d_{INV}. A medium value of d_{INV} could lead to a user notification. Once the system has dealt with the sharp boundaries of the options task application, it has to take into account the user reactions, too. Let the number T = A + B + C denote all the system decisions related to the three options for the application of a task, for a specific user involved in a certain situation. Specifically, A is the number of the system decisions, which refers to the `'take no action'` option for that task. In addition, B denotes the number of the system decisions that refers to the `'notification'` option, and, C is the number pertaining to the '`take action`' option. The ratio $p = \frac{B}{B+C}$ denotes the percentage of the user notifications / interruptions by the system over the total number of the system decisions related to either '`take action`', or to '`notification`' option. Hence, the higher the value of p is, the more times the system is either uncertain about the user's situation or disregards his/her past reactions, by interrupting him/her with '`notification`' messages. Consequently, a high value of p denotes a low degree of the system pervasiveness. However, the number A does not interpret that the system does not disturb user through notifications, which implies high degree of pervasiveness. Instead, the system implies that the user is believed not to be involved in a certain situation, hence, it decides to take no action.

Furthermore, the system ought to be aware of the user's reactions whenever the latter is interrupted by '`notification`' messages. In such case, we assume that in every notification message there is a '`do not disturb again`' choice. Hence, whenever a '`notification`' message arrives to the user display (see Figure 7), implying that the system is not certain enough about how to proceed, the user either accepts it, by replying according to the message suggestion (e.g., select *Yes*/*No*), or *rejects* it by choosing '`do not disturb again`'. In the case of rejection, the system records the user reaction and attempts to self-adapt to the user's response. Such kind of adaptation, results in reducing notification messages sent to the user, even though the system cannot decide on the corresponding task execution.

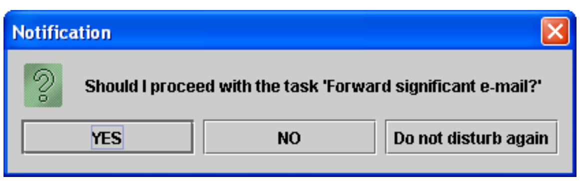

Fig. 7. The user notification message from the system

Let q be the proportion of the user *rejections* on the received '`notification`' messages over B: $q = \frac{\#\text{'do not disturb again' choices}}{B}$. A high value of q denotes a low degree of system pervasiveness, which means that, the system is not aware of the user's past reactions. We define as degree of pervasiveness, d_{PER}, the metric that relates to the weighted sum of the p and q proportions, as shown in the following equation:

$$d_{PER} = \frac{1}{1 + \beta . p + (1-\beta) . q} \qquad \text{Ex.4}$$

where $\beta \in [0,1]$ denotes the significance of p toward to q. d_{PER} assumes values in the range [0,1]. A value of 1 shows that the system acts in a pervasive way, while, a value of 0 indicates the contrary. The latter implication is explained as follows; (i), the user is notified whenever he/she is strongly believed to be involved in a certain situation, (ii), the system does not take into account previous reactions of the user, and, (iii), the system has rather vague knowledge about the current user situation. Note that, whenever the weighted sum of p and q is equal to the value of $a(n, Q, S_i)$, for certain current situation context Q and asserted compatible situation S_i, then, $d_{PER} + d_{INV} = 1$. In this case, the values of d_{PER} and d_{INV} describe a system, which is equivalently certain of the current user situation involvement, and, of the decision for the corresponding task execution.

VI Fuzzy Decision Making

System S_2 is aware of both degrees d_{INV} and d_{PER}, while S_1 is only aware of the d_{INV} degree. The former system revises d_{INV} in order to infer about how to proceed with a certain task execution. Specifically, it combines the two degrees and produces a holistic degree, namely these degrees of improved / enhanced situation involvement, d_{INVP}. The latter degree implies that a system is not only aware of the user's situation, but it also takes into consideration the user's past reactions to the system decisions (i.e., historical context). Moreover, d_{INVP} denotes the revised value of the d_{INV}, since the former determines a more sophisticated value of the latter degree.

We adopt Fuzzy Logic [17] to denote the fuzzy relation between d_{INVP} of two situational contexts, Q and S_i. In fact, we assign for both d_{INV} and d_{PER}, two *membership functions*, μ_{INV} and μ_{PER}, respectively, denoting for each metric the degree of belonging to a fuzzy set. Specifically, μ_{INV} maps d_{INV} into the range [0, 1], where 1 means full membership and 0 means no membership. μ_{PER} is defined, analogously, as illustrated in Figure 8. Let $R(d_{INV}, d_{PER})$ be a fuzzy relation representing the revised d_{INVP} between the two situational contexts. Such fuzzy relation reasons about the d_{INVP} of such contexts based on some degree to the d_{INV} of those contexts, and on some degree to the d_{PER} of the system behavior.

We also define the fuzzy sets of linguistic variables as $S_{INV} = \{high, medium, low\}$ related to d_{INV}, $S_{PER} = \{high, \text{medium}, low\}$ related to d_{PER}, and, $S_{INVP} = \{inactive, notifying, active\}$ related to d_{INVP}. *Low* d_{INV} denotes that the system lacks certainty about the user involvement in the situation S_i, thus, it decides to '`take no action`' for the corresponding task. *Medium* d_{INV} denotes that the system is somewhat certain about the user situation involvement in S_i and, it decides to notify the user about the task execution. *High* d_{INV} denotes that the system is certain about the user situation involvement in S_i and, then, it decides to '`take action`' without disturbing the user. Moreover, *high* d_{PER} implies that the system acts in a pervasive way (i.e., the system is capable of making decisions avoiding user disturbance), *medium* d_{PER} illustrates a system behavior, in which the user is notified for a task execution, and, *low* d_{PER} denotes that the system is unaware of the user's historical context and reactions.

Finally, the revised d_{INVP} is supposed to be aware of the user situational and historical context. Specifically, *active* d_{INVP} denotes that the system acts in a pervasive way, aware of the user context, while, *inactive* d_{INVP} denotes that the system is certain to '`take no action`'. *Notifying* d_{INVP} indicates that the system is certain neither to take action, nor, to take no action and believes that it should notify the user for a specific task execution. Obviously, d_{INVP} illustrates a more certain system with respect to decision making, since it is aware of both the user situational and historical context. To illustrate how the system applies fuzzy inference rules in order to reason about d_{INVP}, we show the five most crucial fuzzy rules in Figure 9. The system generates d_{INVP} by defuzzifying from the aggregation result, taking the centroid of the superimposed membership curve, as depicted in Figure 8, with respect to d_{INV} and d_{PER} values.

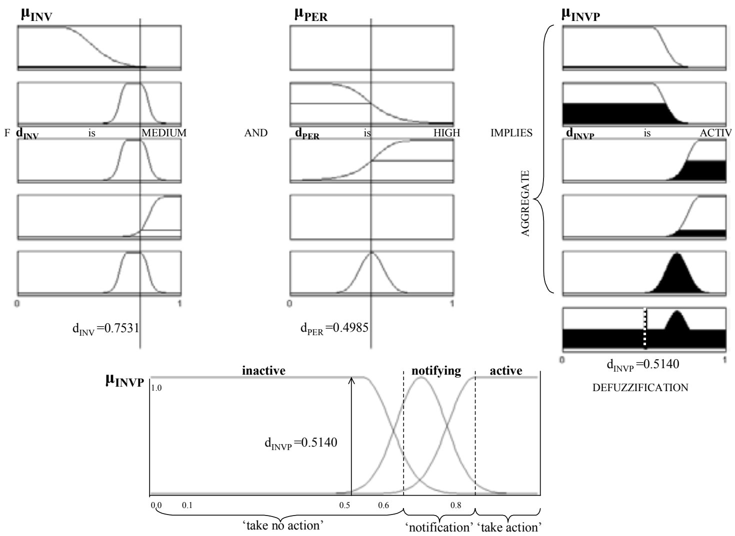

Fig. 8. Membership functions for d_{INV}, d_{PER}, and d_{INVP}

1. if d_{INV} is low then d_{INVP} is inactive
2. if d_{INV} is high then d_{INVP} is active
3. if d_{INV} is medium and d_{PER} is high then d_{INVP} is active
4. if d_{INV} is medium and d_{PER} is medium then d_{INVP} is notifying
5. if d_{INV} is medium and d_{PER} is low then d_{INVP} is inactive

Fig. 9. Fuzzy inference rules

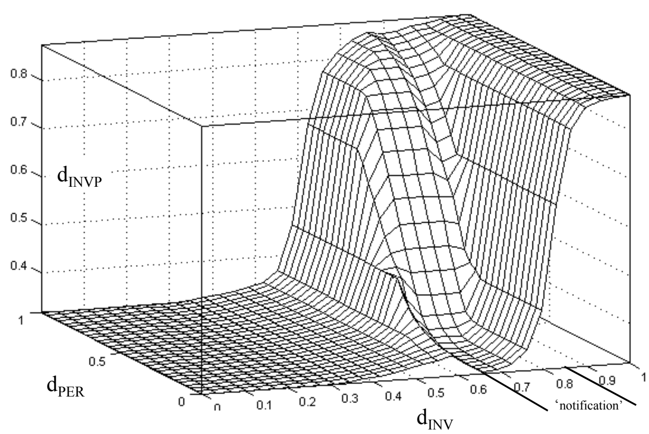

Fig.10. The dINVP behavior with respect to dINV, and dPER metrics

Figure 10 depicts the behavior of the d_{INVP} with respect to certain values of d_{INV} and d_{PER}. Specifically, note that when d_{PER} ranges from 0 to 1 and d_{INV} fluctuates between the value of 0.65 and 0.85 (i.e., the '`notification`' boundary), the d_{INVP} assumes a peak value. This means that, the system appears to be more certain about decision making over the three options for a specific task execution. In addition, it is obvious that, the number of B (notifications) has now noticeably decreased with respect to the A and C numbers. One can conclude that the system is capable of performing a specific task, or not, without having to disturb the user.

VII Experiment & System Evaluation

We have evaluated the two system versions S_1 and S_2, by using abstract and specific ontologies in order to model the local contexts for the user situations. Specifically, to avoid defining ontologies from the beginning, we choose to adopt upper ontologies to ensure generality and expressiveness for the four local contexts of the situation context (Section III). We use the Upper Cyc Ontology[2] from the IEEE SUO Working Group in order to model concepts, such as situation. We extend the Cyc spatial ontology by defining location concepts such as interior building areas, and meeting rooms. We, also, choose to adopt temporal concepts included in the DAML-Time / Time-Entry ontology[3] , import concepts related to user profile (agenda entries, preferences) included in the FOAF[4] and the GUMO [18] ontologies, and, model the artifact context by using terms from the FIPA[5] device ontology. In addition, our implementation consists of four reasoning engines in order to infer situational context. Specifically, we use rules for reasoning based on the DAML-Time axioms and Allen's [19] temporal interval calculus, rules for interpreting user spatial context constructed by the SEP mereologic operators [20], the RACER-DL inference engine [16] in order to reason about compatible situations, and, finally, the Fuzzy-JESS[6] rule engine in order to reason about the action determination rules (Section V).

S_1 makes decisions based only on the d_{INV} metric, while, S_2 makes decisions taking into account the fuzzy inference rules related to d_{INV} and d_{PER} metrics. We carried out 200 experiments, in which the user selected the '`do not disturb again`' option in a random way. Table III illustrates two decision metrics, in order to examine the behavior of systems S_1 and S_2. Specifically, we discuss two possible pieces of evidence. The first one concerns the S_1 decision to notify the user, whilst, the S_2 decides not to perform the task in order to avoid disturbing the user. According to Table III, S_2 turns out to be certain about deciding not to carry out that task. Actually, the number of the notification messages, sent by the S_1 to the user, appears to be increased by a percentage of 21.5%. In the second case, S_2 seems to act in a more pervasive way than S_1, since the former system takes action instead of notifying the user. Finally, in 64 cases (out of 200), the user does not receive a notification message from S_2. This implies that S_1 is less pervasive than S_2 by a proportion of 32%.

Figure 11 depicts the distribution of the d_{INV} values throughout several experiments as far as the S_1 system is concerned. Moreover, Figure 12 illustrates the corresponding

[2] http://www.opencyc.org/
[3] http://www.cs.rochester.edu/~ferguson/daml/
[4] http://xmlns.com/foaf/0.1/
[5] http://www.fipa.org/specs/fipa00086/
[6] http://herzberg.ca.sandia.gov/jess/

distribution of the d_{INVP} values, which refer to system S_2. One can observe the uniformity of the d_{INV} distribution over the three regions, which correspond to the '`take no action`', '`notification`', and '`take action`' options. On the other hand, the d_{INVP} distribution appears to be dense within areas related to '`take no action`' and '`take action`' options, resulting to a sparse '`notification`' region.

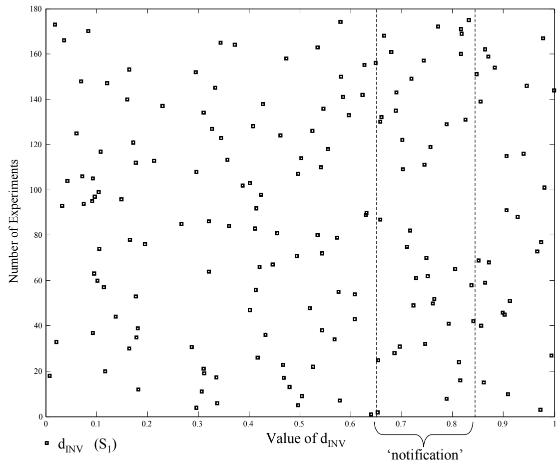

Fig.11. Distribution of the d_{INV} values

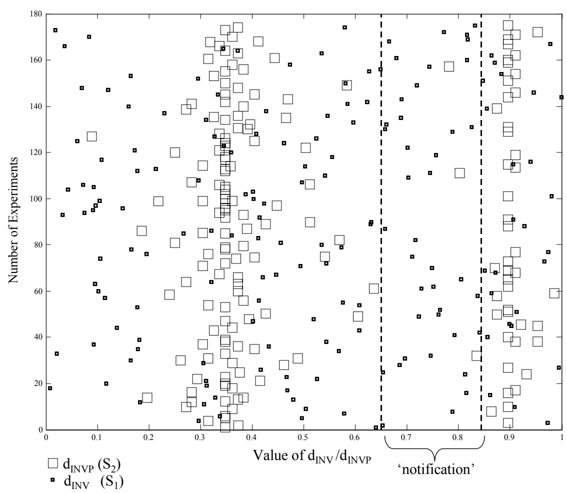

Fig.12. Distribution of the d_{INV} and d_{INVP} values

VIII RELATED WORK

Many situation models [21, 22] appear in the situation awareness literature. Certain models are capable of reasoning about situation knowledge. Using prepositional logics, the authors in [23] describe situations as concepts, consider the compatibility relations among situations, and, apply rules in order to infer the situation of an entity. The authors in [24] reason about consistent situations with respect to situation calculus axioms. In addition, in [25] the authors discussed about core ontologies representing situations, but with lack of enhanced semantics, thus, restricted knowledge reasoning. Such models focus on default reasoning that results in a crisp subsumption of unclassified situations. Furthermore, the authors in [26] modeled the user context as situations. They proposed a method to retrieve such situational knowledge by applying a dynamically logical matching method against system and user expectations related to current/future situations. Situation conceptual modeling has been, also, attempted by several information models, especially in the era of situation awareness and situation calculus as discussed by the authors in [27]. Significant work related to conceptual DL situational modeling and reasoning has been proposed by the authors in [28]. Finally, the authors in [30] deal with situational context recognition through data fusion techniques.

TABLE III. SYSTEM DECISION METRICS

	S1 Decision	S2 Decision	Number of occurrences (out of 200)	Percentage
	'Notification'	'Take no action'	43	21.5%
	'Notification'	'Take action'	21	10.5%
Total			64	32%

IX. CONCLUSION

It is well-known that knowledge related to situations cannot always lead to crisp subsumptions, due to its potential vagueness. In this paper, we propose user context modeling through a situation aware approach. Moreover, the developed system approximately infers the current user situation, since the user context is usually inexact or uncertain. Approximate reasoning over situational contexts is very important in order to measure such uncertainty. The proposed system deals with imprecise situation reasoning based on contextual similarity measures. We also define two metrics, d_{INV} and $d_{PER,}$ which refer to the degree of user situation involvement and system pervasiveness, respectively. Furthermore, the two versions of the proposed system, S_1 and S_2, use the aforementioned metrics in order to act as pervasively as possible. S_1 deals with the current user situation, whilst, S_2 extends the former system functionality by taking into account the user historical context. We adopted fuzzy inference rules to enable both systems to decide whether to perform a specific task based on the current user situation. Finally, S_1 and S_2 system behavior is evaluated through experiments. Experiment results demonstrate that S_2 behaves in a more pervasive way than S_1 in decision making, since the former takes into consideration the user feedback.

ACKNOWLEDGMENT

This project is co-funded by the European Social Fund and National Resources - (EPEAEK II) PYTHAGORAS. University
of Athens Research Project No. 70/3/7411.

REFERENCES

[1] F. Baader, R. Kusters, and R. Molitor. Rewriting concepts using terminologies. In Cohn G, Giunchinglia F, and Selman B, eds. 7th International Conference on Principles of Knowledge Representation and Reasoning (KR2000), pp:297-308, 2000.

[2] U. Meissen, S. Pfenningschmidt, A. Voisard, and T. Wahnfried. Context- and situation-awareness in information logistics. In W. Linder et al.,

editors, Current Trends in Database Technology-EDBT 2004, Heraklion, Greece, pp:335-344, 2004.

[3] R. Reiter. Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press, Cambridge, MA, 2001.

[4] E.Bass, J.Zenyuh, R.Small, S.Fortin,. A context-based approach to training situation awareness. In IEEE HICS,pp.529-551, 1995.

[5] S. Raymond, L. Sonenberg,. Situation Awareness in Intelligent Agents: Foundations for a Theory of Proactive Agent Behavior. IEEE IAT'04, pp. 86-92, 2004.

[6] D. Brickley, and Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Working Draft. http://www.w3.org/TR/PR-rdf-schema (visited April 2004).

[7] F. Baader, D. Calvanese, D. McGiuness , D. Nardi, and p. Patel-Schneider. The Description Logic Handbook – Theory, Implementation, and Applications. Cambridge University Press, 2003.

[8] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML version 0.5 November 19, 2003.

[9] P. Simons P. Parts, A Study in Ontology, Oxford, Clarendon Press, 1987.

[10] P. Speel, and P. Patel-Schneider A whole-part extension for description logics. In: Guarino N, Pribbenow S, and Vieu L (eds.) Workshop Notes of the ECAI-94 Workshop on Parts and Wholes, pp:111-121, 1994.

[11] A. Maedche,, V. Zacharias. Clustering Ontology-based Metadata in the Semantic Web, In Proc of PKDD 2002, pp. 348-360, 2002.

[12] M. Rodriguez, M. Egenhofer. Determining Semantic Similarity among Entity Classes from Different Ontologies, IEEE Trans. on Knowledge and Data Engineering, 15(2), pp. 442-456, 2003.

[13] A. Tversky, Features of Similarity, Psychological Review, pp.327-352, 1977.

[14] S. Santini, R. Jain, Similarity Measures, IEEE Trans. on Pattern Analysis and Machine Intelligence, 21(9),pp. 871-883, 1999.

[15] W. Frakes, and R. Baeza. Information Retrieval: Data Structures and Algorithms, Prentice Hall, 1992.

[16] V. Haarslev, and R. Moeller. RACER system description. In: International Joint conference on Automated Reasoning (IJCAR'01), pp:109-132, 2001.

[17] T. Hung, and E. Walker. A First Course in FUZZY LOGIC. CRC Press, Inc.1996.

[18] Heckmann D, Schwartz T, Brandherm B, Schmitz M, and Wilamowitz-Moellendorff M (2005) GUMO - The General User Model Ontology. In: Ardissono L, Brna P, and Mitrovic A (eds.) 10th International Conference on User Modelling (UM 2005), pp:428-437, 2005.

[19] F. Allen. Towards a general theory of action and time. Artificial Intelligence, 23:123-154, 1984.

[20] Hahn U, Schulz S, and Romacker M Part-whole reasoning: a case study in medical ontology engineering. IEEE Intelligent Systems & Their Applications 14(5):59-67, 1999.

[21] J.Crowley, P.Reigner, J. Coutaz, G. Rey. Perceptual Components for Context Aware Computing, In Proc of UBICOMP02, 2002.

[22] A. Kalyan, S. Gopalan. Hybrid Context Model based on Multilevel Situation Theory and Ontology for Contact Centers. In Proc of IEEE PerCom 2005.

[23] W.Dong,K. Xu, A Situation Calculus-based Approach To Model Ubiquitous Information Services, Computer Science AI, 2004.

[24] R. Michael, D. Funder. Situational similarity and behavioral consistency: Subjective, objective, variable-centered, and person-centered approaches, Journal of Research in Personality 38, pp.421-447, 2004.

[25] Matheus, M. Kokar, K. Baclawski. J. Letkowski, C. Call. The Practical Application of Semantic Web Technologies for Situation Awareness, In *Proc of* ISWC, 2004.

[26] Meissen U, Pfennigschmidt S, Voisard A, and Wahnfried T Matching System Knowledge and User Expectations in Situation-aware Systems. In: IEEE International Conference on Pervasive Services (ICPS'05), pp:271-278, 2005.

[27] Dong W and Xu K A Situation Calculus-based Approach To Model Ubiquitous Information Services, Computer Science, cs.AI/0311052, 2004.

[28] Mrohs B, Luther M, Vaidya R, Wagner M, Steglich S, Kellerer W, and Arbanowski S OWL-SF-A Distributed Semantic Service Framework, In: 1st Workshop on Context Awareness for Proactive Systems (CAPS 2005), pp:66-78, 2005.

[29] Grimm S, and Motik BClosed World Reasoning in the Semantic Web through Epistemic Operators. In: Proc of OWL: Experiences and Directions Workshop, 2005.

[30] Padovitz A, Loke S, Zaslavsky A, Burg B and Bartolini C, An approach to Data Fusion for Context-Awareness, in A. Dey, B. Kokinov, D. Leake, R. Turner (eds.) Fifth International Conference on Modeling and Using Context (CONTEXT) , LNAI 3554, pp. 353 – 367, 2005.