A Model to Compare and Manipulate Situations Represented as Semantically Labeled Graphs

Michał K. Szczerbak[1,2], Ahmed Bouabdallah[2],
François Toutain[1], and Jean-Marie Bonnin[2]

[1] Orange Labs, France Telecom R&D, Lannion, France
[2] Telecom Bretagne, Institut Mînes-Telecom, Cesson-Sévigné, France
{michal.szczerbak,francois.toutain}@orange.com,
{ahmed.bouabdallah,jm.bonnin}@telecom-bretagne.eu

Abstract. In our previous work we have introduced a novel social media that performs collaborative filtering on situations. This enhances user situation awareness with a collaborative effort to learn about importance of situations. In this paper we focus on defining a conceptual graph-based model used to represent situations in our system, so that it would (1) be consistent with existing formal definitions of situation, and (2) enable logical manipulations on situations, namely their detection and semantic generalization, which we employ in the system. In particular, we show how the latter can be accomplished thanks to situation lattices, which we adapt for the model.

Keywords: Situation awareness, situation theory, conceptual graphs, semantics, specialization / generalization, graph hierarchies, situation lattices.

1 Introduction

In the domain of interpersonal communication, we identify a potential in being able to communicate easily one's situation with one another. Users are already given web tools to exchange their availability statuses, location coordinates, moods, applications used, etc. And they use them willingly to share different pieces of information with whole groups of friends. However, we argue that enabling communicating one's complete and meaningful situations could result in more informed decisions on user interactions.

In [24] we introduce a context phonebook application to enable (1) sharing several context dimensions between contacts and (2) defining situations concerning those contacts that users wish to be notified of. We anticipate a stronger communication exchange need among close friends and family members. Therefore, we wish to assist in user situation awareness regarding their close ones. The KRAMER system employs collaborative filtering to suggest its users with notifications found to be important by others.

We model situations in our system with conceptual graphs [23]. Not only does this model make situation representations graphically pleasant and human-readable, but above all it enables reasoning on similarity of situations. Furthermore, we have tested

H.D. Pfeiffer et al. (Eds.): ICCS 2013, LNAI 7735, pp. 44–57, 2013.

logical manipulations on such semantically labeled graphs as KRAMER performs semantic generalization on situations it finds similar.

In this paper we elaborate a model of situations to use in our system. We justify it with a consistency with a theory of situation awareness [11] and the situation theory [3, 16]. We refer to several other works dealing with defining and modeling a situation.

Later, we place our model in a situation lattice, a hierarchical structure introduced in [28] for it maintains naturally the dependence relations between situations. We point out that Sowa's graph reasoning [23] can be seen as traversing such a lattice, which simplifies the process of situation generalization and specialization.

Finally, we focus on two main semantic operations implemented in the KRAMER system, namely situation detection and generalization of situation sets. We have already given the details, in particular of the latter in [25], but from the algorithm implementation point of view. In this paper we explain how do they employ the specialization / generalization reasoning inherited from conceptual graph reasoning.

The remainder of the paper is structured as follows. In Section 2, we gather the theoretical approaches towards defining what a situation is in a technical sense. In Section 3, we discuss different ways to model a situation present in the literature. We show that conceptual graphs are expressive in that manner and mix several good features of other models. Basing on this, we focus on the situation model used in the KRAMER system in Section 4. In Section 5, we explore the problem of comparing situations, as it is the basic problem in situation awareness. Afterwards, in Section 6, we discuss other logical operations that can be performed on situations in order to reason on them. We show how specialization and generalization operators can order situations in conceptual graph hierarchies. In Section 7, we explain how do we apply such hierarchies in our system to (1) detect situations and (2) perform semantic generalization on situations. We conclude and give future work directions in Section 8.

2 Theory of Situation (-Awareness)

A family of context-aware systems is very vast and rich. It gathers all systems that adapt their behaviors in function of changing context in general. However, context in its raw, low-level form is known to be often meaningless, trivial, uncertain and vulnerable to small changes [28]. Intelligent context-aware systems are more and more often interested in identifying situations as processed, more abstract context data, which provide a direct input to determine systems' adaptations and reactions.

As a result, situation-awareness is a property with a crucial impact on decision making and performance of both human and artificial system [11]. In fact, Endsley presents a model of human situation-awareness as ability to percept surrounding elements, comprehend their meaning and project their status into the near future. The author argues for that ability to require a much more advanced level of understanding than just being aware of numerous pieces of data.

The very same approach should be applied to artificially intelligent systems. In [28] situations are defined as semantic interpretations of context. They are more

abstract than low-level context and in turn they are also more stable, more certain, and, most of all, more meaningful to context-aware systems.

The effort to better grasp the concept of a situation has been made in the situation theory [3]. Situation theory is an interdisciplinary theory of meaning, which combines perspectives of philosophical discussion, mathematical rigor, and implementation practicality [16]. Indeed, Devlin states that it is not possible to define a situation in terms of familiar mathematical concepts, whereas they can be modeled as such [8]. The real situations are, therefore, distinct elementary abstract objects.

Having argued that, Devlin draws a line between those real situations and abstract situations individuated by agents. The latter are imprecise representations, models that one can create to reason about the real situations that he picked out [8]. In fact, this individualization represents only a part of the reality, as limited was situation awareness in [11]. And this common understanding of a situation to be a relevant subset of the state of the universe [9] is used in situation-aware researches either explicitly, e.g. [17], or implicitly, e.g. [2].

3 Situation Models

Situations learned by systems are limited to a part of what is really going on in real situations [8]. Therefore, they can be structured and modeled. In return, systems would be able to process them, comprehend them, and reason upon them. This would finally make such systems truly aware of situations with respect to Endlay's model in [11]. In this section we survey different situation models.

There are many ways scientists model situations for their needs. The simplest representation, as a straight forward attempt to capture real situations in a sense of situation theory, is by elementary situation concepts. Each recognizable situation has its corresponding semantic concept, like "meeting", "running", etc. Frequently they are related with one another with "is-a" unidirectional relations, forming a taxonomy. For example, "business meeting" is a particular type of a "meeting" in [2]. The same authors give an impression in their following paper [1] that their situation taxonomy can be treated as a taxonomy of activities ("checking e-mails", "meeting", etc.) quite distinct from concepts of an agent or its context.

Nevertheless, taxonomies require for all specialized concepts to be disjoint from one another and cover all cases of the super-concept, which can be too limiting in terms of measuring similarity between concepts [13]. Moreover, defining a comprehensive taxonomy for a complex domain of situations is merely impossible.

Therefore, authors of [2] seek expressivity in a model based on OWL-DL[1] ontology language. Ontologies support more relations between concepts. This enables modeling situations as interrelated concepts of diverse context taxonomies, namely spatial, temporal, artifact, and personal. In fact, a situation is said to involve a composition of such different concepts connected by "AND" logical operator. Existential and quantificational restrictions are also introduced in this model.

[1] http://www.w3.org/TR/owl-ref/

Having a set of context values is consistent with the definition of an abstract situation in situation theory. There, situation is a collection of infons that it supports [4]. Infons are the elementary informational items of a form $\ll R, a_1, \ldots, a_n, \{1,0\} \gg$, where R is n-ary relation of objects $\{a_i\}$. These pieces of information about different context dimensions are called characteristic features in [17]. The situation is, therefore, modeled as a set of such features in a given time interval.

Padovitz's Context Spaces [21] can be seen as a graphical representation of such composed situations. Each context type has its own dimension in the space and different values on scales are characteristic to different situations. As a result, situations are subspaces within the whole space. As such, they can be compared in terms of a geometric distance. Furthermore, one might see an equivalent of taxonomy relations as subsumption in space. The latter is harder for a vector representation of context in [7].

[27] presents another attempt to ontologically model situations. It is explicitly shown that a concept of a situation is on a different layer of the ontology than the concept of context. Any composite situation is a logical (conjunction, disjunction, negation) or temporal composition of atomic situations. The latter are further extended by three concepts: context type, boolean operator, and context value. This means that a situation can be seen as a combination of such triplets. Whereas those triplets are nothing else than infons restricted to binary relations.

Costa et al. notice further that context is only meaningfull with respect to an entity, whose concept is fundamentally different from the concept of context [5]. Therefore, context can be treated as a moment inhered in a substantial – an entity. As a result, a situation is a composition of such pairs of entities and their context. The authors introduce a graphical notation for situations involving different types of contexts and formal relations. This model is explained to be more applicable to context-awareness.

In consequence, we have investigated conceptual graphs. Such graph notations were developed at first to represent first-order logic and to create a mapping between queries in natural language and relational databases. In general, they are graphical representations of logical expressions, conjunctive first order logic formulas [20], and semantically-rich knowledge. If restricted to binary relations conceptual graphs become directed. Similarly to graphs in [5], they can be labeled with both entity and context value concepts.

Nodes and associated edges would form context type-value pairs, similar to [27], while multiple edges directed to one node would stand for "AND" operator, as in [2], matching multiple context moments with one entity [5]. Conceptual graphs can also represent if-then relations and negations, and might be extended with temporal relations. In the remainder of this paper we shall consider only conceptual graphs without cycles, namely conceptual trees. Their origin and definition are explained in the following section.

4 Situation Model in KRAMER

We start defining our model with a meta model inspired by a CONON upper ontology [26]. We distinguish, however, substantials (entities) from moments (context

description) as proposed in [5]. As a result, we define concepts of a context entity and its context state. Computation entity and person are subclasses of an entity. Location and activity are domain specific moments.

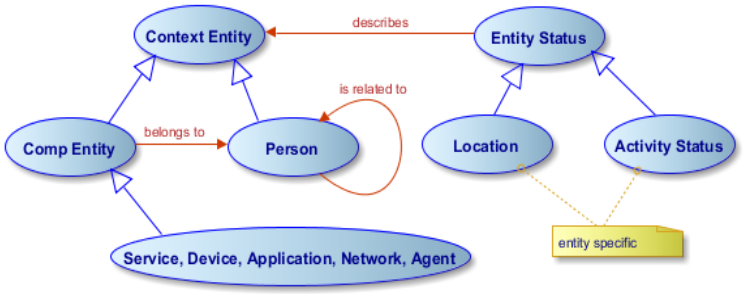

Fig. 1. Meta model of situation

We also introduce relations between moments and substantials, and between entities and a person to model the fact that one's situation is in fact one's context along with context of his or her close ones and his or her devices, services, etc. As a result, by instantiating concepts representing a situation and by inferring the respective meta-concepts relations, we receive a conceptual graph, a conceptual tree to be accurate. This conceptual graph has a "me" concept in its root. We say, that this complex situation exists with respect to one particular entity, a person who perceives the situation.

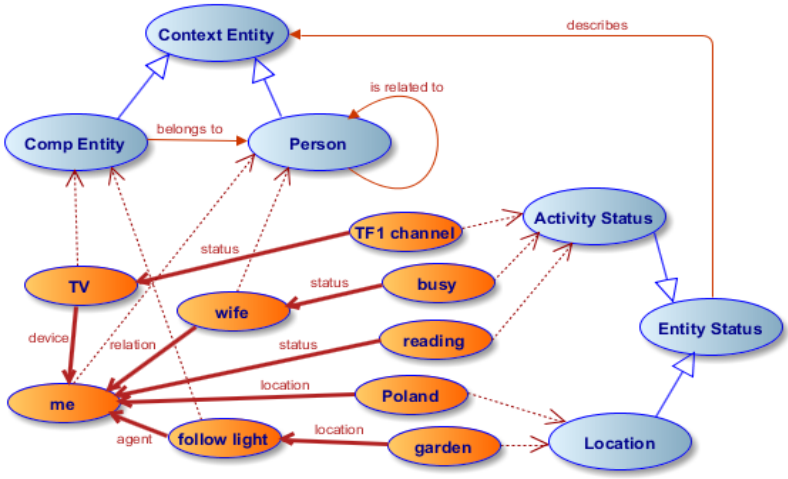

Fig. 2. Instanciating a conceptual graph from the meta model

Following the notation in [20] and [6] we define a situation conceptual graph (tree) *SCG* used in the KRAMER system along with its support. It should be noted that concept types are of four kinds (four concepts in the meta model) and relations are connecting either two entities or an entity with its context (red arrows in Fig. 1).

Definition 1. A support is a 5-tuple $S = (T_H, \{T_E\}, \{T_L\}, \{T_S\}, T_R)$, where:

- T_H is a finite, ordered set of human relations types $(T_H, \leq)$;
- $\{T_E\}$ is a set of finite, ordered sets of entity types $(T_E, \leq)$, e.g. services, devices, application, etc.;
- $\{T_L\}$ is a set of finite, ordered sets of location types $(T_L, \leq)$ specific the entity type;
- $\{T_S\}$ is a set of finite, ordered sets of status types $(T_S, \leq)$ specific the entity type;
- T_R is a finite set of binary relation types divided into two categories: those connecting entities to other entities $T_R^e = \{relation, device, service, agent, \dots\}$, and those connecting entities to statuses $T_R^S = \{status, location\}$.

Definition 2. A situation conceptual graph is a 3-tuple $SCG = [S, G, \lambda]$, where:

- $S = (T_H, \{T_E\}, \{T_L\}, \{T_S\}, T_R)$ is a support;
- $G = (V_C, V_R, E_G, l)$ is an ordered, directed graph having edges $E_G = (c_1, r, c_2)$:
  $\forall e \in E_G\ c_1^e \in T_H \cup \{T_E\},\ r^e \in T_R^e \Leftrightarrow c_2^e \in T_H \cup \{T_E\}, r^e \in T_R^S \Leftrightarrow c_2^e \in \{T_L\} \cup \{T_S\}$
  and meeting a condition: $\forall c \in T_H \cup \{T_E\}\ \exists e \in E_G\colon\ c_1^e = c \wedge c_2^e \in \{T_L\} \cup \{T_S\}$;
- λ is a labeling of the nodes of G with elements from support S:
  $\forall c \in V_C\ \lambda(c) \in T_H \cup \{T_E\} \cup \{T_L\} \cup \{T_S\}$; $\forall r \in V_R\ \lambda(r) \in T_R$.

Every concept in nodes of such conceptual graphs is a semantic concept taken from a respective taxonomy. Taxonomies model different context dimensions: human relations, types of devices, locations, etc. For example, Figure 3 presents a situation, for which being located in Poland is more relevant than being in any city in particular. As a result, these semantically labeled graphs become more expressive than situation taxonomies as presented in [1]. This will also enable logical manipulation presented further in this paper.

Fig. 3. Graphs are built of concepts from respective taxonomies

Moreover, our conceptual graph-based model is consistent with a definition of an abstract situation in situation theory [8]. Indeed, graphs represent only a part of the reality, of the real situation. In fact, every other entity taking marginal part in the

situation can be represented as “any” concept, extended further by “any” concept for its context. “Any” is a root concept for every context taxonomy used in our model and is omitted in a situation representation.

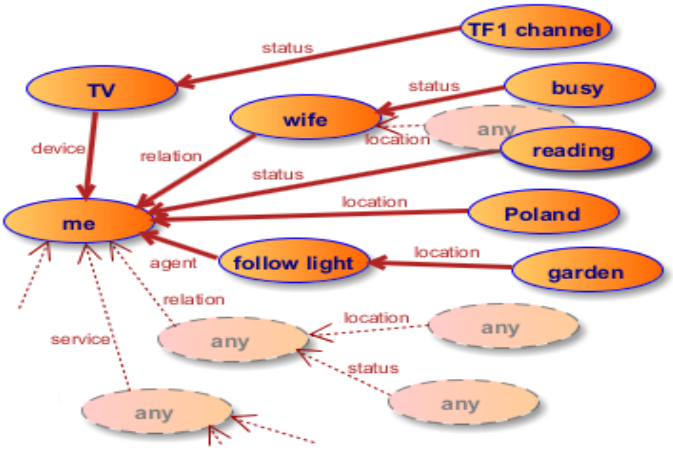

Fig. 4. Conceptual graphs model a part of a real situation

The motivation for us to select conceptual graphs as a model representing situations was its expressiveness, but also an easy comparison of conceptual graphs, which enables logical operations on situations, i.e. their generalization. We focus on those mechanisms in the following sections.

5 Comparing Situations

In order to reason about situations, understand them, agents need to be able to compare them with each other. They need to measure a degree of similarity between a current situation and their knowledge about situations, e.g. patterns. Therefore, a situation model should enable and facilitate this operation. In this section we show how situations can be compared in different models, i.e. conceptual graphs.

The first model discussed in Section 3 was the plain uni-concept representation of situations. In this case, similarity between two situations is measured as a similarity between two semantic concepts. Often, it is a distance measure between those concepts in an ontology. Gandon summarizes popular metrics in [13] and points out several open research questions. Basically, he states that strict mathematical distance on a static ontology is not necessarily the human way to reason about semantic closeness. Richer representations should be used to deal with a resolution error.

In that sense, having situations as a composition of several context dimensions could help. Scalar difference is a measure of distance between two points in Context Spaces theory [21]. Situation subspaces can also be compared, either by the distance measure or by the intersection operator, which finds if two subspaces have a common part. A context space makes detecting a situation extremely intuitive.

The introduction of context-operator-value triples [27] or substantial-moment pairs [5] combines the two preceding approaches. Comparing two situations requires measuring similarities of semantic concepts for each context dimension separately and calculating their weighted mean. The same principle applies to conceptual graphs that represent such multidimensional semantic spaces.

Even though optimal algorithms for matching graphs in general are reported to be exponential with respect to the number of nodes in either graph [14], we should remember that abstract situations do not represent the whole knowledge [8]. Instead, the number of nodes is limited to what is necessary for an agent to detect a situation. For instance in [17], "travelling" situation is defined only by using any transportation mean and by a fact of moving significantly.

Furthermore, Mugnier reports in [20] that many inter- conceptual graph operations become polynomial, should the involved graphs be restricted to trees. As shown in Section 4, situations are indeed considered to be represented by conceptual trees. In [29] one of the implementations of conceptual trees matching is reported to be polynomial. Furthermore, [6] gives an ontology similarity measure based on a projection between conceptual graphs, and [18, 19] present comparison of two conceptual graphs as a calculation of their overlapping parts with and without semantic subsumption.

6 Logical Manipulations on Situations

Most of the conceptual graph-based comparison algorithms mentioned in the previous section exploit the fact that concepts in nodes are structured in taxonomies per context dimension. As a result, "chasing an animal" is supposed to be matched with "chasing a mouse" [19], rather than "travelling by train", as concept of a "mouse" is a specialization of an "animal". On the other hand, in [22] the authors seek for the most interesting common generalization of two graphs in order to evaluate "thematic" similarity between two conceptual graphs.

In fact, according to [20], generalization and specialization are said to be the key computational notions in every reasoning concerning conceptual graphs. Sowa discusses 6 canonical formation rules as semantic graph-based operators for equivalence (copy, simplify), specialization (join, restrict) and generalization (detach, unrestrict) of conceptual graphs [23]. These operators can be interpreted by either logical subsumption or graph morphism. Just a negation operator needed to be added to handle full first-order logic.

Different researches make use of specialization rules, for instance [15] employs maximal join operator to perform high-level fusion on heterogeneous information represented by conceptual graphs. In our work [24], we are more interested in generalizing situations, and therefore generalizing associated with them conceptual graphs. We present the procedure and its motivation in Section 7.

Mugnier explains in [20] that for one graph to be a specialization of another, there needs to be a projection from the second graph to the first. Projection is a sequence of graph morphisms in a classical graph theory sense but implying equality of relation types and taxonomic specialization of concept types. As a result, a specialized graph

is a super-graph of the original one (external join operation) with possibly semantically narrowed labels (restrict operation).

This makes the specialization relation a preorder because it is not anti-symmetric as redundant graphs are still possible. Should the injectivity constraint be introduced and internal join operator forbidden, the relation becomes a full order. Therefore, conceptual graphs can form a hierarchy, like in [10]. As a result, reasoning about relation between two graphs can be transformed into a problem of traversing such hierarchy. One graph is a generalization of another, if it is an ancestor of that other graph.

Considering that conceptual graphs represent situations, reasoning about similarity of two situations is reduced in a way to semantic distance measures as presented in [13]. Moreover, finding more abstract / detailed situations implies traversing the hierarchy upwards or downwards. Ye et al. introduce this idea in a concept of situation lattices [28]. Although they model situations as simple unitary concepts, similarly to [2], they notice that this organization reflects the internal structure of situations and is beneficial in identifying situations.

We argue that situations modeled as conceptual graphs can naturally form situation lattices. Situation awareness would benefit from seeing a situation space as such order structures. In the following section we show how our system implements the explained situation model.

7 Situation Operations in KRAMER

In [24] we present an overview of our system, KRAMER. The ambition there is to empower users with spreading information about importance of situations, which is established in a collaborative manner. We model situations as conceptual trees, a special case for conceptual graphs (see Section 4). We say that situation is related, meaningful to an entity, the “me” concept in a tree root (see Fig. 2).

In previous sections we showed how this model is consistent with situation theory and how researchers have applied sophisticated relation structuring to ease reasoning about specialization / generalization of conceptual graphs. In this section we discuss (1) detecting complex situations and (2) generalizing them by the KRAMER system based on those approaches.

7.1 Detecting Situations in KRAMER

Our system enables users’ smart devices to sense the user context, share it among close phonebook contacts and fire programmed actions, i.e. notifications, should a set of conditions be fulfilled. This set of conditions concerns one’s context and the context of his or her close ones. Simply put, an action is fired once a required situation matches the current one. Therefore, we need an efficient multiple situation detection mechanism.

For this matter we use a Rete production system [12]. This choice is well motivated in associating productions (decisions to perform an action) with complex set of

conditions (situations). In order to introduce semantic reasoning, we enhanced our Rete implementation by replacing equality (=) condition with subsumption (≤) one in alpha network. Therefore, an enhanced Rete takes full advantage of situation lattices concept introduced in [28].

For example, wanting to be notified about a friend being in Poland requires a Rete condition `C: (<x> ^location Poland)`. Admitting that a friend shares that he is currently in Warsaw, known to be a capital of Poland and a descendant of a "Poland" concept on a location taxonomy, one would expect the condition to be matched. Indeed, a situation "a friend is in Warsaw" implies a situation "a friend is in Poland". The actual situation would be, therefore, a specialization of the situation expressed by condition C.

One might notice that a condition is a context triple representing an atomic situation [27], always a context relative to an entity as in [5]. Should the situations become more complex, a set of conditions is introduced to a production system. From a Rete network point of view, conditions are connected by join nodes in beta network (see Fig. 5). From situation model point of view, atomic situations are logically connected with AND logical operator as in [1]. Complex situations form therefore a conceptual tree.

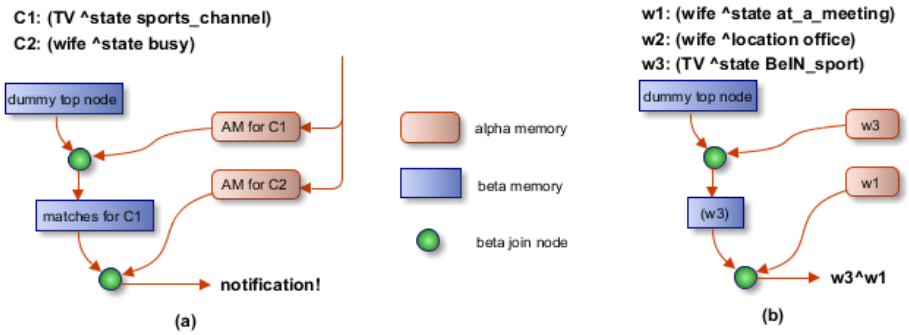

Fig. 5. Example Rete network, (a) structure, and (b) instantiation with working memory sets *w* for a situation "my wife is busy and my TV is on sports channel"

One might also notice that having redundant situations is not possible in the KRAMER system. Therefore, every possible situation for a given set of contacts in a phonebook forms a finite lattice. The supremum is the most abstract situation, "anything is going on". It stands for a trivial graph made of one node labeled "me". It might be extended to a full structure filled with "any" labels, see Fig. 4. The infimum would be therefore a set of most specialized full conceptual trees (see Fig. 6).

As a result, an abstract situation detecting problem can be transformed into determining whether a concrete current situation is a descendant of the first one. A situation the system needs to detect should be a generalization of the situation perceived. Therefore, there should exist a set of join (to merge multiple pieces of information coming from different sources of interest for a particular abstract situation) and unrestrict (to generalize current situations for particular context dimensions) operations from the complete situation to the abstract one.

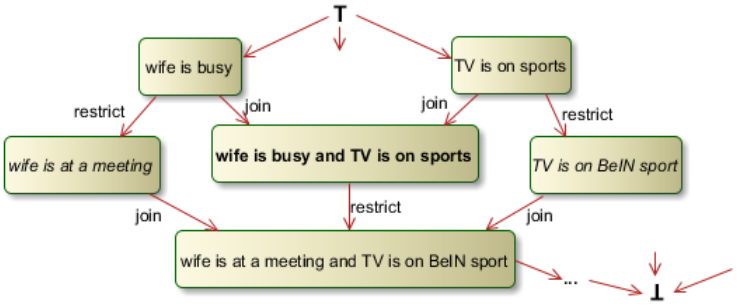

Fig. 6. Part of a situations lattice for the example in Fig. 5

Our Rete implementation performs a check for such set of operations. Firstly, it joins every atomic situation introduced separately to its network (in order to perform individual matching tests for each condition in its alpha network). Secondly, it introduces a semantic subsumption operator (restrict / unrestrict operation equivalent) rather than equality in Rete alpha network. As a result, it browses quickly a situation hierarchy [10, 28] to determine whether a current situation matches any of the situations it seeks for action launching.

7.2 Generalizing Situations in KRAMER

The main purpose of our system is to enable collaboration on situation awareness. Users are invited to define situations they wish to be notified of thanks to context sharing subsystem. Those situations are defined on contextual concepts and are not associated with any private data. Furthermore, every such situation is anonymously uploaded on a server that is supposed to evaluate each in terms of their use rating. As a result, the system suggests important situations to other users and reevaluates them accordingly to whether those users find them interesting or not. See [25] for details.

However, depending on the context granularity, it may be distorting to the situations rating if we treat two very similar situations as completely different ones. For example, a complex situation of having “a daughter leaving school and wife being busy, unable to take her home” might be essentially the same as having “a son leaving school and wife being busy, unable to take him home”. It could only depend on what gender child one has. Therefore, our system is able to generalize similar situations and merge their ratings [25].

Our algorithm has, however, a couple of restrictions in order for two situations represented as conceptual graphs to be generalized. The first is a requirement for the graph structures to match. The other, for the corresponding concepts on those graphs to be semantically similar. We shall explain how do those restrictions relate to the situation hierarchy from [10].

As discussed in the previous subsection, every possible situation in a system and all corresponding abstractions exist in a common situation lattice. Therefore, finding a least common ancestor by mean of generalization for any two situations is always possible. For example, generalizing “wife is at work” and “wife is busy” results in

trivial "wife (is anywhere and doing anything)". This does not make sense with respect to suggesting meaningful situations to system users. Therefore, our algorithm first groups situations by matching graph structures with respect to number of edges and their relation concept labels.

As a result, one might see this as an elimination of join operator from the situation hierarchy (like in Fig. 6) to create a family of smaller structures based on restrict operator only. The second step of the algorithm will be performed within the scope of each small hierarchy separately, see Fig. 7 for one example.

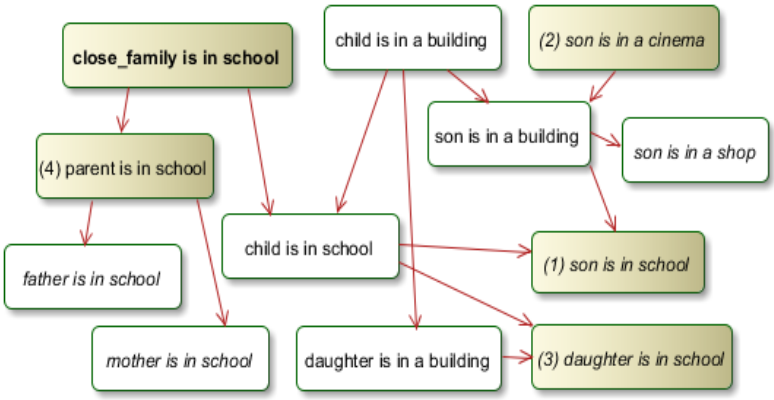

Fig. 7. Part of a hierarchy of situations with restrict-only relations

Having found two or more situations sharing the same graph structure, the algorithm proceeds to finding opportunistic generalizations of those that are similar. We define one requirement for a set of situations to be generalized into one: there cannot be a series of restrictions from the abstract situation to any leaf that does not pass through any of the situations in a set. One might say that the abstraction is the least common ancestor of situations in a hierarchy that covers all of those situations and none of the situations that does not subsume any of the situations in the set.

For example, let's consider the following set of situations: (1)"son is in school", (2)"son is in a cinema", (3)" daughter is in school", and (4)"parent is in school. A part of a restriction hierarchy for a situation matching scheme `<a person> is in <a location>` is presented in Fig. 7. As a result, a situation can be generalized from (1), (3) and (4) into "a member of a close family is in school". Meanwhile, the situation (2) remains not generalized, because there are many situation nodes missing if we were to generate an abstract situation "son is in a building", for example.

As a result, the KRAMER's algorithm performs two steps on a common situation lattice. First, it eliminates any join / detach operation connectors, which splices the structure into a family of hierarchies, each involving only one situation graph structure. Second, it scans all generalized situations (products of unrestrict operation) so that they are common for a subset of given situations while not having any restrict operation chain that would not lead to any given situation. This approach is found to be efficient from the computational complexity point of view in [25].

8 Conclusions

In the KRAMER system we perform two main semantic operations on situations: their detection and generalization. In this paper we present those operations as logical manipulations on situation hierarchies, lattices that constitute a space of all possible situations. We show that such hierarchies are a natural product of specialization / generalization relations between situations. To assure expressiveness of situation comparison we model situations as conceptual graphs. In addition, we discuss that model to be consistent with the situation theory.

As a result, we transform reasoning on situations similarity into a problem of traversing a conceptual graph hierarchy. This approach is less implementation centric, much more situation model driven. Nevertheless, it provides a set of straight-forward logical directives for our algorithms to implement.

For future works, we plan to investigate its further impact upon situation prediction and reasoning in situation uncertainty. It is very likely that a sensed situation is similar to the searched one but they are not the same in terms of subsumption relation. However, this may mean, for example, that an agent is unable to perceive some context dimensions necessary for detecting a situation, or that the searched situation would possibly appear in the near future. In either case, narrowing the situation hierarchy to the nearest neighbors with respect to join / detach operators, and discovery of their restrict / unrestrict operation products might result in a respective measure of probability.

References

1. Anagnostopoulos, C.B., Ntarladimas, Y., Hadjiefthymiades, S.: Reasoning about Situation Similarity. In: International IEEE Conference on Intelligent Systems, pp. 109–114 (2006)
2. Anagnostopoulos, C.B., Ntarladimas, Y., Hadjiefthymiades, S.: Situation Awareness: Dealing with Vague Context. In: ACS/IEEE International Conference on Pervasive Services, pp. 131–140 (2006)
3. Barwise, J., Perry, J.: Situations and Attitudes. Bradford Books, The MIT Press (1983) ISBN 0-262-02189-7
4. Cooper, R., Kamp, H.: Negation in Situation Semantics and Discourse Representation Theory. In: Situation Theory and Its Applications, vol. 2. Stanford University (1991)
5. Costa, P.D., Guizzardi, G., Almeida, J.P.A., Pires, L.F., van Sinderen, M.: Situations in Conceptual Modeling of Context. In: 10th IEEE International Enterprise Distributed Object Computing Conference Workshops, p. 6 (2006)
6. Croitoru, M., Hu, B., Dashmapatra, S., Lewis, P., Dupplaw, D., Xiao, L.: A Conceptual Graph Based Approach to Ontology Similarity Measure. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS 2007. LNCS (LNAI), vol. 4604, pp. 154–164. Springer, Heidelberg (2007)
7. Delaveau, L., Loulier, B., Matson, E.T., Dietz, E.: A vector-space retrieval system for contextual awareness. In: IEEE International Multi-Disciplinary Conference on Cognitive Metheods in Situation Awareness and Decision Support, pp. 162–165 (2012)
8. Devlin, K.J.: Situations as Mathematical Abstractions. In: Situation Theory and Its Applications, vol. 2. Stanford University (1991)
9. Dey, A.K.: Providing architectural support for building context-aware applications. PhD thesis, Georgia Institute of Technology (2000)

10. Ellis, G., Levinson, R.: Multi-Level Hierarchical Retrieval. Knowledge-Based Systems, Conceptual Graphs Special Issue 5, 233–244 (1992)
11. Endsley, M.R.: Toward a Theory of Situation Awareness in Dynamic Systems. Human factors 37, 32–64 (1995)
12. Forgy, C.L.: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. Artificial Intelligence 19, 17–37 (1982)
13. Gandon, F.: Graphes RDF et leur Manipulation pour la Gestion de Connaissances, Ch. 4: Graphes comme espaces métriques, HdR, Nice Sophia-Antipolis (2008)
14. Jiang, X., Bunke, H.: Graph Matching. SCI, vol. 73, pp. 149–173 (2008)
15. Laudy, C., Ganascia, J.G., Sedogbo, C.: High-level Fusion based on Conceptual Graphs. In: 10th International Conference on Information Fusion, pp. 1–8 (2007)
16. Mechkour, S.: Overview of Situation Theory and its application in modeling context, Seminar Paper, University of Fribourg (2007)
17. Meissen, U., Pfennigschmidt, S., Voisard, A., Wahnfried, T.: Context- and Situation-Awareness in Information Logistics. In: Lindner, W., Fischer, F., Türker, C., Tzitzikas, Y., Vakali, A.I. (eds.) EDBT 2004. LNCS, vol. 3268, pp. 335–344. Springer, Heidelberg (2004)
18. Montes-y-Gómez, M., Gelbukh, A., López-López, A.: Comparison of Conceptual Graphs. In: Cairó, O., Cantú, F.J. (eds.) MICAI 2000. LNCS, vol. 1793, pp. 548–556. Springer, Heidelberg (2000)
19. Montes-y-Gómez, M., Gelbukh, A., López-López, A., Baeza-Yates, R.: Flexible Comparison of Conceptual Graphs. In: Mayr, H.C., Lazanský, J., Quirchmayr, G., Vogel, P. (eds.) DEXA 2001. LNCS, vol. 2113, pp. 102–111. Springer, Heidelberg (2001)
20. Mugnier, M.L.: On Generalization / Specialization for Conceptual Graphs. Journal of Experimental & Theoretical Artificial Intelligence 7, 325–344 (1993)
21. Padovitz, A., Loke, S.W., Zaslavsky, A.: Towards a Theory of Context Spaces. In: 2nd IEEE Conference on Pervasive Computing and Communications Workshops, pp. 38–42 (2004)
22. Poole, J., Campbell, J.A.: A Novel Algorithm for Matching Conceptual and Related Graphs. In: Ellis, G., Rich, W., Levinson, R., Sowa, J.F. (eds.) ICCS 1995. LNCS, vol. 954, pp. 293–307. Springer, Heidelberg (1995)
23. Sowa, J.F.: Conceptual Graphs. Foundations of Artificial Intelligence, vol. 3, pp. 213–237 (2008)
24. Szczerbak, M.K., Toutain, F., Bouabdallah, A., Bonnin, J.M.: Collaborative Context Experience in a Phonebook. In: 26th IEEE International Conference on Advanced Information Networking and Applications Workshops, pp. 1275–1281 (2012)
25. Szczerbak, M.K., Bouabdallah, A., Toutain, F., Bonnin, J.M.: Generalizing Contextual Situations. In: 6th IEEE International Conference on Semantic Computing (to be published, 2012)
26. Wang, X.H., Gu, T., Zhang, D.Q., Pung, H.K.: Ontology Based Context Modeling and Reasoning using OWL. In: 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops, pp. 18–22 (2004)
27. Yau, S.S., Liu, J.: Hierarchical Situation Modeling and Reasoning for Pervasive Computing. In: 4th IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, pp. 5–10 (2006)
28. Ye, J., Coyle, L., Dobson, S., Nixon, P.: Using Situation Lattices to Model and Reason about Context. In: 4th International Workshop on Modeling and Reasoning in Context, pp. 1–12 (2007)
29. Zhong, J., Zhu, H., Li, J., Yu, Y.: Conceptual Graph Matching for Semantic Search. In: Priss, U., Corbett, D.R., Angelova, G. (eds.) ICCS 2002. LNCS (LNAI), vol. 2393, pp. 92–106. Springer, Heidelberg (2002)