Situation Modeling and Smart Context Retrieval with Semantic Web Technology and Conflict Resolution

Dominik Heckmann

German Research Center for Artificial Intelligence
heckmann@dfki.de

Abstract. We present a distributed service to model and control contextual information in mobile and ubiquitous computing environments. We describe the general user model and context ontology GUMO for the uniform interpretation of distributed situational information in intelligent semantic web enriched environments. Furthermore, we present the relation to the user model and context markup language USERML, that is used to exchange partial models between different adaptive applications. Our modeling and retrieval approach bases on semantic web technology and complex conflict resolution concepts.

1 Integrated Model for Context-Awareness and User-Adaptivity

The research areas *user-adaptivity*, *context-awareness* and *ubiquitous computing* find their intersection in the concept of context, while *semantic web* technology could serve as a mediator between them. In [1] it is pointed out that throughout the different research communities and disciplines, there are various definitions of what exactly is contained in the *context model* [2], the *user model* [3], and the *situation model* [4]. Therefore, it is necessary to clarify how those terms will be used in our approach. A *situation model* is defined in our approach as the combination of a *user model* and a *context model*. Figure 1 presents a diagrammatic answer to the question: *What is situated interaction and how can we conceptualize it? Resource-adaptivity* overlaps with *user-adaptivity* and *context-awareness* because the human's cognitive resources fall into the user model, while the system's technical resources can be seen as part of the context model. The fundamental data structure in our approach is the SITUATIONALSTATEMENT, see [5], that collects apart from the main contextual information also meta data like temporal and spatial constraints, explanation components and privacy preferences. Distributed sets of SITUATIONREPORTS form a coherent, integrated, but still hybrid accretion concept of ubiquitous situation (user and context) models.

2 Context Modeling with `UserML` and GUMO

Ontologies provide a shared and common understanding of a domain that can be communicated between people and heterogeneous and widely spread application systems. Since ontologies have been developed and investigated in artificial intelligence to facilitate knowledge sharing and reuse, they should form the central point of interest

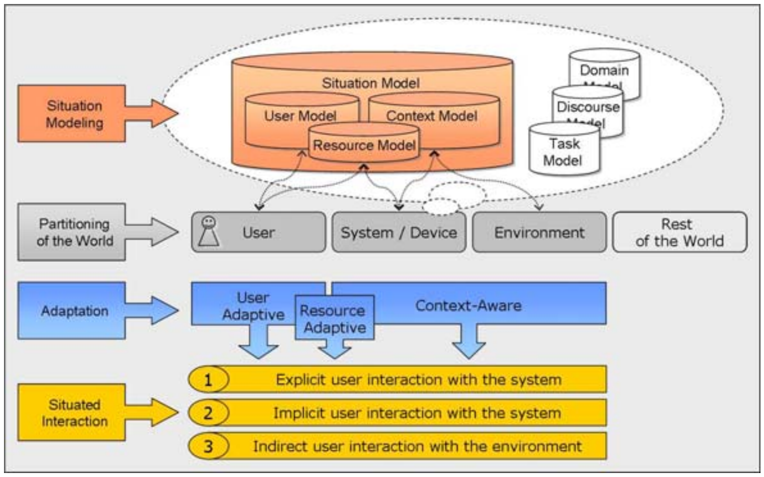

Fig. 1. Situated interaction and the system's situation model for mobile computing

for the task of exchanging situation models. The user model & context markup language USERML is defined as an XML application, see [6]. However, XML is purely syntactic and structural in nature. Nonetheless, the web ontology language OWL has more facilities for expressing semantics. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. Thus, OWL is our choice for the representation of user model and context dimension terms and their interrelationships. This ontology should be available for all user-adaptive and context-aware systems at the same time, which is perfectly possible via internet and wireless technology. The major advantage would be the simplification for exchanging information between different systems. The current problem of syntactical and structural differences between existing adaptive systems could be overcome with such a commonly accepted ontology. GUMO[1] collects the user's dimensions that are modeled within user-adaptive systems like the *user's heart rate*, the *user's age*, the *user's current position*, the *user's birthplace* or the *user's ability to swim*. Secondly, the contextual dimensions like *noise level* in the environment, *battery status* of the mobile device, or the outside *weather* conditions are modeled. The main conceptual idea in SITUATIONALSTATEMENTS, is the division of user model & context dimensions into the three parts: `auxiliary`, `predicate` and `range`. Apart from these `mainpart` attributes, there are predefined attributes about the `situation`, the `explanation`, the `privacy` and the `administration`. Thus, our basic context modeling is more flexible than simple attribute-value pairs or RDF triples. If one wants to say *something about the user's interest in football*, one could divide this into the `auxiliary`=*hasInterest*, the `predicate`=*football* and the `range`=*low-medium-high*. GUMO is designed according to this USERML approach. Approximately one thousand groups of `auxiliaries`, `predicates` and `ranges` have so far been

[1] GUMO homepage: `http://www.gumo.org`

identified and inserted into the ontology. However, it turned out that actually everything can be a `predicate` for the `auxiliary` *hasInterest* or *hasKnowledge*, what leads to a problem if work is not modularized. The suggested solution is to identify basic user model dimensions on the one hand while leaving the more general world knowledge open for already existing other ontologies on the other hand. Candidates are the general suggested upper merged ontology SUMO, see [7], and the UBISONTOLOGY[2], see [8], to model intelligent environments. Identified user model and context `auxiliaries` are for example *hasKnowledge*, *hasInterest*, *hasBelief*, *hasPlan*, *hasProperty*, *hasPlan* and *hasLocation*. A class defines a group of individuals that belong together because they share some properties. Classes can be organized in a specialization hierarchy using `rdfs:subClassOf`.

3 Smart Situation Retrieval with Semantic Conflict Resolution

The architectural diagram in figure 2 shows the SMARTSITUATIONRETRIEVAL or smart context retrieval process. The focus is set on the semantic conflict resolution part. The oval numbers indicate the reading direction. Item (1) shows the request in `UserQL`

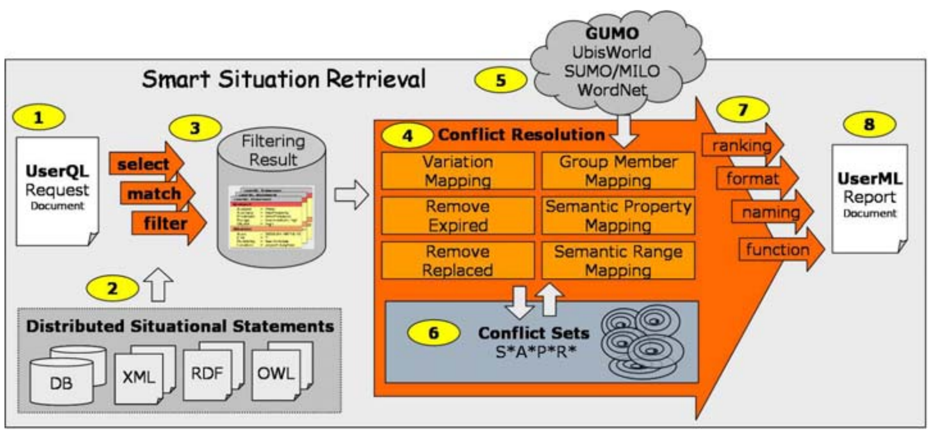

Fig. 2. Smart Situation Retrieval with Focus on Semantic Conflict Resolution

that has to be parsed first. Item (2) points to the distributed retrieval of SITUATIONAL-STATEMENTS. Item (3) summarizes the three macro-steps `select`, `match` and `filter` and presents the FILTERINGRESULT as input to the conflict resolution process. Item (4) stands for the three syntactical procedures VARIATIONMAPPING, REMOVEEXPIRED and REMOVEREPLACED. Item (5) shows the three semantical procedures GROUPMEMBERMAPPING, SEMANTICPROPERTYMAPPING and SEMANTICRANGEMAPPING that base on knowledge in the ontologies GUMO, UbisWorldOntology, SUMO/MILO and the knowledge base WorldNet. Item (6) shows the detection of

[2] UbisWorld homepage: `http://www.ubisworld.org`

syntactic and semantic conflicts and the construction of $\langle S^*, A^*, P^*, R^* \rangle^{nonExpired}_{nonReplaced}$ conflict sets. Item (7) points to the post-processing of `ranking`, `format`, `naming` and `function` that control the output format. Item (8) forms the resulting `UserML` report, that is send via HTTP to the requestor. The *matching procedure* compares all given match attributes with the corresponding statement attributes. Furthermore it integrates semantic functionality like ontological *extension* and spatial *inclusion*. The *filtering procedure* operates on the MATCHINGRESULT. Each statement is individually checked if it passes the *privacy filter*, the *confidence filter* and the *temporal filter*. The *privacy filter* checks if the `statement.access` is either set to public, or if it is set to friends, that the *friends relation* holds between the `query.requestor` and the `statement.owner`, or if it is set to private that the `query.requestor` is the same as `statement.owner`. As every user and every system is allowed to enter statements into repositories, some of this information might be contradictory. Conflicts among SITUATIONALSTATEMENTS like for example a contradiction caused by different opinions of different creators or changed values over time are loosely categorized in the following listing.

1. ON THE SEMANTICAL LEVEL: the systems are not forced to use the same vocabulary, to say the same ontology, to represent the meaning of the concepts, which leads to the user model integration problem number one: *ontology merging* and *semantic web integration*.
2. ON THE OBSERVATION AND INFERENCE LEVEL: several sensors can see same things differently and claim to be right, measurement errors can occur, systems may have preferred information sources
3. ON THE TEMPORAL AND SPATIAL LEVEL: information can be out of date or out of spatial range, a degree of expiry can hold. Reasoning on temporal and spatial meta data is necessary
4. ON THE PRIVACY AND TRUST LEVEL: information can be hidden, incomplete, secret or falsified on purpose, a system of trustworthiness could be applied

Conflict Resolvers are a special kind of filters that control the *conflict resolution process*. An ordered list of these resolvers define the *conflict resolution strategy*. They are modeled in the `query.strategy` attribute. These resolvers are needed if the *match process* and *filter process* leave several conflicting statements as possible answers. Three kinds of conflict resolvers can be identified: the *most(n)*-resolvers that use meta data for their decision, the *add*-resolvers that add expired or replaced statements to the conflict sets, and the *return*-resolvers that don't use any data for their selection.

mostRecent(n) Especially where sensors send new statements on a frequent basis, values tend to change more quickly as they expire. This leads to conflicting non-expired statements. The *mostRecent(n)* resolver returns the *n* newest non-expired statements, where *n* is a natural number between 1 and the number of remaining statements.

mostNamed(n) If there are many statements that claim A and only a few claim B or something else, than *n* of the "most named" statements are returned. Of course it is not sure that the majority necessarily tells the truth but it could be a reasonable rule of thumb for some cases.

mostConfident(n) If the confidence values of several conflicting statements can be compared with each other, it seems to be an obvious decision to return the *n* statements with the highest confidence value.

mostSpecific(n) If the `range` or the `object` of a statement is more specific than in others, the *n* "most specific" statements are returned by this resolver. For example if:

auxiliary=*hasKnowledge*, predicate=*chess* and first range=*yesNo* while the second range=*Novice-Occasional-Professional-Expert-Grandmaster*, the statement with the second range contains a more specific information. Another specificity range ordering is for example: *yesNo* $<$ *lowMediumHigh* $<$ *0%-100%*

mostPersonal(n) If the creator of the statement is the same as the statement's subject (a self-reflecting statement), this statement is preferred by the *mostPersonal(n)* resolver. Furthermore, if an *is-friend-of relation* is defined, statements by friends could be preferred to statements by others.

These conflict resolver rules are based on common sense heuristics, however they need not to be true for specific sets of statements. An important issue to keep in mind is the problem that resolvers and strategies imply uncertainty. To contribute to this fact, the confidence value of the resulting statement is appropriately changed, furthermore the conflict situation is added to the evidence attribute.

Summary and Acknowledgements

We have introduced an integrated architecture for *Situation Modeling* and *Smart Context Retrieval*. We have clarified a model for situated interaction and context-awareness. The context exchange language UserML has been presented as well as the general user model & context ontology GUMO. Our approach bases on semantic web technology and a complex conflict resolution and query concept, in order to be flexible enough to support adaptation in human-computer interaction in ubiquitous computing. This work has been supported by the German Ministry of Education and Research within the project SPECTER at the German Research Center for Artificial Intelligence (DFKI).

References

1. Kray, C.: Situated Interaction on Spatial Topics. Volume 274 of Dissertations in Artificial Intelligence. INFIX (2003) ISBN: 1 58603 391 3.
2. McCarthy, J., Buvac, S.: Formalizing context (expanded notes). In Aliseda, A., van Glabbek, R.J., Westerstahl, D., eds.: Computing Natural Language. Volume 81 of CSLI Lecture Notes. Center for the Study of Language and Information, Standford University, CA (1998) 13–50
3. Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context-awareness. Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology, Atlanta, Georgia, U.S.A. (1999)
4. Jameson, A.: Modeling both the context and the user. Personal Technologies **5** (2001) 29–33 Available from http://dfki.de/∼jameson/abs/Jameson01PT.html.
5. Heckmann, D.: Introducing situational statements as an integrating data structure for user modeling, context-awareness and resource-adaptive computing. In Hoto, A., Stumme, G., eds.: LLWA Lehren - Lernen - Wissen - Adaptivität (ABIS2003), Karlsruhe, Germany (2003) 283–286
6. Heckmann, D., Krüger, A.: A user modeling markup language (UserML) for ubiquitous computing. Lecture Notes in Artificial Intelligence **2702** (2003) 393–397
7. Pease, A., Niles, I., Li, J.: The suggested upper merged ontology: A large ontology for the semanticweb and its applications. In: AAAI-2002Workshop on Ontologies and the Semantic Web. Working Notes (2002) http://projects.teknowledge.com/AAAI-2002/Pease.ps.
8. Stahl, C., Heckmann, D.: Using semantic web technology for ubiquitous location and situation modeling. The Journal of Geographic Information Sciences CPGIS: Berkeley **10** (2005)