

The Impact of Context on the Trustworthiness of Communication: An Ontological Approach

Santtu Toivonen¹ and Grit Denker²

¹ VTT Information Technology
P.O.Box 1203, FIN-02044 VTT, FINLAND
santtu.toivonen@vtt.fi

² SRI International
Menlo Park, CA 94025, USA
grit.denker@sri.com

Abstract. We outline a Semantic Web approach for considering the impact of context information on the trustworthiness of communication. We show that the contexts of message sender, receiver, and mediating network can have influence on the degree of trust the receiver assigns to a message. We define ontologies to capture context-sensitive messaging and trust, as well as propose trust evaluation functions. We illustrate their usage with rules that determine trust factors of context information.

1 Introduction

Messages conveyed to us in everyday life often have an influence on our decisions and behaviors. The more trustworthy a message is considered, the higher will be its impact on the receiver. Trust is thereby a very significant commodity in communication. But how is trust established? A variety of factors influences trust establishment, among them integrity, reputation, credibility, reliability, congruity, predictability, and responsibility (cf. proceedings of the iTrust [12], conferences on trust management, and [11, 1]).

The decision whether or not to trust a piece of information can depend on many contextual factors including creator (who) of the data (what), time (when), location (where), and intent (why) of origination, social context of receiver and many more. Generally, context information can characterize a situation of any entity—be it a person, place, or object—that is relevant to the interaction between a user and an application, including the user and the application themselves [17]. We argue that for message-based communication, a notion of context information needs to be taken into consideration to determine the trustworthiness of a message.

A specific advantage of making context explicit in message exchanges is that this information can be used in trust policies. For example, a policy can state that news information related to a particular location is to be trusted more if the reporting entity was at the location at the time when the event occurred. In this sense, policies define how to process context information to derive trustworthiness assertions. On the basis of our integrated context and message ontology, defined in OWL [15], we will illustrate sample trust policies and discuss reasoning strategies.

We start the paper with a motivating scenario that illustrates the usefulness of integrating context information in messaging systems (see Section 2). Generalizing from the observations of the example in Section 3, we address features of context information and consider its relationship with trustworthiness to define a context-sensitive messaging ontology and a trust ontology that takes context into account. The use of context information in rules to determine its impact factor for trust is presented in Section 4. In this section we also propose evaluation function to determine a context-sensitive trust value for a message. We then present some related work in Section 5, and give a general discussion about the idea in Section 6. Finally, we close by drawing concluding remarks and future research directions in Section 7.

2 Motivating Example: An Accident Report

Consider a case where on his way to work Jim is witnessing the town’s city hall caught in fire. He decides to share this information with his local friend Jane, who he knows of currently being abroad. Jim is close to the scene of the accident, along with a huge number of other curious people. Jane is on vacation on a remote island with her friends. In this example the sender’s, i.e. Jim’s, relevant context attributes are his location, his activity, and the time of sending the message. The network context concerns several networks: The network Jim is connected to, the network Jane is connected to, and the set of networks in between them that are responsible for transmitting the message onwards. Let’s say that the network characterization is a compound of them all. Jane’s relevant context attributes are her location, activity, mood and social context.

The contextual information has impact on the trust Jane assigns to the message in the following way. First of all, Jim’s being close to the site of the accident increases Jane’s trust in the message. Jane can assume that he is an eyewitness of the accident. Secondly, his activity increases the trust as well. Jane knows that Jim works as a journalist for a newspaper and this gives Jane the impression that he is really trying to find out the true state of affairs in order to not report falsehoods. Third, Jane’s being abroad increases her interest and trust in the message. If she was in her home town, she might well want to verify Jim’s message from some news coverage. Finally, the long delay in receiving Jim’s message— for example due to a technical network failure—decreases Jane’s trust in the message. This is because the message content is in the present tense, but the fire might actually be already put out once Jane receives the message.

3 Context Information and Trustworthiness

3.1 Features of Context Information

As mentioned, our focus is on communication in context-sensitive environments. The smallest unit we consider to be enriched with trust values is one message sent between two communicating entities. Figure 1 depicts the concept of CONTEXT and how it is connected to MESSAGES. A message, concerning a TOPIC, is exchanged between SENDER and RECEIVER using a NETWORK. There are three kinds of contexts associated to a message, the context of the network used for transmitting the message (NETCTX), and

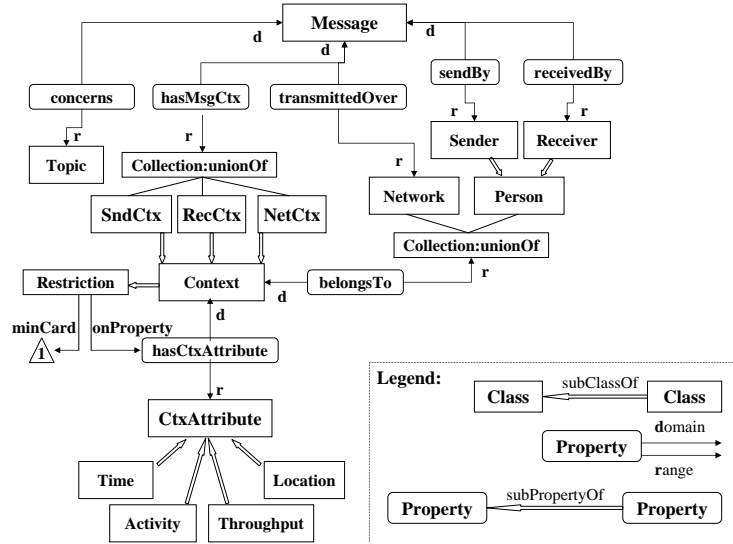


Fig. 1. OWL Ontology for attaching context information to messages

the contexts of the sender (SNDCTX) and receiver (RECCCTX). In the following we refer to these contexts as ctx_n , ctx_s , and ctx_r , respectively. Context is described through C-TXATTRIBUTES. Formally, ctx with n context attributes can be defined as follows:

$$ctx = \{ctxAttr(1), ctxAttr(2), \dots, ctxAttr(n)\}$$

The specific attributes that constitute a user's context are addressed in several previous work and we do not go into details of that issue in this paper. Instead, we take a couple of context attributes that are frequently considered as relevant. Such are for example TIME, LOCATION and ACTIVITY. As for network, context can be expressed for example with network's quality of service (QoS) values such as THROUGHPUT [21]. Some of these attributes are included as examples in Figure 1. Restrictions on the kinds of context attributes for classes are needed, e.g., THROUGHPUT is a possible context attribute for ctx_n , but not for ctx_s or ctx_r . The BELONGSTO relation between the context and an entity (person or network), connects the message contexts with the associated sender, receiver and network. Appropriate ontological definitions (restrictions), assure that the sender of a message is the same entity as the entity to which the sender context of that message belongs. Moreover, appropriate restrictions on the range classes of the BELONGSTO property for three context subclasses assure that a sender context always belongs to a sender entity and so on. For reasons of space limitation, we do not show all restriction definitions of our ontologies. Instead, we refer the reader to <http://www.csl.sri.com/users/denker/owl-sec/context/>.

In Figure 2 we give an overview of the proposed trust ontology that extends topic-dependent trust and agent trust relations as proposed in [9], with the notion of context-sensitive trust in messages. The topic of a message can have impact on its trust level. For example, a geography student can have complete trust on his professor when it comes

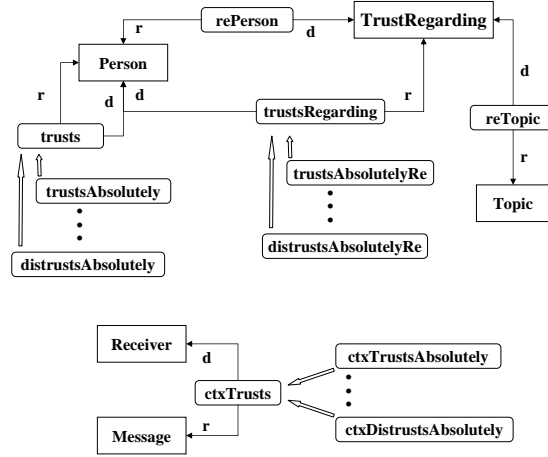


Fig. 2. Core concepts for context-sensitive trust

geographical phenomena, but much less trust in the area of cooking. This information is captured in the TRUSTSREGARDING property. Trust relationships between persons are captured by the TRUSTS relationship. Subproperties of these two relationships, depicted in Figure 2 with (TRUSTSABSOLUTELY, . . . , DISTRUSTSABSOLUTELY), and (TRUSTSABSOLUTELYRE, . . . , DISTRUSTSABSOLUTELYRE), conform with the nine levels of trust defined in [9, 8], namely TRUSTSABSOLUTELY, TRUSTSHIGHLY, TRUSTSMODERATELY, TRUSTSLIGHTLY, TRUSTSNEUTRALLY, DISTRUSTSLIGHTLY, DISTRUSTSMODERATELY, DISTRUSTSHIGHLY, and DISTRUSTSABSOLUTELY.

To capture the dependency of trust on context, we introduce the CTXTRUSTS relation between a message and its receiver. Since a message has a receiver, a sender and (possibly) a mediating network associated with it, which in turn have contexts attached to them, the value of the CTXTRUSTS relation will be computed using the TRUSTS and TRUSTSREGARDING relations, and context information. We envision the use of rules that describe how context-sensitive trust is computed on the basis of context-independent trust relations such as TRUSTS and TRUSTSREGARDING (See Section 4).

3.2 Accident Example Revisited

Let us now briefly return to the accident example presented above, but this time utilizing the concepts defined in the ontology. The contexts of our accident report example message “The city hall is on fire!” are as follows. The message sender’s, e.g. Jim’s, context can be formalized as follows:

$$ctx_{Jim}^{fireMsg} = \{time^{fireMsg}, location^{fireMsg}, activity^{fireMsg}\}$$

with certain instances for the context attributes, such as $activity^{fireMsg} = Journalist$. Network context, in turn, consists of the following:

$$ctx_{Net}^{fireMsg} = \{throughput^{fireMsg}\}$$

And finally, the receiver:

$$ctx_{Jane}^{fireMsg} = \{location^{fireMsg}, activity^{fireMsg}, mood^{fireMsg}, socCtx^{fireMsg}\}$$

The context values influence the overall trust value the receiver assigns to the message. This is defined using rules for reasoning about context-sensitive trust and functions for assigning context-sensitive trust values.

4 Rules and Functions for Context-sensitive Trust

On the basis of the context-sensitive message and the trust ontologies, we define rules that express how the value of the CTXTRUSTS relation is influenced by the values of context-sensitive relations. As a first simplified approach, rules are of the form:

```
[Rule label:]
IF expression
THEN ctxTrust_label(receiver, msg) =
    f(ctx-independent-trust(receiver, msg), factor)
```

A rule describes how certain context-sensitive information (captured in the expression) reduces or enhances the trust in a message, relative to its context-independent trust value. *Expression* is a boolean combination of terms constructed of variables and predicates defined in our two ontologies. Though we will use an ad hoc notation in this paper, the example rules could easily be specified in a semantic rule notation such as SWRL [10] or Rei [13]. *f* is a function to compute context sensitive trust using trust factor *factor*. Before we go into details of the trust rules, we propose some possibilities for trust functions as well as functions to compute context-independent trust.

4.1 Trust Function for Context-sensitive Messages

In this section we consider context-independent trust that is computed using the topic-independent trust relations and the topic-dependent trust relations given in Figure 2 (cf. [9]). Topic-independent trust can be defined to extend the nine trust-levels as follows:

$$ctx\text{-independent}\text{-trust}(r, msg) = trust(r, s) + trustsRegarding(r, tr)$$

where *msg* is a message with sender *s*, receiver *r*, concerning topic *t* and *tr* is a variable of type TrustRegarding with *rePerson*(*tr*, *r*) and *reTopic*(*tr*, *t*). Thus, the degree to which a receiver trusts a message independent of the message context is computed by adding the trust the receiver has in the sender of the message in general and more specifically concerning the topic of the message. This definition requires to define how trust relations are added. We propose the following simple definition, though other, more complex computations (e.g., using weights) could be defined if deemed necessary by the application. Moreover, if other context-insensitive trust relations are defined in a given application domain, then these could be included in a more complex definition of ctx-independent trust between a receiver and a message.

Under the assumption that all instances of the TRUSTS and TRUSTSREGARDING properties belong to one of the subclasses TRUSTSABSOLUTELY, ..., DISTRUSTSABSOLUTELY, we define the following value assigning function³

$$\text{value} \begin{pmatrix} \text{distrustsAbsolutely}(r, s) \\ \text{distrustsHighly}(r, s) \\ \dots \\ \text{trustsNeutrally}(r, s) \\ \dots \\ \text{trustsHighly}(r, s) \\ \text{trustsAbsolutely}(r, s) \end{pmatrix} = \text{value} \begin{pmatrix} \text{distrustsAbsolutelyRe}(r, tr) \\ \text{distrustsHighlyRe}(r, tr) \\ \dots \\ \text{trustsNeutrallyRe}(r, tr) \\ \dots \\ \text{trustsHighlyRe}(r, tr) \\ \text{trustsAbsolutelyRe}(r, tr) \end{pmatrix} = \begin{pmatrix} -4 \\ -3 \\ \dots \\ 0 \\ \dots \\ 3 \\ 4 \end{pmatrix}$$

Similarly, an inverse function value^{-1} is defined to assign context-independent trust predicates to integers, namely

$$\text{value}^{-1} \begin{pmatrix} -4 \\ -3 \\ \dots \\ 0 \\ \dots \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} \text{cxt-independent-distrustsAbsolutely}(r, msg) \\ \text{cxt-independent-distrustsHighly}(r, msg) \\ \dots \\ \text{cxt-independent-trustsNeutrally}(r, msg) \\ \dots \\ \text{cxt-independent-trustsHighly}(r, msg) \\ \text{cxt-independent-trustsAbsolutely}(r, msg) \end{pmatrix}$$

With this, we can define

$$\begin{aligned} \text{cxt-independent-trust}(r, msg) &= \text{trust}(r, s) + \text{trustsRegarding}(r, tr) \\ &= \text{value}^{-1} \left(\left\lceil \frac{\text{value}(\text{trusts}(r, s)) + \text{value}(\text{trustsRegarding}(r, tr))}{2} \right\rceil \right) \end{aligned}$$

The context-insensitive trust is used to compute context-sensitive trust by applying a function that takes into account the trust factors determined by the rules. In the next section we illustrate these trust rules.

4.2 Rules for Computing Context-sensitive Trust

Next we show example rules for context-sensitive trust. The following rules formalize the implicit policies of the accident example presented in Section 2. The rules are defined for a variable `msg` of type `message`, and variables `s` of type `sender`, `r` of type `receiver`, `n` of type `network`, and `l` of type `location`. Moreover, we assume that additional predicates are defined for concepts in our ontology. For example, we will assume a boolean predicate `CLOSE` on locations, a `CONTENT` predicate for messages, and a `LOCATION` predicate for message content.

```
[Rule 1:]
IF sendBy(msg,s) AND receivedBy(msg,r) AND
   hasMsgCtx(msg,ctx_s) AND belongsTo(ctx_s,s) AND
   hasCtxAtt(ctx_s,l) AND close(l, location(content(msg)))
THEN ctxTrusts_l(r,msg)=f(ctx-independent-trust(r,msg), 2)
```

where, for example, f is defined as follows: For cit being one of the nine cxt-independent trust relations between receivers and messages (i.e., CXT-INDEPENDENT-ABSOLUTELYTRUSTS, ..., CXT-INDEPENDENT-ABSOLUTELYDISTRUSTS)

$$f(cit, factor) = \text{value}^{-1} \left(\left\lceil \frac{\text{value}(cit) + factor}{2} \right\rceil \right).$$

³ The same function name (*value*) is used for both TRUSTS and TRUSTSREGARDING.

Thus, the rule says that context-sensitive trust is increased by two trust levels if the message is sent by someone close to the location about which the message reports. Similar rules could express that the receiver's trust is increased if the message talks about a location that is far from where the receiver resides when receiving the message.

```
[Rule 2:]
IF receivedBy(msg,s) AND receivedBy(msg,r) AND
   hasMsgCtx(msg,ctx_r) AND belongsTo(ctx_r,r) AND
   hasCtxAtt(ctx_r,l) AND far(l, location(content(msg)))
THEN ctxTrusts_2(r,msg)=f(ctx-independent-trust(r,msg), 1)
```

A "0" trust factor in a rule does not influence the context-insensitive trust value, and negative trust factors decrease trust. As there are possibly more than one rule that apply to a (r, msg) pair, the overall trust value is given as

$$ctxTrusts(r, msg) = \frac{\sum_{rules} ctxTrusts_{-l}}{number\ of\ rules} \quad (1)$$

As our example shows, predicates relating context information with message content are useful. We are currently developing our ontology to include predicates that we deem most common for the context attributes we consider. For example corresponding predicates for location and time on messages content will be useful to express rules. Other extensions of our ontology will be concerned with different kinds of trust. For example, a rule could express that a receiver of a message decreases the trust value of a message received, if the throughput of the network is very small and the message content large (e.g., including picture showing the town hall on fire). Though the receiver might generally trust the message content, she might doubt its temporal validity.

5 Related Work

Approach to reputation-based trust such as [9, 8] require explicit trust ratings. It depends on the definition of the trust rating to what extent contextual information is taken into account. REGRET [19, 20] is an approach for adding reputation in multi-agent systems. It can be said to come quite close to social contexts and their utilization in trust-assignment. However, it is limited to the social aspects of context-sensitivity. To our knowledge there is no general approach to include context information in trust ratings. Bizer et al [2] point out the importance to investigate context- and content-based trust policies on the Semantic Web. They very briefly introduce an architecture that has a query and trust evaluation layer to handle trust decisions, but give no further details. Trust, security, and privacy issues in context-aware environments can be approached in various ways. For example, people can be provided with control over who can access their personal information in various contexts as investigated in [7, 22].

Our approach defines a context and message ontology and extends well-known trust concepts into a context-sensitive trust ontology where trust is linked to message context. On the basis of this ontology, we can formalize rules that take context-sensitive information into account when computing trust values. However, our context-sensitive trust ontology could also be connected with other existing ontologies. For example,

SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications) [3] could be used for expressing mobile contexts in a more detail, time ontology [18] for representing the temporal aspects, the network QoS ontology [21] for characterizing the network's properties, and FOAF (Friend of a Friend) [6] for expressing the contact details and other characteristics of people and their interrelationships.

6 Discussion

A relevant question to ask is who or what needs to provide trust functions and rules to enable context-sensitive trust calculation. If the burden is entirely put on the users themselves without any assistance from the information technology (IT) infrastructure, this approach might not be adopted. If, on the other hand, the trust is calculated either partially or entirely by the IT infrastructure, instantiating some of our trust relations, the adoption would be easier. Though one still needs to consider who in the value chain of operators in the Semantic Web would benefit from providing (semi)automatic trust calculations. A related question is whether the context information is automatically determined by the infrastructure or manually entered by the users, and does this have any significance to the trust people assign to messages.

Besides the “who”, “how” is another question to be answered. How the context-sensitive trust factor is to be determined? Retrieving context information of users and other components of the system is difficult enough in the first place, but it should additionally be somehow transformed into meaningful factors of context-sensitive trust. This task encapsulates both a technological challenge of retrieving contextual information from text—such as Jim’s being close to the city hall in the accident example above—and a broader challenge of determining the meaningful impact of the retrieved context information. In another kind of message, such as “I like the city hall’s architecture”, Jim’s being in the immediate vicinity of the city hall is not so important. In this paper, we illustrated rules for transforming the context-information into a factor of trust, but did not address that the trust factor might be different for different aspects of trust, such as “trusting a message” vs. the more specific issue of “trusting the temporal validity of a message.”

If the above “who” and the “how” questions get answered, one can still ask what purpose does the context-sensitive trust calculation serve. This is naturally related to the “who”-question, but is slightly different. Considering this “what purpose”-question entails reflecting the goal of the Semantic Web in general, i.e., the web of trust. The decentralized nature of the Semantic Web implies also decentralized trust, which brings about trust networks. In some cases the trust networks can vary based on the contextual information, as argued above, so the general purpose of the context-sensitive trust is to contribute to achieving the Semantic Web’s general goal.

Additional complexity comes into play when reliability of context information becomes a parameter. In our current approach, we assume context information to be reliable. If we loosen this assumption, then context information needs to be evaluated for its trustworthiness. Finally, one needs to address the issue of disseminating the policies that constitute the relevant context information to trust determination for other agents in addition to message receivers.

7 Conclusions and Future Work

We presented ontologies to capture message context and context-dependent trust relations. We illustrated how rules can be used to define context-sensitive trust, and we proposed several trust evaluation functions.

Much work lies ahead and further investigations are necessary with respect to topics we did not address in this paper. For example, we intend to provide an implementation of context-sensitive trust rules and evaluation functions using SRI's rewriting framework Maude [16, 14, 4]. Rule declarations are at the heart of Maude and the Maude execution allows to define strategies for evaluating rule-based systems. This way we can automate the process of trust evaluation, as well as experiment with different kinds of trust functions, as new functions can be added easily. Maude is a very efficient, fast rewrite engine with thousands of rewrites per second. Thus, experimentation with a large number of rewrite rules and facts to evaluate the effects of various trust functions can be handled in this framework.

Future work will also address experiments evaluating the adequacy of our trust ontology by illustrating its use in various application domains. On one hand, it will be important to exploit existing ontologies, such as those developed in the DAML project [5] for time and space, and other application specific ontologies. On the other hand, experiments have to be performed to compare results with and without the use of context information for trust determination.

Assigning context-dependent trust on messages presupposes extracting relevant information from the message contents (such as the location of the city hall) and comparing it with the contexts of the sender, network, and the receiver. Information extraction from free-form messages like "The city hall is on fire!" is by no means an easy task. Instead, if the message contents were structured and conformed to some machine-understandable semantics, the information extraction and combination with context attributes would become much easier. In our future work we are taking steps towards this direction by analyzing trust people assign in structured service descriptions, such as pizzeria recommendations and bus timetables. These service descriptions would be passed to the users both directly from the service providers and also—and more importantly, as far as our research is concerned—via other users such as their friends or colleagues. The information extraction would likely become easier, since the pizzeria recommendation, for example, would contain the location of the pizzeria and its opening hours, which could be mapped with the contexts of the message sender and receiver.

Acknowledgements. We would like to thank the anonymous referees for their helpful comments and for suggesting interesting questions, e.g., reliable context information and dissemination of context policies, for future research.

References

1. B. Acrement. Elements for building trust. Available at: http://www.imakenews.com/smei/e_article000051474.cfm.
2. C. Bizer. Using context- and content-based trust policies on the Semantic Web. In *Poster in Proceedings of WWW 2004, May 17-22, 2004, New York*, pages 228–229. ACM, 2004.

3. H. Chen, F. Perich, T. Finin, and A. Joshi. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In *International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Boston, MA, August 2004.
4. M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and J. Quesada. *Maude: Specification and Programming in Rewriting Logic*. SRI International, Computer Science Laboratory, Menlo Park, CA, January 1999. Available at: <http://maude.csl.sri.com/manual>.
5. Darpa agent markup language (daml) project web site. Available at: <http://www.daml.org>.
6. Friend of a friend (foaf) project web site. Available at: <http://www.foaf-project.org>.
7. F. Gandon and N. Sadeh. Semantic web technologies to reconcile privacy and context awareness. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3):241–260, 2004.
8. J. Golbeck and J. Hendler. Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In *Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW) 2004*, Northamptonshire, UK, October 2004. Springer.
9. J. Golbeck, B. Parsia, and J. Hendler. Trust networks on the semantic web. In M. Klusch, S. Ossowski, A. Omicini, and H. Laamanen, editors, *Proceedings of Cooperative Intelligent Agents (CIA) 2003*, pages 238–249, Helsinki, Finland, August 2003. Springer.
10. I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, and M. Dean. Swrl: A semantic web rule language combining owl and ruleml. Technical Report Version 0.5, November 2003. Available at: <http://www.daml.org/2003/11/swrl>.
11. J. Hradesky. *Total Quality Management Handbook*. McGraw Hill Text, December 1994.
12. iTrust - an information society technology (ist) working group. Available at: <http://www.itrust.uoc.gr>.
13. L. Kagal, T. Finin, and A. Joshi. A policy language for a pervasive computing environment. In *IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, 2002.
14. Maude Web site. Available at: <http://maude.csl.sri.com/>, 2000.
15. D. McGuinness and F. van Harmelen. Owl web ontology language overview. The World Wide Web Consortium (W3C), February 2004. Available at: <http://www.w3.org/TR/owl-features>.
16. J. Meseguer. A logical theory of concurrent objects and its realization in the Maude language. In G. Agha, P. Wegner, and A. Yonezawa, editors, *Research Directions in Concurrent Object-Oriented Programming*, pages 314–390. The MIT Press, 1993.
17. T. Moran and P. Dourish. Special issue on context-aware computing. *Human Computer Interaction*, 2001.
18. F. Pan and J. Hobbs. Time in owl-s. In *Proceedings of the 2004 AAAI Spring Symposium on Semantic Web Services*, Stanford, CA, March 2004.
19. J. Sabater and C. Sierra. Regret: reputation in gregarious societies. In *Proceedings of the fifth international conference on Autonomous agents*, pages 194–195. ACM Press, 2001.
20. J. Sabater and C. Sierra. Reputation and social network analysis in multi-agent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 475–482. ACM Press, 2002.
21. S. Toivonen, H. Helin, M. Laukkanen, and T. Pitkäranta. Context-sensitive conversation patterns for agents in wireless environments. In S.K. Mostefaoui, Z. Maamar, and O. Rana, editors, *Proceedings of the The First International Workshop on Ubiquitous Computing (IWUC'04) held in conjunction with ICEIS 2004*, pages 11–17, Porto, Portugal, April 2004. INSTICC Press.
22. S. Toivonen, J. Kolari, and T. Laakko. Facilitating mobile users with contextualized content. In *Proceedings of the Artificial Intelligence in Mobile Systems (AIMS) 2003 Workshop held in conjunction with Ubicomp 2003*, Seattle, WA, October 2003.