Smart Spaces for Ubiquitously Smart Buildings

Kary Främling
Helsinki University of Technology
PO Box 5500, Espoo, Finland
Email: Kary.Framling@hut.fi

Ian Oliver, Jukka Honkola
Nokia Research
Helsinki, Finland
Email: firstname.lastname@nokia.com

Jan Nyman
Posintra Oy, Electrical Building Services Centre
Kipinätie 1, FIN-06150 Porvoo, Finland
Email: jan.nyman@posintra.fi

Abstract

Building automation is a domain where interoperability between equipment made by different manufacturers is rare. In addition to competing standards, completely proprietary solutions are also common. This is a great challenge for implementing 'ubiquitously smart buildings', where building automation systems, user interfaces and services can interact seamlessly. The paper describes how a Semantic Information Broker (SIB) can be used as an enabler of interoperability, where an ecosystem of supplementary services is created through manufacturer-agnostic agents.

1. Introduction

In science fiction literature and movies, domestic robots, electrical doors and fancy space shuttles are some of the most visible artifacts of the future. In our current world, smart environments may not be as flashy as in science fiction, they will rather provide inobtrusive means to make people's lives easier, reduce energy consumption and environmental footprint, as well as improve the quality of life in general. In this paper, we show how smart buildings can be created as parts of smart environments.

Creating smart buildings and smart environments in general has been a topic of research and development for a long time. Despite those efforts, such environments are still largely found only in experimental or pilot environments. In this paper, we describe a distributed information architecture that makes it possible to implement such smart environments on a large scale by integrating information access to and control of different building automation systems. Building automation is a domain where interoperability is a challenge due to conflicting interface and communication standards, e.g. KNX, LON, Modbus etc., in addition to a great number of prorietary solutions.

In the ongoing DIEM project (Devices and Interoperability Ecosystem, http://www.diem.fi) and its predecessors, solutions to these interoperability challenges have been developed using device and protocol adapters that enable unified information access to them all on the Internet Protocol level and notably through Service Oriented Architecture (SOA) solutions. Such SOA-based solutions are good in the case where standards (real or de-facto) exist for the semantic representation of the information. In practice, there is a lack of universal standards. Meanwhile there tends to be many potential interfaces available developed by different organisations and projects, which are not interoperable. This lack of compatibility is a major obstacle for creating *Smart Spaces* where humans and devices could interact smoothly.

The Smart Spaces notation is heavily overloaded and has been used for describing a wide variety of things. In this paper, we use it to signify a geographical space where information is available about the space itself, the devices and services available in it, the people present in it and about other potentially useful information or services. Such a Smart Space concept has been initially proposed in [1] and developed further in [2] as a solution to enabling interoperability. As no standards exist that would cover the information representation needs of such Smart Spaces, we believe an incremental process will occur [3]. In the first phase, devices and systems will publish their available information and services using their current semantic notations (standardised or not). When the information becomes available, that makes it possible to create new services that use the information, while augmenting it with information about themselves and information produced by themselves.

In the Sedvice/M3 architecture [2], information is expressed as subject-relation-object Triples that build up labeled, directed multi-graphs (one or more). In the rest of this paper we will call such graphs *semantic nets* even though graph theory and semantic net theory use partially different vocabularies and present other potential incompatibilities due to their background and history. The Triples are represented using Resource Description Framework (RDF) notations. Triples are stored and managed by the Semantic Information Broker (SIB), which can be distributed over many devices. The Smart Space Access Protocol (SSAP) is used for performing operations on the semantic net.

After this introduction, the paper gives a state of the art overview of building automation systems and Smart Spaces. In Section 3 we describe the whole system architecture and in Section 4 we show the current level of implementation, followed by conclusions.

2. Background

Building automation systems and Smart Spaces are currently two disctinct domains with different technological and scientific backgrounds, which is the reason for splitting this section. We will provide an overview of the state-of-the-art for both domains, as well as some background for the work reported in this paper.

2.1. Building automation systems

Systems integration in buildings has traditionally been about physical dimensions, voltage, plug dimensions etc. Control mechanisms usually control either one device only (e.g. a lamp, refrigerator etc.) or power supply for security reasons (e.g. fuses, main switch etc.). Implementing integrated functions such as switching the power off from certain appliances, cutting off water supply and activating the burglar alarm with one single 'leaving home' command has required a lot of dedicated cabling and custom devices, installed by professionals.

Different communication standards have been defined in order to provide more feasible solutions, such as LON (http://www.lonmark.org/), KNX (http://www.knx.org/) and ModBus (http://www.modbus.org/). However, none of these has become a global standard that all manufacturers would support. They also tend to be expensive to install, maintain and upgrade. Furthermore, they are not conceived in a way that would allow for easy integration between them; in fact, they may even on purpose be designed in a way that makes interoperability more difficult due to commercial reasons.

Meanwhile, remote monitoring and control of buildings has become a common functionality at least for bigger buildings such as shopping centers, office buildings, libraries etc. Remote monitoring services are becoming an increasingly important part of the business of traditional building companies as well as other companies. These systems tend to use internet as the information channel because it is cheap to set up and use. As people become increasingly connected to the internet from their homes, internet and the communication protocols associated with it have become an interesting option also for building automation solutions. The fact that many multimedia devices (including mobile phones) integrate internet connectivity by default, makes it possible to take systems integration and usability to levels that are not possible with 'classical' building automation systems.

Figure 1 illustrates how different devices can be connected to a 'protocol converter' that makes device information available through internet protocols. The 'protocol converter' can be an ordinary computer or a cheaper and more energy-efficient solution, such as the Home Control Center (http://smarthomepartnering.com/cms/) proposed by Nokia. Device connectivity is implemented through adapters that convert the underlying protocols into a generic internet interface.

In practice, a generic internet interface nowadays signifies a browser-compatible format (HTML and others) for user interfaces and XML messages for machine-readable information. For successful machine-to-machine communication, the semantics of the XML messages have to be understood in the same way by both parties. The currently most used method for describing message semantics is XML Schemas. In building automation, the oBIX (Open Building Information Xchange, http://www.obix.org/) is an example of such a protocol. Devices Profile for Web Services (DPWS) is another initiative with similar goals. In addition to these, more generic messaging protocols exist that are intended for communication with any kind of devices (not only related to building automation). The PROMISE Messaging Interface (PMI) [4] is an example of such an interface. The IP for Smart Objects (IPSO) alliance (www.ipso-alliance.org) has similar objectives but it is unclear whether they have yet specified any messaging protocols. In practice, none of these has obtained global acceptance.

Figure 1. Connecting devices to the Internet.

2.2. Smart Spaces — The M3 Concept

The M3 system [1], [5], [2] consists of a space based communication mechanism [6], [7] for independent agents. The agents communicate implicitly by inserting information to the space and querying the information in the space. The space is represented by one or more Semantic Information Brokers (SIBs), which store the information as a Resource Description Framework graph (RDF). The agents can access the space by connecting to any of the SIBs making up the space by whatever connectivity mechanisms the SIBs offer. Usually, the connection will be over some network, and the agents will be running on various devices. The information in the space is the union of the information contained in the participating SIBs. Thus, the agent sees the same information content regardless of the SIB to which it is connected.

The agents may use five different operations to access the information stored in the space:

Insert	: Insert information in the space
Remove	: Remove information from the space
Update	: Atomically update the information, i.e. a combination of insert and remove executed atomically
Query	: Query for information in the space
Subscribe	: Set up a persistent query in the space; changes to the query results are reported to the subscriber

In addition to these access operations there are *Join* and *Leave* operations. An agent must have *joined* the space in order to access the information in the space. The join and leave operations can thus be used to provide access control and encrypted sessions, though the exact mechanisms for these are still undefined.

In its basic form the M3 space does not restrict the structure or semantics of the information in any way. Thus, we do not enforce nor guarantee adherence to any specific ontologies, neither do we provide any complex reasoning[1] [8], [9]. Furthermore, information consistency is not guaranteed. The agents accessing the space are free to interpret the information in whatever way they want.

We are planning to provide, though, a mechanism to attach agents directly to the SIBs. These agents have a more powerful interface to access the information and can be e.g. guaranteed exclusive access to the information for series of operations. Such agents may perform more complex reasoning, for example ontology repair or translation between different ontologies. However, they may not join any other spaces but are fixed to a single SIB and thus a single space.

The M3 spaces are of local and dynamic nature, in contrast to semantic web which embodies Tim Berners-Lee's idea of semantic web [10] as a "giant global graph". We envision that the spaces will store very dynamic context information, which poses different challenges than the internet-wide semantic web. For example, in order to provide a true interoperability for local ubiquitous agents, the space (i.e. SIBs) will have to provide a multitude of connectivity options in addition to http: plain TCP/IP, NoTA [11], Bluetooth, RFID [12]. Furthermore, the space should be fairly responsive. While we do not aim for real-time or near real-time system, even half minute long response times for operations are unacceptable.

The responsiveness is one of the factors behind the fundamental decision to not enforce any specific ontologies and allowing the agents to interpret the information freely, as it lessens the computational burden of the infrastructure. Another, and more important reason is that we explicitly want to allow mashing up information from different domains in whatever way the agents see best. Strict ontology enforcement would make this kind of activity extremely difficult as all new ways of mashing up the information would require approval from some ontology governance committee. However, we still plan to provide means for ontology enforcement for cases where the space provider explicitly wishes to restrict the ways the information is. Such situations will occur in reality where such enforcement is the best approach.

The information content in a M3 space may be distributed over several SIBs. The distribution mechanism assumes that the set of SIBs forming a M3 space are totally routable but not necessarily totally connected. The information content that the agents see is the same regardless the SIB where they are connected [13]. Distribution may also occur between first order space interaction as described in [14]

Security is provided firstly as an effect of the localised nature of spaces coupled with the agent-join mechanisms. Within the space there is need for a more sophisticated policy mechanism to regulate access, update and the trust of the information at both invidiual triple and larger RDF graph structure levels [15].

2.3. Applications in M3 Spaces

The notion of application in M3 space differs radically from the traditional notion of a monolithic application. Rather, as a long term vision, we see the applications as possible scenarios which are enabled by certain sets of agents [16] [17] [18]. Thus, we do not see an email application running in M3 space, but we could have a collection of distributed agents present which allow for sending, receiving, composing and reading email. Figure 2 pictorially depicts the relationship between the user, her agents and, in this case, one space, while figure 3 shows the user (via agents) interacting with many spaces.

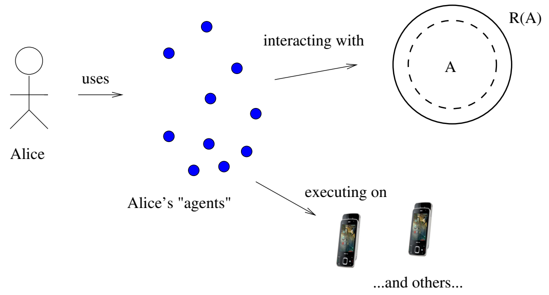

Figure 2. A User's Agents, Devices, Spaces and Information

For this kind of scenario based notion of application, we also would like to know whether the available agents can succesfully execute the scenario. The envisioned model of using this system is that the user has a set of agents which are capable of executing certain scenarios. If a user needs to perform a new scenario that the current set of agents are not capable of executing, she could go and find a suitable agent

1. The current implementation of the concept understands the owl:sameAs concept

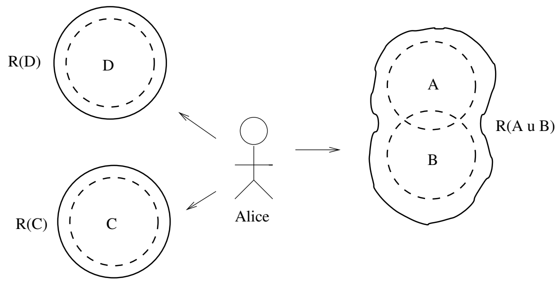

Figure 3. A User and Multiple Spaces

from some directory by describing the desired scenario and the agents she already has.

Thus, we need some formal or semi-formal way of describing agent behavior both with respect to the M3 space and to the environment. While there exists research addressing behavior in multi-agent systems, for example by Herlea, Jonker, Treur and Wijngaards [19], this kind of ad-hoc assembly of agents in order to execute a certain scenario seems to be quite unaddressed in current research. However, slightly similar problems have been addressed in e.g. web service orchestration research [20], but these still seem to concentrate on design-time analysis rather than run-time analysis. As for shorter term, our vision is that sets of existing applications would be enhanced by being able to interoperate and thus allow execution of (automatic) scenarios that would have been impossible or required extensive work to implement without the M3 approach.

3. Bulding space and services it can provide

Despite the lack of universally accepted ontologies for representing information related to buildings and the systems found in them, the application domain still presents some advantages [21]. It is possible to identify a common name for some key concepts, such as ‘temperature’ and ‘humidity’. When it comes to the CO_2 level it already becomes more difficult to agree on a common name. There may also be several different sensors of the same type. For instance, a ventilation machine with heat recovery would normally have at least four temperature sensors that need to be named. Still, the number of manufacturers of such machines is typically not too big (less than ten in Finland) so even though all manufacturers would choose their own names for those sensors, it could still remain manageable. We also believe that as equipment manufacturers start publishing information in the smart space and there are services built upon that information, there may also be more incentive for the manufacturers to start using common ontologies such as the simple example in figure 4; written in a UML or Entity-Relationship style notation. Another approach could be to use ‘forced semantics’ [22] implemented by ‘translation agents’ that would automatically translate information from one ontology to another [23].

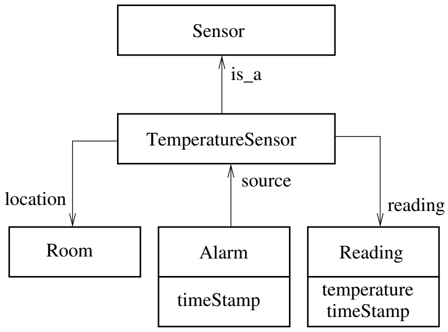

Figure 4. Example of an Ontology (Schema).

Figure 5 shows an example of a small semantic net [24] expressed as an RDF graph for representing sensor values from three different temperature sensors, of which two are located in the same room. This graph satisfies the ontology given in figure 4; the reader should be able to figure the names of relationships (other than object typing) which are not shown for clarity.

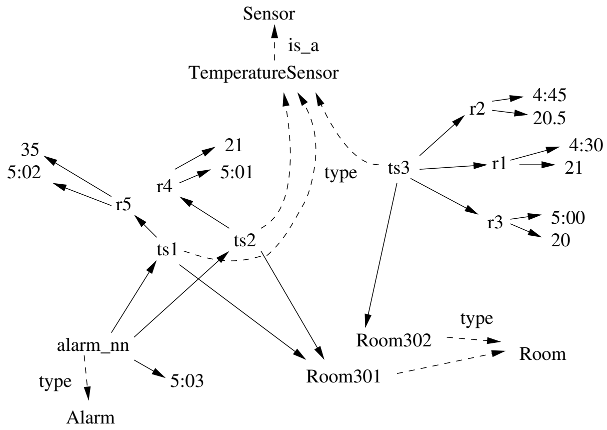

Figure 5. Example of partial semantic net for a home with several temperature sensors.

This limited net also illustrates some basic processing needs, implemented by agents. In Figure 5, sensor *ts3* has produced three temperature readings, which is the beginning of a reading history. With an increasing number of sensors that may store historical information in the space, it becomes necessary to at least implement *cleaning agents* who take care of removing expired information or removing the ‘least useful’ information if memory is filling up. To avoid losing too much information when ‘cleaning’, *summarization agents* become essential. Summarization agents will keep track of minimum, maximum, running average etc. values even after the cleaning agents have removed the original values.

Using figure 5 we can write queries across this graph to obtain readings such as those described above. We nominally use here a graph traversal language such as WQL [25] or

XPath - M3 specifically supports WQL at this time and a SPARQL parser is being developed.

Given a specific temperature sensor (ts2), the query to obtain the current temperature would take the pseudo-code form:

```
ts2 | readings.filter(latest(timestamp).
          temperature
```

Given a specfic room, the average temperature would take the form:

```
Room | ( location-1.readings.
          temperature.asBag() )
          / size(location-1.readings)
```

where the suffix -1 denotes inverse traversal of a link and the functions `latest()`, `asBag()`, and `size()` take their common sense meanings when working with sets (or bags) of values.

Figure 5 also shows the alarm event *alarm_nn* in the space as an example of how to handle anomaly detection. A sensor consistency check agent notices an abnormally great value difference for sensors *ts1* and *ts2* that are in the same room. The presence of a new alarm event can be detected by a user notification agent that takes care of notifying the user about the situation. The user can then take the appropriate action, after which the alarm event is removed manually or automatically when the anomaly is no longer present.

It is quite easy to imagine a great number of other functionality that agents could implement based on the information in the space. However, the purpose of this paper is not to claim that some specific service or functionality is useful as such, the objective is rather to show that Smart Spaces significantly simplify the creation of such services.

Finally, Smart Spaces are not constrained to buildings. They can also cover greater geographical areas and be dedicated for other application domains. One example of such a domain would be publishing weather information collected from private weather stations, ventilation machines etc. for improved local weather forecasts, thereby improved control of heating and cooling in buildings and, as a consequence, improved energy-efficiency as a whole.

4. Implementation

At the Electrical Building Services Centre we have developed an experimental building automation facility, which is able to bridge between the Internet, and a number of building automation protocols. The software platform is based upon OpenWRT (http://www.openwrt.org), a Linux software distribution for embedded systems.

The platform makes it possible to communicate with proprietary building automation protocols, and translate the messages to a common format, namely oBIX. The platform itself is running on inexpensive consumer grade hardware (a wireless router with Universal Serial Bus). So far, we have interfaced to the integration platform a heat exchanger and ventilation machine, a control unit for electrical systems of small buildings, a sensor floor, an electronic water tap and an ‘intelligent’ power outlet.

By connecting together various building automation protocols, we make it possible to combine the functionalities of various subsystems, and create new services that would not be possible without seamless integration of the subsystems. Currently, the subsystems are combined together by the oBIX protocol, which makes it possible to build hierarchal systems by interconnecting the devices on a local level and export the combined information to upper-level systems via oBIX.

However, optimal control of a building’s automation system also needs information from other sources than the various systems located in the building. A simple example is to use weather forecast information from a meteorological website so that the control system can decide to start heating the house during the night (when electricity is cheaper) if the weather forecast says the next day will be colder. Including this type of information from outside the domain of building automation is difficult, if we have to use a building automation specific data format like oBIX.

The Sedvice/M3 architecture is being integrated to the demonstration plaform to make it possible to combine information from various data sources, and to do automated reasoning over it as illustrated in figure 6. Having the building’s information pushed to the SIB along with information about the environment, the users etc., and make the information available to reasoning systems. Reasoning agents might change over time, or be only temporarily available, e.g. if they are located in a visitor’s mobile phone, PDA or similar.

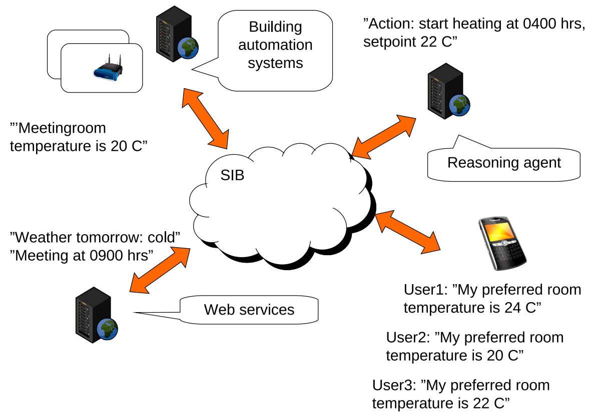

Figure 6. The SIB is an implementation of a data store supporting reasoning over cross-domain information.

The integration to the SIB is not intended to replace the control logic embedded in building automation systems, but rather to facilitate using the information from the building automation systems together with other information.

5. Conclusions

Rather than attempting to define and describe specific use cases, this paper describes an information publishing mechanism that is easily accessible to any third-party solution provider. Such solution providers can provide agents or agent frameworks that implement new functionality. Therefore, our goal is to provide an easy to use basic mechanism that makes it possible to create an open ecosystem where the set of potential applications is open and impossible to predict in advance. Building Automation is a domain where such ecosystems have great potential and can be implemented in real environments.

Acknowledgment

This work has been carried out in the Devices and Interoperability Ecosystem (DIEM) project (www.diem.fi), financed by the Technology Development center of Finland (TEKES) and by the partner organisations of DIEM, and in the AsEMo project financed by the European Regional Development Fund.

References

[1] I. Oliver and J. Honkola, "Sedvice: A triple space computing exploration environment," in *Tripcom workshop*, April, 2008.

[2] ——, "A base-line semantic computation environment for local information spaces," *International Journal of Semantic Computing*, 2009, to appear.

[3] D. Lewis, *Convention: a philosophical study*. Blackwell Publishing, 2002, 0-631-23257-5.

[4] PROMISE, "Volume 3: Architecture reference: Promise messaging interface (pmi)," http://cl2m.com/system/files/private/PROMISE AS Volume 3 Architecture Reference PMI.pdf, 2008, last visited: May 28, 2009.

[5] I. Oliver, "Information spaces as a basis for personalising the semantic web," in *11th International Conference on Enterprise Information Systems*, May 2009.

[6] L. J. B. Nixon, E. Simperl, R. Krummenacher, and F. Martin-Recuerda, "Tuplespace-based computing for the semantic web: a survey of the state-of-the-art," *The Knowledge Engineering Review*, vol. 23, no. 2, pp. 181–212, June 2008.

[7] B. Hayes-Roth, "A blackboard architecture for control," *Artif. Intell.*, vol. 26, no. 3, pp. 251–321, 1985.

[8] A. Passant, ":me owl:sameas flickr:33669349@n00," in *Linked Data on the Web (LDOW 2008), Beijing, China*, April 2008.

[9] K. Idehen and O. Erling, "Linked data spaces and data portability," in *Linked Data on the Web (LDOW 2008), Beijing, China*, April 2008.

[10] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, May 2001.

[11] "Network on terminal architecture," http://www.notaworld.org, 11 2008.

[12] J. Jantunen, I. Oliver, S. Boldyrev, and J. Honkola, "Agent/space-based computing and rf memory tag interaction," in *The 3rd International Workshop on RFID Technology - Concepts, Applications, Challenges (IWRT 2009)*, May 2009.

[13] S. Boldyrev, I. Oliver, and J. Honkola, "A mechanism for managing and distributing information and queries in a smart space environment," *The 1st International Workshop on Managing Data with Mobile Devices (MDMD 2009) 6-7 May, 2009 - Milan, Italy*, 2009.

[14] I. Oliver and S. Boldyrev, "Operations on spaces of information," in *Proceedings of IEEE Conference on Semantic Computation. Berkley, CA.*, September 2009.

[15] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila, "Proteus: A semantic context-aware adaptive policy model," in *POLICY*. IEEE Computer Society, 2007, pp. 129–140.

[16] I. Oliver, E. Nuutila, and S. Törmä, "Context gathering in meetings: Business processes meet the agents and the semantic web," in *The 4th International Workshop on Technologies for Context-Aware Business Process Management (TCoB 2009)*, May 2009.

[17] J. Honkola, H. Laine, R. Brown, and I. Oliver, "Cross-domain interoperability: a case study," September 2009.

[18] S. Balandin, I. Oliver, and S. Boldyrev, "Distributed architecture of a professional social network on top of m3 smart space solution made in pcs and mobile devices friendly manner," in *Proceedings of Ubicomm 2009. Malta*, 2009.

[19] D. E. Herlea, C. M. Jonker, J. Treur, and N. J. E. Wijngaards, *Multi-Agent System Engineering*, ser. LNCS. Springer, 1999, vol. 1647, ch. Specification of Bahavioural Requirements within Compositional Multi-agent System Design, pp. 8–27.

[20] H. Foster, S. Uchitel, J. Magee, and J. Kramer, "Compatibility verification for web service choreography," in *Proceedings of IEEE International Conference on Web Services, July 6–9 2004*. IEEE, 2004, pp. 738–741.

[21] H.-G. Kim and Anseo-Dong, "Pragmatics of the semantic web," in *Semantic Web Workshop. Hawaii, USA*, 2002.

[22] S. Staab, "Emergent semantics," *IEEE Intelligent Systems*, vol. 17, no. 1, pp. 78–86, 2002.

[23] B. Hu, S. Dasmahapatra, P. H. Lewis, and N. Shadbolt, "On capturing semantics in ontology mapping," in *AAAI*. AAAI Press, 2007, pp. 311–316.

[24] M. A. Rodriguez and J. Bollen, "Modeling computations in a semantic network," *CoRR*, vol. abs/0706.0022, 2007.

[25] O. Lassila, "Programming Semantic Web Applications: A Synthesis of Knowledge Representation and Semi-Structured Data," Ph.D. dissertation, Helsinki University of Technology, November 2007.