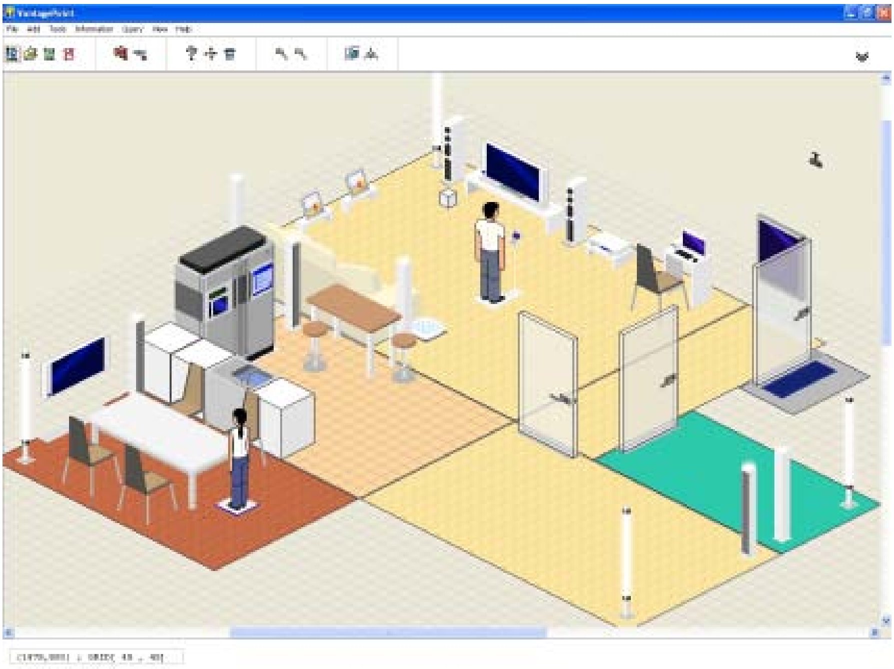
ESPOO 2007 VTT PUBLICATIONS 649



Ilkka Niskanen

An interactive ontology visualization approach for the domain of networked home environments

VTT PUBLICATIONS 649

An interactive ontology visualization approach for the domain of networked home environments

Ilkka Niskanen

ISBN 978-951-38-7033-1 (URL: http://www.vtt.fi/publications/index.jsp)
ISSN 1455-0849 (URL: http://www.vtt.fi/publications/index.jsp)

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 3, PL 1000, 02044 VTT
puh. vaihde 020 722 111, faksi 020 722 4374

VTT, Bergsmansvägen 3, PB 1000, 02044 VTT
tel. växel 020 722 111, fax 020 722 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 3, P.O.Box 1000, FI-02044 VTT, Finland
phone internat. +358 20 722 111, fax + 358 20 722 4374

VTT, Kaitoväylä 1, PL 1100, 90571 OULU
puh. vaihde 020 722 111, faksi 020 722 2320

VTT, Kaitoväylä 1, PB 1100, 90571 ULEÅBORG
tel. växel 020 722 111, fax 020 722 2320

VTT Technical Research Centre of Finland, Kaitoväylä 1, P.O. Box 1100, FI-90571 OULU, Finland
phone internat. +358 20 722 111, fax +358 20 722 2320

Technical editing Maini Manninen

Niskanen, Ilkka. An interactive ontology visualization approach for the domain of networked home environments [Ratkaisumalli verkottuneita kotiympäristöjä kuvaavien ontologioiden interaktiiviseen visualisointiin]. Espoo 2007. VTT Publications 649. 112 p. + app. 19 p.

Keywords Ontology visualization, interactivity, domains, networked home environments, software tools, visualization tools, Web Ontology Language, semantic mark-up language, validation, VantagePoint

Abstract

This study concentrates on the construction process of a software tool which is able to interactively visualize ontologies, particularly in the domain of networked home environments. Special attention in this study is given to the design and implementation of the interaction mechanisms between the user and the visualization. This study is carried out by using a constructive research method supported by literary review and a test case, in which the constructed approach is validated.

In networked home environments domestic devices can interact seamlessly with each other and with-in home and external networks, and users can easily manage home devices, both locally and remotely. Networked home environments are described by ontologies, which are controlled vocabularies that describe objects and the relations between them in a formal way. The Web Ontology Language (OWL) semantic mark-up language was created for publishing and sharing ontologies, and it provides mechanisms for creating all the components of an ontology: concepts, instances, relations and axioms.

The OWL language is complex and looking at an OWL ontology for the first time can be overwhelming. Several visualization approaches have been developed to enhance the understanding of ontologies, but these existing visualization approaches are domain independent, graph based and not suitable for visualizing ontologies that describe networked home environments. The prototype software tool constructed in this study addresses these shortcomings by visualizing ontologies in a domain-specific way. It provides extensive interaction possibilities to help users to communicate with the visualized data.

This approach makes OWL ontologies more interesting and concrete, and above all easier to comprehend.

The main achievement of this study is an ontology visualization tool prototype called VantagePoint. The results of the validation indicate that with VantagePoint it is possible to accurately create and visualize semantic models of real-world networked home environments. Through the extensive query construction features it is possible to access any data contained by the models. Although the validation also reveals some minor deficiencies, it must not be forgotten that VantagePoint is still in a prototype phase and further research and development work will be undertaken.

Niskanen, Ilkka. An interactive ontology visualization approach for the domain of networked home environments [Ratkaisumalli verkottuneita kotiympäristöjä kuvaavien ontologioiden interaktiiviseen visualisointiin]. Espoo 2007. VTT Publications 649. 112 s. + liitt. 19 s.

Avainsanat Ontology visualization, interactivity, domains, networked home environments, software tools, visualization tools, Web Ontology Language, semantic mark-up language, validation, VantagePoint

Tiivistelmä

Tämän tutkimuksen tarkoituksena on rakentaa ohjelmistotyökalu, joka visualisoi verkottuneita kotiympäristöjä kuvaavia ontologioita. Tutkimuksessa keskitytään erityisesti käyttäjien ja visualisoidun datan väliseen vuorovaikutukseen. Tutkimus toteutetaan käyttämällä konstruktiivista tutkimusotetta, jota tukee kirjallisuuskatsaus sekä validointi.

Verkottuneessa kotiympäristössä laitteet ovat saumattomassa vuorovaikutuksessa sekä toistensa että kodin ulkopuolisten verkkojen kanssa. Käyttäjät voivat helposti hallita laitteita niin paikallisesti kuin etäältä käsin. Verkottuneet kotiympäristöt mallinnetaan ontologioilla, jotka kuvaavat formaalilla tavalla ympäristössä olevia objekteja sekä niiden välisiä suhteita. OWL-kieli mahdollistaa ontologioiden julkaisemisen ja jakamisen sekä tarjoaa mekanismit ontologioiden sisältämien komponenttien kuvaamiseen.

Työskentely ontologioiden kanssa on monimutkaista ja etenkin aloittelevilla OWL-kielen lukijoilla on usein suuria ymmärtämisvaikeuksia. Datan visualisoinnin on todettu merkittävästi edesauttavan datan sisältämän informaation omaksumista ja siksi useita ontologioita visualisoivia työkaluja on kehitetty. Nämä työkalut eivät kuitenkaan huomioi ontologioiden sisältämän datan luonnetta, vaan hyödyntävät pelkästään graafipohjaisia visualisointimetodeja. Tässä tutkimuksessa rakennettava visualisointityökalu ottaa huomioon ne erityisvaatimukset, joita verkottuneita kotiympäristöjä kuvaavat ontologiat asettavat. Se tarjoaa käyttäjille kattavat vuorovaikutusmahdollisuudet, joiden avulla he voivat kommunikoida visualisoidun datan kanssa. Lisäksi työkalu tekee OWL-ontologioista mielenkiintoisempia, konkreettisempia sekä ennen kaikkea ymmärrettävämpiä.

Tämän tutkimuksen tärkein saavutus on ontologiavisualisointityökalun prototyyppi VantagePoint. Suoritetun validoinnin tulokset osoittavat, että VantagePointin avulla on mahdollista luoda ja visualisoida olemassa olevia verkottuneita kotiympäristöjä kuvaavia semanttisia malleja. Työkalun prototyypin jatkotutkimus- ja kehitystyö pyrkii korjaamaan validoinnin aikana ilmitulleita puutteita sekä laajentamaan työkalun käyttömahdollisuuksia entisestään.

Preface

This thesis was written at VTT, the Technical Research Centre of Finland, in the Software Architectures and Platforms unit. The work was carried out as a part of Amigo (Ambient intelligence for the networked home environment) project, which is a collaborative EU project with fifteen partners.

I would like to present my thanks to my supervisors, Prof. Petri Pulli at University of Oulu, and Lic.Sc. (Tech.) Julia Kantorovitch at VTT. Petri helped me to define my research subject and Julia gave valuable comments concerning the content of my thesis through the writing process. Furthermore, I would like to thank Toni Piirainen, who cooperated with me in the construction process, M.Sc Jarmo Kalaoja, who gave valuable information during the construction process and all my colleagues at VTT for the help and support they gave during the writing of this thesis.

Contents

Appendices

Abbrevations

2D/3D	Two dimensional/Three dimensional
API	Application Programming Interface, interface to an existing application
CE	Consumer Electronics
ID	Identification
IO	Input/Output
MVC	The Model-View-Controller, software architecture pattern
OWL	Web Ontology Language, language for defining machine interpretable vocabularies specified by W3C
OWL-S	Ontology built on top of Web Ontology Language for describing Semantic Web Services
PC	Personal Computer
PDA	Personal digital assistant, is a device that include some of the functionality of a computer
PNG	Portable Network Graphics, is a bitmapped image format that employs lossless data compression
RDF	Resource Description Framework, is a World Wide Web Consortium specification for a metadata model
RDQL	RDF query language, is able to retrieve and manipulate data stored in ontologies
SHriMP	Simple Hierarchical Multi-Perspective, is a domain independent visualization technique
SOA	Service oriented architecture that defines the use of services to support the requirements of users

SOUPA	Standard Ontology for Ubiquitous and Pervasive Applications, is a set of ontologies for supporting pervasive computing applications.
SPARQL	SPARQL Protocol and RDF Query Language, a query language that is able to retrieve data from ontologies. Is on track towards W3C Recommendation status
URI	Uniform Resource Identifier, unique identifier for resources
XML	Extensible Markup Language, information representation technology
W3C	World Wide Web Consortium, standardization organization for web technologies

1. Introduction

As early as 1991, Mark Weiser stated that “the most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it”. Although Weiser presented this claim over 15 years ago, the same idea still seems to be the predominant trend in many of today’s emerging technologies. Terms such as ambient intelligence, pervasive computing and networked home environments are all tied into making computer technology as ubiquitous and as invisible a part of our everyday lives as possible.

In Amigo (2005), a networked home is defined as a home in which several pieces of equipment are connected using an infrastructure, and in which the traditional separation of activities is no longer valid. This networked home leads to many new opportunities, such as ambient intelligence, which enables the complete integration of technology into our environment so people can freely and interactively use it. According to Boekhorst (2002), ambient intelligence refers to the presence of an environment that is sensitive, adaptive, and responsive to the presence of people or objects. In the case of networked home environments, ambient intelligence will improve quality of life by creating the desired atmosphere and functionality via intelligent, personalized interconnected systems and services.

Characteristic of ambient intelligence is that appliances and devices disappear into the environment of the individual; services come into focus instead (Weber et al. 2003). Thus, ambient intelligence systems are often designed using a service-oriented approach, in which devices in the environment provide independent services. By composing distributed services, Service-Oriented Architecture (SOA) makes it possible to supply clients with more complex services. With SOA, services can better adapt to changing situations, taking into account contextual information as well as users’ preferences (Vallée et al. 2005).

Ontologies are broadly used in the context of networked home environments. As Gu et al. (2004) have pointed out, with ontologies it is possible to define and store context information, as well as to model different kinds of physical environments. In addition to physical contexts, ontologies can also be used to describe the services provided by devices, since ontologies give the system the necessary semantics to provide users with enough functionality in a limited user interface without really accessing services themselves (Masuoka et al. 2003). Ontologies can be considered as central to networked home environments, as they carry the meaning, and as Ranganathan et al. (2003) have stated, some of the most important problems in the development of pervasive computing environments can be overcome with ontologies.

An ontology, by definition, is a controlled vocabulary that describes objects and the relations between them in a formal way (Uschoold & Cruninger 1996). Lijun et al. (2006) have added that an ontology resembles a faceted taxonomy, but uses richer semantic relationships between terms and attributes, as well as strict rules about how to specify terms and relationships. Ontologies represent the shared meaning of a domain and they can be used to explicitly describe almost any kind of domain. Thus, they are seen as an enabler for many applications, such as data integration, e-commerce and semantic web services. (Lanzenberger & Sampson 2006.) Although research into ontologies is relatively new and ontology technologies are not fully developed or commercially available, research into ontology is becoming increasingly widespread in the computer science community (Guarino 1998; Broens 2004).

The OWL (Web Ontology Language) semantic mark-up language was created for publishing and sharing ontologies on the World Wide Web. OWL is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. Besides just explicitly representing the meanings of terms in vocabularies and the relationships between those terms, OWL can also be used to implicitly represent inferred terms or entire taxonomies by means of logical reasoners. (Taniar & Wenny 2006, p. 341.) The OWL language provides mechanisms for creating all the components of an ontology: concepts, instances, relations and axioms (Davies et al. 2006, p. 4).

Working with ontologies is often complex. Because of the possibilities of expression with OWL, the language can be hard to understand (Baymani & Strifeldt 2005; Tane et al. 2004). Therefore, there is an urgent need to make ontologies more concrete and easier to comprehend by visualizing them. Spence (2001) has defined information visualization as the process of turning abstract data into a visual shape easily understood by the user, making it possible for him/her to generate new knowledge about the relationships between the data. Visualization of data makes it possible to obtain insight into these data in an efficient and effective way, thanks to the unique capabilities of the human visual system, which enables us to detect interesting features and patterns in a short time (Wijk 2005).

Currently there exist several approaches to visualizing ontologies. For example, a navigation engine called Spectacle has a component called Cluster Map, which is able to visualize ontological data (Fluit et al. 2002). The ontology editor Protégé also has visualization components such as OntoViz and Jambalaya (Mutton & Golbeck 2003). Common to these approaches is that they are domain independent and graph based. Graph-based visualization is a powerful approach to visualizing hierarchical data such as file hierarchies and web sitemaps (Herman et al. 2000). However, as Wehrend & Lewis (1990) have mentioned, the visualization technique should always be relevant to the given problem and support the user's goal in viewing the representation. Thus, it can be stated that the domain of the visualization problem should have an influence on how the data is being visualized, and no single visualization method is capable of effectively visualizing every kind of data.

Information and scientific visualization are constantly affected by new trends produced by other industries, especially the industry of computer games. Computer games have increased in popularity and, in recent years, the financial growth of the computer game industry has made it the driving force in the development of consumer graphics applications and hardware. Although games focus on play and not on knowledge, the type of play a game depicts strongly affects the graphics it requires and each game type facilitates the development of visual thinking concepts. For example, the 3D views commonly used in computer games have been proved to increase the user's sense of wonder and user interaction. Because of the rapid development of the computer game industry, there has been a growing interest in exploiting the techniques used in game graphics in scientific and information visualization. (Rhyne 2000.) Computer games often aim to visualize objects realistically, and it is asserted (Luymes 2000) that, especially when considering computer visualization which aspires to simulate real environments; the realism itself breeds the expectation of accuracy, reliability and authority in the representation.

The influence of computer games has increased the use of interactive elements in the scientific and information visualization applications. With these interactive elements, users can communicate with the visualized data more efficiently and more effectively. Chen et al. (1996) have stated that interactive visualization can help us to understand the results better, since it allows us to visualize the presentations from different perspectives. Besides just changing the perspective, interactive visualization techniques enable the versatile manipulation of the graphical objects through multiple operations such as adding, deleting and positional moving (Chuah & Roth 1996). The ability to change parameters, resolution or representation and to see the effects of these interactions makes it possible for users to test different scenarios and to present "what if" questions. In its most effective form, interactive visualization allows the user not only to edit different parameters in graphical presentations, but to directly manipulate data through the visualization. (Johnson et al. 1999.)

In this study a new approach to interactively visualizing ontologies is constructed. In this approach the specific requirements, set by the networked home domain, are taken into consideration. Special attention is given to the interaction between the user and the visualization. The purpose of the constructed approach is to make ontologies more concrete, interesting and above all easier to comprehend. This approach will help people who are not familiar with ontologies and the OWL language. It will also support the work of application developers that need ontologies and contextual data in their work. The approach is validated in a test case, in which a real intelligent environment is semantically modelled and visualized using the constructed approach.

1.1 Research problem

The purpose of this study is to construct an interactive ontology visualization approach specifically for the domain of networked home environments. The constructed approach aims to make ontologies more understandable by visualizing them realistically and support people who want to gain insights into the data contained by ontologies. At the end of this study the approach is validated by visualizing and semantically modelling a real physical environment. The research problem can be stated thus:

How can ontologies be visualized, particularly in the domain of networked home environments, and how should interaction between the visualization approach and the user be implemented?

To answer this problem, three research questions must be answered:

1. *What special requirements does the networked home domain set for the visualization approach?*
2. *How should context information be presented and managed in the visualization?*
3. *How should the interaction be implemented and managed?*

The first question is answered by analyzing the current literature on networked home environments and discovering the special requirements set by this particular domain, as well as how these requirements can be fulfilled. The answer to the second question is received by analyzing different visualization methods and determining the ones that are most efficient in presenting contextual data and spatial relationships. The third question is answered by defining the necessary operations to establish interaction between the visualization and the user.

The interactive elements in the constructed approach are divided into two separate segments. The interaction elements of the first segment provide the means to manipulate and edit the visualization. This segment contains such interaction operations as adding, deleting, moving and rotating of instances. The second segment considers how users are able to acquire additional information about the visualized ontological models. With the interaction elements provided by the constructed approach, users are able to acquire information about the visualized instances, as well as to retrieve such information from the ontological models that could not be visualized

The final answer to these research questions is an implementation of a prototype software tool which is able to interactively visualize ontologies. The constructed approach should act as an example of how the actual research problem can be solved. To evaluate how well the approach fulfils its requirements, it is validated at the end of this study.

1.2 Research methods

This study was carried out by using a constructive research method, supported by literary review and a test case. The literary review consisted of finding out the theoretical background essential to this study and the assessment of four existing ontology visualization approaches. In the implementation phase, a prototype software tool was designed and implemented and finally, in the test case, the approach was validated.

1.2.1 Literary review

Literary review can serve multiple purposes in research. For a start it provides some theoretical background for the study and develops an understanding of the nature and structure of the topic area (Daymon & Holloway 2002, p. 43). Furthermore, as Hart (1998, p. 27) has stated, one of the most important purposes of literary review is to distinguish what has been done from what needs to be done. In this study the literary review had two objectives: to clarify the theory base of the subject field and to ensure that the work to be done is going to be unique. The literary review was carried out by reading through material relevant to the subject field and becoming familiar with existing ontology visualization approaches in order to ensure that there no similar approaches exist. Four ontology visualization approaches were also examined in more detail and assessed at the end of the literary review.

1.2.2 Constructive research

The constructive research typically builds a new artifact or model which is based on existing knowledge. After the artifact or the model has been constructed, an evaluation of the constructed model should be performed. The new artifact or the new model is constructed in an implementation process which can be divided into three separate states: the initial, the implementation and the target. (Järvinen & Järvinen 2000, p. 105.)

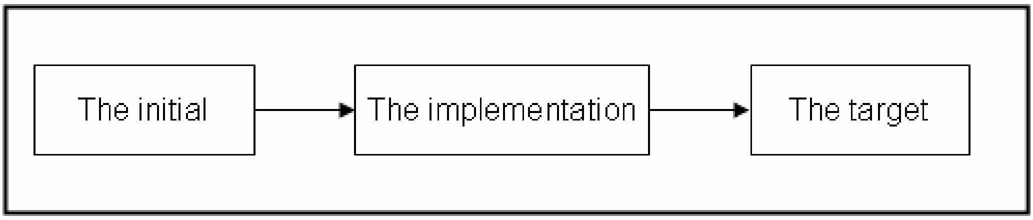


Figure 1. The implementation process (Järvinen & Järvinen 2000).

A description of the initial state models the present situation, its weaknesses and its strengths. A description of the target state is in fact a model of the situation in which we wish things to be when we have implemented our ideas. (Järvinen & Järvinen 2000, p. 105). In this study the initial state was the state where some ontology visualization tools exist, but there is no approach that would take into consideration the particular requirements set by the networked home domain. The target state was the state where a suitable approach for the given domain is constructed and validated.

The implementation work was performed at VTT, in the EU research project Amigo (see Section 2.2), in which the author participated in the actual implementation work. The implementation was divided into two phases: the specification process and the implementation process. Järvinen & Järvinen (2000, p. 106) have stated that the purpose of the specification process is to produce a description of the target state. In this study the specification process included two sub-processes: the requirements specification process and the design process. The implementation process was carried out by programming a prototype software tool according to the specification process. The work was carried out in an evolutionary manner, which means that each process was done in parallel.

1.2.3 Test case

Validation by definition is the determination of the correctness of the software produced from a development project with respect to user needs and requirements (Adrion et al. 1982). Kleijnen (1995) has said that the validation of a simulated model is concerned with determining whether the simulation model is an accurate representation of the system under study. In this study the validation was carried out in a case in which a real physical environment was semantically modelled and visualized using the constructed approach. The object of the validation was to find out how well the application was able to visualize and to simulate a real physical environment and how accurate the resultant model actually was. The ability of the constructed approach to find services contained in the environment was also tested.

Careful planning and clearly stated objectives are critical to successful validation. (Adrion et al. 1982). Therefore the validation process and the acceptability criteria were defined exactly before the validation was carried out. After the validation, the results were compared to the acceptability criteria, and how well the approach fulfilled its requirements was judged.

1.3 Research scope

This study focuses on the visualization of ontologies. To be more exact, it examines the construction process of an approach which is able to interactively visualize and model data described by ontologies, particularly in the domain of networked home environments. Although the construction process and the resulting approach are examined thoroughly, special attention is given to the design and implementation of the interaction mechanisms between the user and the visualization.

The context of networked home environments comprises at least houses, rooms, devices, services and persons. To be able to accurately simulate networked home environments, these elements must be included in the visualization process and thus they are also within the research scope. However, the accurate examinations of ontologies defining the context vocabulary and the hierarchy of different device types are restricted out of the scope of this thesis. Also, the examination of different visualization algorithms and the algorithms used to produce 2D or 3D views are also beyond its scope.

1.4 Structure of the thesis

In Chapter 2, the concepts of networked home environments are examined at a more detailed level. Terms such as ambient intelligence, context awareness, service-oriented architecture, ontology, RDF, OWL, OWL-S, Jena, SPARQL and RDQL are explained. This chapter contains lots of technical details and its purpose is to clarify some of the theory base behind ontologies and networked home environments. In this chapter more detailed information about the Amigo project is also given. The end of the chapter explains what kind of role the ontologies play in networked home environments and what kind of possibilities and visions they enable.

Chapter 3 is about visualization. Some of the basic concepts related to visualization are explained and terms such as information visualization, scientific visualization, 2D/3D displays, and interactive visualization are defined. The influence of the computer games industry on scientific and information visualization is also estimated. The end of the chapter includes the assessment of four existing ontology visualization approaches and a discussion of their suitability for different kinds of visualization problems.

Chapter 4 presents the construction of the approach. First, the most important requirements of the approach are specified. Second, the design specifications are presented and finally, the actual implementation work with its evolutionary nature is described. At the end of the chapter the user interface and the most important user interaction mechanisms of the constructed approach are presented.

In Chapter 5, the validation process of the approach is described. One validation case is presented by describing the phases and the results of the validation in detail. In addition, the possible deficiencies that were exposed during the validation are explained. At the end of the chapter there is a report on what kind of reactions the constructed approach received in the Amigo project review, which was held in Eindhoven, in the Netherlands.

In Chapter 6, a summary of the construction process is presented. The constructed approach is discussed as a whole and its present usage is described in detail. In addition, the overall results of the study are explained and the approaches’ abilities and limitations to visualizing the ontologies of other domains are estimated. Finally, the limitations of the study, suggestions for future research and conclusions are presented.

Three appendices are at the end of the study. An example OWL-S service description is found in Appendix A. Appendix B includes the semantic model of the environment used in validation. Finally, some excerpts from the Amigo project review report, relating to the constructed approach, can be found in Appendix C.

2. Ontologies and networked home environment

This chapter describes some of the basic concepts and terminology relevant to ontologies and networked home environments. Firstly, the concept of a networked home environment is explained and secondly, terms such as ambient intelligence, context awareness system, service oriented architecture, ontology, RDF, OWL, OWL-S, Jena, SPARQL and RDQL are clarified. The end of the chapter explains the nature of the relationship between ontologies and networked home environments. There is also an argument as to why ontologies play such an important role in the development of networked home environments.

2.1 Networked home environments

The concept of the home has changed during recent decades, and it has become a very important and significant site for technological development. Venkatesh et al. (2003) have examined the changing role of home environments in their study and identified seven centres where home life is possible (see Figure 2). In the 1950s, the concept of the home was in terms of an activity centre, and most technologies in the home were targeted toward specific household activities relating to cleaning, meal preparation and other various household activities where technologies were primarily labour-saving or time-saving devices. However, in recent years new media and information technologies, and the internet in particular, have begun to transform the home dramatically. The home is now viewed as a shopping centre, as a communication centre, as an information centre and as a learning centre. This recent development has contributed significantly to reconfiguring the home in terms of networks.

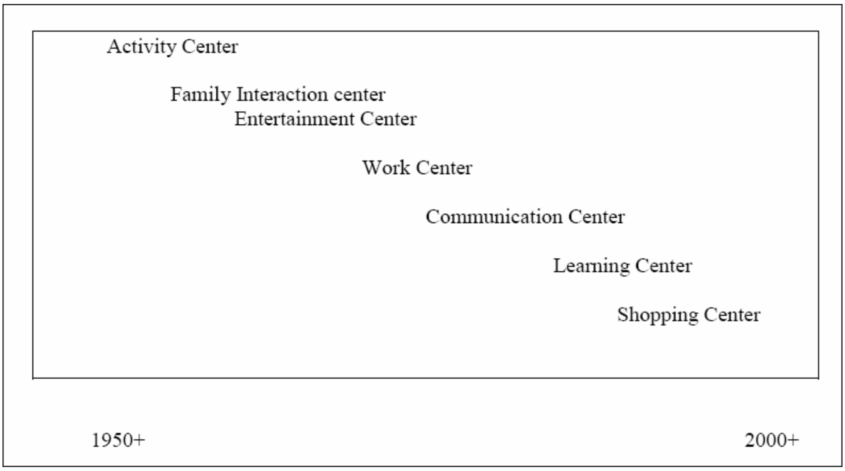


Figure 2. The networked home and the concept of centres (Venkatesh et al. 2003).

During the past decade, numerous definitions for networked home environments have been presented. For example, Petriu et al. (2000) have stated that in networked home environments domestic devices can interact seamlessly with each other and in-home or external networks, and users can easily manage home devices, both locally and remotely. Venkatesh et al. (2003) have emphasized the social aspect of networked home environments in their study and defined it as an internal household network which primarily consists of network relationships with family and friends and social circles, and an external network connecting the home to outside agencies such as schools, shopping centres, work/office, and other civic/community centres. In this study the focus is on internal (in-home) connections between devices, managed by the networked home system.

Figure 3 presents a concrete example of a home network in which devices are connected to each other in a progressive way. In such a networked environment, domains such as PC, CE, mobile and home automation are intertwined. Some of the devices are connected using a wired backbone, while different portable devices such as mobile phones and PDAs are connected through a wireless infrastructure, thus extending the home environment. A home control centre controls multiple home devices, which are connected to a home control network.

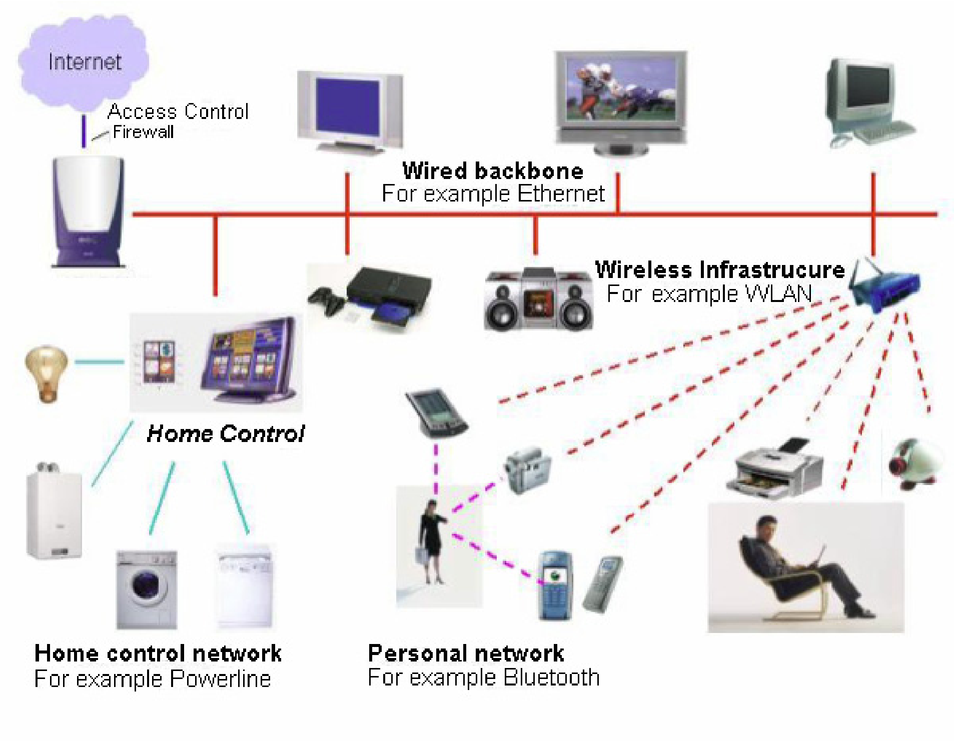


Figure 3. Example of home network where different domains are intertwined (Amigo 2005).

Networked home environments have been a popular subject of research, mostly because of their enormous possibilities to improve peoples´ lives. An essential aim for networked home environments is to support long-term and low-pace communication and interaction between people that have close emotional ties (Rocker et al. 2005). For example, by connecting mobile devices to the home network, our feeling of being at home can be extended when we are on the move. (Amigo 2005.) Furthermore, in networked home environments, user-friendly, intelligent and meaningful interfaces to handle home information and services can be offered to home residents (Kalaoja et al. 2006). An important reason to develop networked home systems is also their ability to improve the quality of life by automating home duties such as household tasks, cleaning etc., and thus increasing people’s free time. (Rocker et al. 2005.)

However, the networked home environment does not appraise just positive feelings. Some people consider the idea of pervasive computing as oppressive and feel it threatens their autonomy. Thus, a lot of research considering user preferences and requirements has been carried out in order to ensure that networked home environments are being developed in the right direction. For example, studies have shown that people want to maintain control over their environments by being able to shut the networked home system down and use everything conventionally at any time (Ringbauer et. al. 2003). People also want to have well-defined responsibilities; they want to be responsible for their children and to control and protect them from inappropriate entertainment and information (Rocker et al. 2005).

A networked home environment must have awareness of its context in order to be able to operate properly. For example, the home system needs to know in which room certain devices are located or in which room residents of the house currently are. **Context awareness** is a research area that uses context information to improve the operation of applications (Ensing 2002). Dey & Abowd (2000) have studied context aware computing in their research and discovered four primary context types: location, identity, time and activity. These context types not only answer the questions of who, what, when, and where, but also act as indices into other sources of contextual information. For example, given a person's identity, we can acquire many pieces of related information such as phone numbers or relationships to other people in the environment. With an entity's location, we can determine what other objects or people are near the entity and what activity is occurring near the entity. With these four primary pieces of context, it is possible to characterize different situations occurring in a given context.

Context-aware applications are needed in order to be able to process context information. Context-aware applications dynamically change or adapt their behaviour based on the context of the application and the user (Brown et al. 1997). Furthermore, Dey & Abowd (2000) have specified that "a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task". So besides just location, a context-aware application needs to process context information from all four primary context categories listed in the previous paragraph. As Schilit et al. (1994) have concluded: context awareness includes three elements: where you are, who you are with and what resources are near.

Ambient intelligence is one of the most important opportunities enabled by the networked home environment. Ambient Intelligence refers to the presence of an environment that is sensitive, adaptive, and responsive to the presence of people or objects (Boekhorst 2002). Weber et al. (2003) have researched ambient intelligence in their study and stated that, "Ambient intelligence is the vision that technology will become invisible, embedded in our natural surroundings, present whenever we need it, enabled by simple and effortless interactions, attuned to all our senses, adaptive to users and context and autonomously acting". Research into networked home environments is closely linked to ambient intelligence, since they both are dealing with the same basic problems. As Boekhorst (2002) has said, ambient intelligence aims to improve quality of life by creating the desired atmosphere and functionality via intelligent, personalized inter-connected systems and services.

A primary feature of ambient intelligence is that appliances and devices disappear into the environment and services come into focus (Weber et al. 2003). **Service-Oriented Approach** (SOA) is a framework that supports communication between software modules independently designed, developed and deployed. In an SOA, services are self-contained, modular applications deployed over standard middleware platforms that can be described, published, located, and invoked over a network. SOA enables devices to connect to any existing network, communicate, discover services, publish their own services and start interacting in the home transparently. The basic idea is that each component can be seen at the same time as a service provider and a service client. For example, a washing machine can provide information about its status to a home controller and it can be a client when connecting via the Internet to an electrical power supplier to negotiate quantity, price and timeframe for power usage. (Aiello et al. 2005.) The benefits of SOA are its ability to make systems evolve as the networked environment changes, and to enable dynamic integration of application components deployed in the diverse devices of today's wireless networks (Georgantas et al. 2005.)

2.2 The Amigo project

Concepts such as networked home environments, context awareness and ambient intelligence are among the central topics of an EU research project, Amigo, in which the implementation work for the prototype visualization tool constructed in this study was performed. In the Amigo project, fifteen of Europe's leading companies and research establishments in mobile and home networking, software development, consumer electronics and domestic appliances have joined together to realize the full potential of home networking. The aim of the Amigo project is to research and develop interoperable middleware and intelligent user services for the networked home environment, which offers intuitive and personalized interaction by providing seamless interoperability of services and applications. Furthermore, the Amigo project intends to show the attractiveness of networked home systems by creating and demonstrating prototype applications improving everyday life, addressing all vital user aspects: home care and safety, home information and entertainment, and extension of the home environment by means of ambience sharing for advanced personal communication. (Amigo 2005.)

To demonstrate the full potential of home networking to improve people's lives, numerous scenarios are defined within Amigo project. These scenarios illustrate the actual functionality of a networked home environment system and are an important part of the research work, since they make the concept of networked home environments more concrete and understandable. By defining different scenarios the possibilities of networked home environments can be exposed to people that are not familiar with this particular research topic. In many cases, scenarios also serve as an actual starting point for the research work on networked home environments. In the following, two short examples taken from the Amigo (2005) project description are presented.

Scenario 1: A house resident (here named Maria) is having breakfast and requests to watch the news on the digital TV set. The system displays a summary of the main news with Maria's preferences (for example sport, local news and technology). Maria decides to download some of the most interesting news to her personal device, to review it while she is going to work. This information is automatically stored in her personal device, which is in the bedroom.

Scenario 2: After work Maria comes home. The front door recognizes her and lets her in without the need for her to grope for keys. A photo frame shows that Robert is home. She wants to start cooking quickly as she is a bit late today. Maria is cooking a new recipe downloaded from the internet. Maria continues cooking. She's wondering about her son Robert and asks the home system to connect to him. His image playing in the bedroom appears on the same screen as the recipe. She talks to him and she asks Robert to come to the kitchen and help set up the dinner table.

These short scenarios illustrate many of the advances the technologies developed in the Amigo project can bring. For example, in the first scenario the intelligent home is able to identify the home resident and show the news summary according to the preferences of this particular person. The home system is also able to automatically store the news in Maria's personal device. The second scenario illustrates the context awareness of the intelligent home, as the system is able to identify the room that Robert is in and connect Maria to him. In general, these scenarios demonstrate the ability of a networked home system to adapt its functionality according to current situations. The system is aware of the locations and preferences of the home residents and is able to provide the user with the best services possible.

In a nutshell, the Amigo project provides solutions for the major problems that are encountered in the use of home networking today. The project aims to improve the usability of a home network by developing open, standardized, interoperable middleware, and to improve its attractiveness by developing interoperable intelligent user services. Furthermore, prototype applications are built within the project to show the end-user usability and attractiveness of such an intelligent home system. (Amigo 2005.)

2.3 The semantic framework

Research on ontology is becoming increasingly widespread in the computer science community. The importance of ontologies is being recognized in many diverse research fields of computer science and ontologies are now gaining a specific role in, for example, artificial intelligence, computational linguistics, and database theory (Guarino 1998). A commonly agreed definition of ontology, made by Gruber (1993), is the following: "An ontology is an explicit and formal specification of a conceptualisation of a domain of interest". Furthermore, ontology is defined as a controlled vocabulary that describes objects and the relations between them in a formal way; ontology resembles faceted taxonomy but uses richer semantic relationships between terms and attributes, as well as strict rules about how to specify terms and relationships (Uschoold & Cruninger 1996; Lijun et al. 2006*)*.

A distribution of ontologies can be distinguished between top-level ontologies, domain ontologies, task ontologies and application ontologies. Top level ontologies describe general concepts like 'space', 'time' or 'action'. These concepts are independent of a particular problem or domain. Domain ontologies and task ontologies describe the vocabulary related to a generic domain or activity by specializing the terms introduced in the top-level ontology. Finally, application ontologies describe concepts depending on both a particular domain and task, which are often specializations of both the related ontologies. (Guarino 1997.)

As discussed in the introduction, ontologies are used for numerous different purposes. Currently perhaps the most popular research area of ontologies is their essential role in the development of the semantic web. **The semantic web** is not a separate web but an extension of the current one in which information is given well-defined meaning, better enabling computers and people to work in cooperation. With the help of ontologies, computers will "understand" the meaning of semantic data on a web page by following links to specified ontologies. The most typical kind of ontology for the semantic web has a taxonomy which defines classes of objects as well as relations among them and a set of inference rules. Inference rules can be demonstrated with the following example: "If a city code is associated with a state code, and an address uses that city code, then that address has the associated state code." (Berners-Lee et al. 2001.)

Doan et al. (2002) have stated that ontologies make possible the widespread publication of machine-understandable data, opening myriad opportunities for automated information processing. Furthermore, they have stated that this data is often fragmented across diverse sources and information processing across ontologies is not possible without knowing the semantic mapping between their elements. The real power of the semantic web will not be realized until people create programs that collect web content from diverse sources, process the information and exchange the results with other programs (Berners-Lee et al.

2001). This is not possible without the use of taxonomies and inference rules. The research work on the semantic web has contributed to the development of networked home environments, since both fields of research deal with ontologies, the interoperability of different technologies, and machine-understandable data.

RDF (Resource Description Framework) is a foundation for processing metadata; it provides interoperability between applications that exchange machine-understandable information on the Web (Lassila & Swick 2000). RDF defines a general data model based on triples: object, property and value, and is suitable for representing a broad range of information. (Fensel et al 2001: McBride 2002). RDF can be written as XML for communication between applications. RDF allows the assertion of collections of simple statements, such as “The sky has the colour blue”. In this case a subject of the statements is “the sky”, a predicate is “has colour” and a value for the predicate is blue. With this kind of simple construct it is possible to describe almost anything. The information model defined by RDF is best represented as a directed graph. An example of an RDF graph can be seen in Figure 4, where elliptical nodes represent resources and arcs represent resource properties. (McBride 2002.)

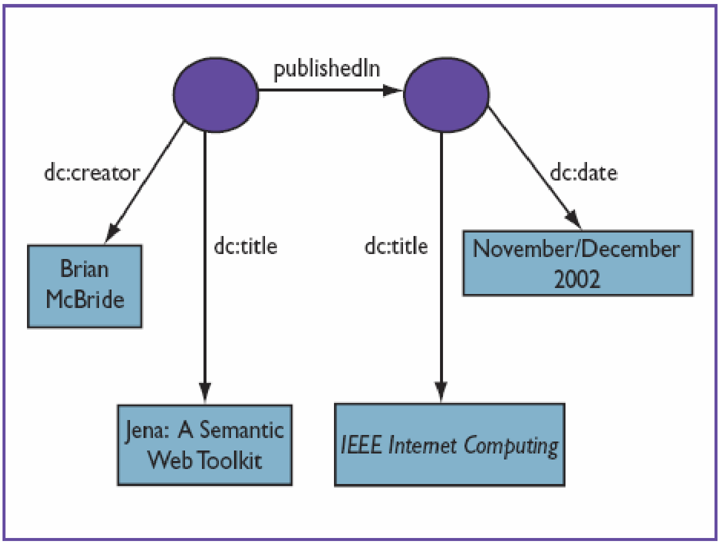


Figure 4. Example RDF graph where elliptical nodes represent resources and arcs represent resource properties (McBride 2002).

RDF was the first ontology description language developed for the semantic web. However, the lack of a capability to describe the semantics of concepts and relations beyond those provided by inheritance mechanisms makes it a rather weak language. For example, while support for modelling of ontological concepts and relations has been extensively provided in RDF, the same cannot be said about the modelling of ontological axioms, which are the key ingredients in ontology definitions and one of the major benefits of ontology applications. RDF offers only the most basic modelling primitives for ontology modelling. However, it must be remembered that RDF was not meant to be the definitive answer to all knowledge representation problems, but rather an extensible core language. (Staab et al. 2000.) The true value of RDF is that many of the posterior and more sophisticated ontology description languages are developed on the basis of RDF.

OWL (Web Ontology Language) is one of the semantic mark up-languages that build on the RDF. OWL is created for publishing and sharing ontologies on the Web and it is intended to be used when the information contained in documents needs to be processed by applications as opposed to situations where the content only needs to be presented to humans. OWL has more facilities for expressing meaning and semantics than XML or RDF, and thus OWL goes beyond these languages in its ability to represent machine interpretable content (Taniar & Wenny 2006, p. 4). OWL is also used to define context ontologies, since it has the capability to support semantic interoperability to exchange and share context knowledge between different systems (Gu et al. 2004). OWL can be also used to implicitly represent inferred terms or whole taxonomies by means of logical reasoners (Taniar & Wenny 2006, p. 341). Since OWL is based on XML, it has a strong ability that can be shared and exchanged between different types of computers using different types of operating system and application languages (Aref & Zhou 2005).

The OWL language provides mechanisms for creating all the components of an ontology: concepts, instances, properties and axioms (Davies et al. 2006, p. 4). Properties define relationships between classes or between classes and instances (Lehti & Fankhauser 2004). There are two sorts of properties: object properties and datatype properties. Object properties relate instances to instances and datatype properties relate instances to datatype values, for example text strings. Concepts can have super and sub-concepts, thus providing a mechanism for subsumption reasoning and inheritance of properties. Axioms are used to provide information about classes and properties, for example to specify the equivalence of two classes (Davies et al. 2006, p. 4). Figure 5 shows an example of an OWL schema for bibliographic data where a relationship between classes *Person* and *Author* has been defined with a property *hasEditor*.

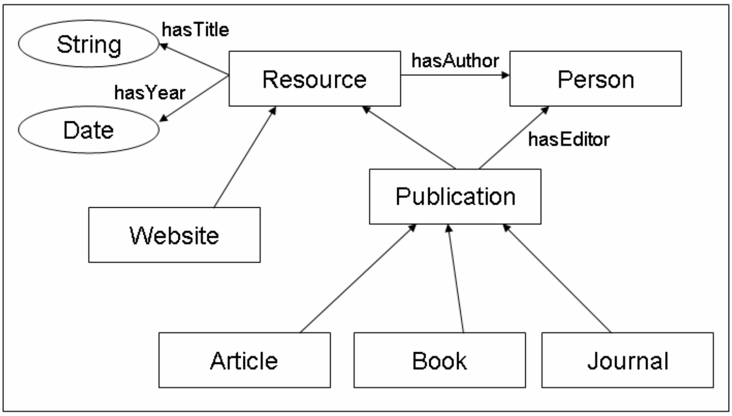
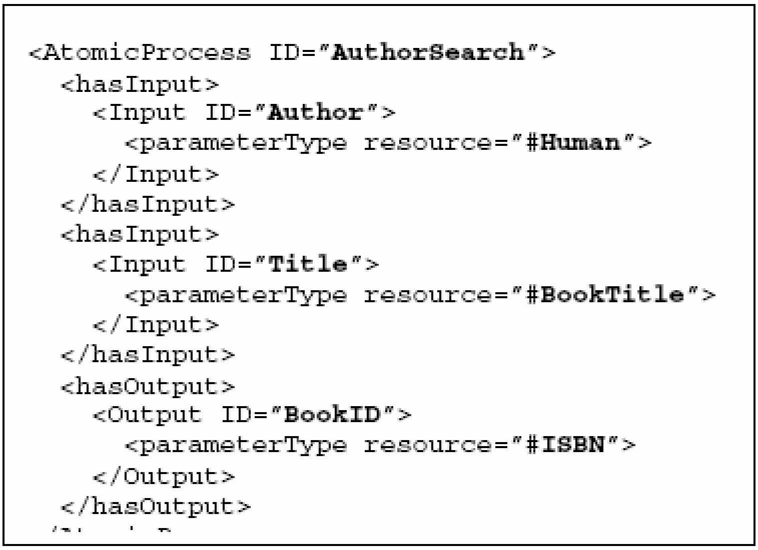


Figure 5. OWL example (Lehti & Frankhauser 2004).

OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full. OWL Lite offers a limited feature set and supports those users primarily needing a classification hierarchy and simple features. For example, it only permits cardinality values of 0 or 1. OWL DL is a superset of OWL Lite and it supports those users who want the maximum expressiveness while retaining computational completeness. OWL DL ensures that all conclusions are guaranteed to be computable and all computations will finish in a finite time. OWL DL includes all OWL language constructs, but they can be used only under certain restrictions. (McGuinness & van Harmelen 2004.) Finally OWL Full, a superset of OWL DL, removes some restrictions from OWL DL but at the price of introducing problems of computational tractability (Davies et al. 2006, p. 4). OWL Full is meant for users who want maximum expressiveness with no computational guarantees. OWL Full provides much more expressive constructs than OWL Lite but it is also less predictable, so ultimately it is up to the ontology developer to decide how much expressiveness is really needed. (McGuinness & van Harmelen 2004.)

Ontology Web Language for Services (OWL-S) is a service description language which can be used to define services hosted by devices. Although OWL-S was originally developed for the semantic web, it nowadays has numerous areas of application. In this study a general presentation of OWL-S is given in order to clarify the basic structures of the language. OWL-S provides means for describing services in an unambiguous and **computer**-interpretable form. The machine-interpretable descriptions of services enhance the automation of service discovery and selection. OWL-S offers an easily extensible generic model, in which each operation involved is described semantically in terms of inputs/outputs. (Mokhtar et al. 2005.) OWL-S is an OWL ontology which can be divided into three sub-ontologies known as the profile, process model and grounding. The profile expresses what a service does, the process model describes how it works and the grounding maps the constructs of the process model onto detailed specifications of message formats, protocols, and so forth. OWL-S also provides a construct called the atomic process, which is characterized primarily in terms of its inputs, outputs, preconditions and effects. (Martin et al. 2004.) Figure 6 shows a simplified OWL-S declaration of an atomic process with its inputs and outputs.



```
<AtomicProcess ID="AuthorSearch">
  <hasInput>
    <Input ID="Author">
      <parameterType resource="#Human">
    </Input>
  </hasInput>
  <hasInput>
    <Input ID="Title">
      <parameterType resource="#BookTitle">
    </Input>
  </hasInput>
  <hasOutput>
    <Output ID="BookID">
      <parameterType resource="#ISBN">
    </Output>
  </hasOutput>
```

Figure 6. OWL-S declaration of an atomic process with its IO specifications (Martin et al. 2004).

2.4 Jena

Jena is a Java-based open source framework to manage RDF graphs and it includes a rule-based inference engine. Its heart is the RDF API, which supports the creation, manipulation, and querying of RDF graphs (McBride 2002). Jena also includes an API for OWL support. Jena defines an interface called OntModel, which supports the kinds of objects expected to be in an ontology: classes, properties, and individuals. (Carroll et al. 2004). Jena is a powerful tool to manage ontologies described by OWL or RDF ontology languages.

Jena provides an effective means of manipulating ontology classes and relations, and performing reasoning. There are also numerous other tools providing ontology reasoning capabilities besides Jena. Pellet, FaCT++, and RacerPro are examples of tools that help ontology testing and provide support for working with ontologies. However, Jena was selected as the most appropriate ontology management tool for this study, because it provides comprehensive support for working with OWL ontologies in a Java environment and it offers an effective means of loading and storing ontologies. In addition, Jena's support for the RDQL and SPARQL ontology query languages was considered important for the purposes of this study.

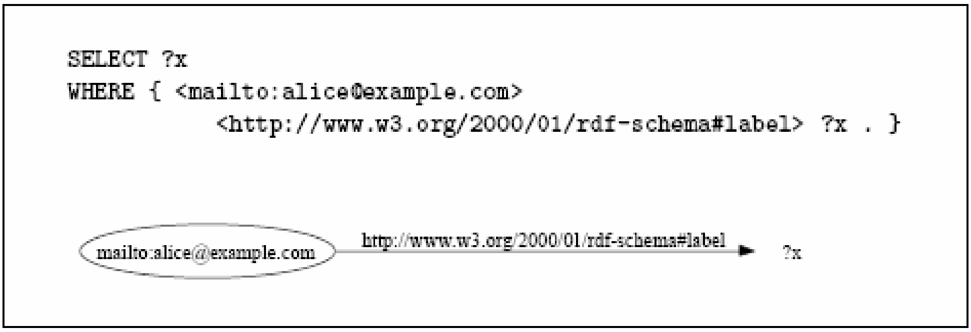
2.5 SPARQL and RDQL

Ontology query languages are used to retrieve data from ontologies defined by different ontology description languages. In this study, two of them, RDQL and SPARQL, are explained in greater detail. RDQL is able to retrieve information stored in the model which is based on a set of triple statements, but it provides no reasoning mechanisms. Another limitation of RDQL is that it does not support disjunction in a query. (Zhang 2005.) However, RDQL is a relatively simple query language and its syntax is similar to that of SQL, where the select clause allows the projection of the variables (Broekstra et al. 2004). An RDQL query consists of a graph pattern, expressed as a list of triple patterns in which each triple pattern is comprised of named variables and RDF values (URIs and literals) (Carroll et al. 2004). Figure 7 presents an example of a simple RDQL query. This triple pattern matches all statements that have the predicate ”http://www.w3.org/1999/02/22-rdf-syntax-ns#type” and the object “http://example. com/someType” and returns the subjects of the matching statements.

```
SELECT ?x
WHERE (?x,  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>,
                                    <http://example.com/someType>)
```

Figure 7. An example of an RDQL query (Seaborne 2004).

Like RDQL, SPARQL also offers a way to retrieve data from ontologies defined using, for example, OWL or RDF. SPARQL is strongly supported by W3C and it represents their latest work. In some parts the development work of SPARQL is still ongoing. SPARQL includes the capability of querying by triple patterns, conjunctions, disjunctions, and optional patterns. The results of SPARQL queries can be ordered, limited and offset in number, and presented in different forms. (Prud'hommeaux & Seaborne 2004). SPARQL is a graph-matching query language where a query consists of a pattern which is matched against a data source. The values obtained from this matching are processed to give the answer. The data source to be queried can be composed of multiple sources. (Perez et al. 2006.) In Figure 8 an example of an RDF structure and an equivalent SPARQL query is presented. As can be noticed, the syntax of a SPARQL query is very similar to the syntax of an RDQL query. Also in this example the defined triple pattern is matched against statements from a data source and the subjects of the matching statements are returned.



```
SELECT ?x
WHERE { <mailto:alice@example.com>
           <http://www.w3.org/2000/01/rdf-schema#label> ?x . }
```

Figure 8. An example of an RDF structure and an equivalent SPARQL query (Harris 2005).

2.6 Ontologies in networked home environments

Although Tim Berners-Lee, the creator of the Web, has mainly been interested in ontologies because of their essential role in the development of the semantic web, he has also envisioned that some day ontologies will enable our phones to tell the TV and stereo to quiet down when they ring (Berners-Lee et al. 2001). A big part of the research work in the area of networked home environments is based on the exploitation of ontologies. With ontologies, some of the biggest challenges of constructing and managing pervasive computing environments, including networked home environments, can be overcome. Ontologies can be used to describe context sources and services, thus facilitating common semantics for context information and service descriptions (Hesselman et al. 2006).

Ranganathan et al. (2003) have researched the use of ontologies in pervasive computing environments and mentioned some reasons why ontologies form a good basis for the constructing work of such intelligent and networked environments. For a start, they have stated that with ontologies different kinds of entities and their properties can be described. For example, applications, services, devices, users and other entities and their properties can be defined with ontologies. In addition, various relations between the different entities can be defined with ontologies. Furthermore, with ontologies it is possible to establish axioms on the properties of these entities that must always be satisfied. Another important aspect is the ability of ontologies to describe contextual information. With ontologies, locations, activities and other context information that may be used by context-aware applications can be defined. The use of ontologies for the representation of context information also ensures that the different entities that use contexts have a common semantic understanding of contextual information. (Ranganathan et al. 2003.)

Chen et al. (2004) have also studied the use of ontologies in pervasive computing environments and described a shared ontology called SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications) which is designed to model and support pervasive computing applications. In their work they have listed some of the factors defining why ontologies are the most suitable approach for representing context information. They have stated that in addition to just describing contextual information, ontologies enable efficient context reasoning and knowledge sharing in open dynamic systems. Ontologies enable the interoperability of services as well as the collaboration of networked services in an unambiguous manner (Chen et al. 2004).

In addition, ontologies can increase the understanding of a pervasive, networked environment among users and the system itself. As Ranganathan et al. (2003) have stated, ontologies enable semantic discovery of entities and allow users to gain a better understanding of the environment and how different pieces relate to each other. Ontologies allow both humans and automated agents to perform searches on different components and to interact with different entities easily. Ontologies can be used to make better user interfaces and they allow networked environments to interact with humans in a more intelligent way. Gu et al. (2004) have concluded that the main advantage of using ontology-based context modelling is the ability to share a common understanding of the structure of the context information among users, devices and services to enable semantic interoperability between users and the system.

3. Visualization

In this chapter the concept of visualization is described in more detail. The discrimination between two visualization categories, information visualization and scientific visualization, is made and both categories are explained. In addition, terms such as '2D visualization', '3D visualization' and 'interactive visualization' are explained. Furthermore, the influence of the computer games industry on scientific and information visualization is estimated. The rest of the chapter is dedicated to the assessment of four current ontology visualization approaches.

Visualization links the two most powerful information processing tools known – the human mind and the modern computer. Visualization is a process in which data, information and knowledge are transformed into a visual form exploiting people's natural strengths in rapid visual pattern recognition. Effective visual interfaces enable us to observe, manipulate, search, navigate, explore, filter, discover, understand, and interact with data far more rapidly and far more effectively to discover hidden patterns. The impact of visualization has been widespread and fundamental, leading to new insights and more efficient decision making. (Gershon & Eick 1998.) However, the power of visualization to exploit human perception presents both a challenge and an opportunity. The challenge is that incorrect patterns can be perceived in visualizations, leading to incorrect decisions and actions. The opportunity is to use knowledge about human perception when designing visualizations. (Mackinlay 2000.) Thus, Duke et al. (2005) have stated in their study that visualizers have to think thoroughly about how people extract meaning from pictures, what people understand from the picture, how pictures are imbued with meaning, and how in some cases meaning arises within a social and/or cultural context. In order to create effective visualizations we must understand the system and the user's needs, and select a proper visualization technique for the given problem. (Gershon & Eick 1998; Wehrend & Lewis 1990.)

Duke et al. (2005) have divided the visualization process into four different phases which can be seen in Figure 9. Visualization begins when someone has data that they wish to explore and interpret (Phase A). In this phase, the dialogue is between domain and visualization experts to explore the problem requirements. In Phase B, the data are encoded as input for a visualization system which, in Phase C, may interact with other systems to produce a representation. In the final phase the visualization is communicated back to the users, who have to assess this against their goals and knowledge, possibly leading to further cycles of activity. (Duke et al. 2005.)

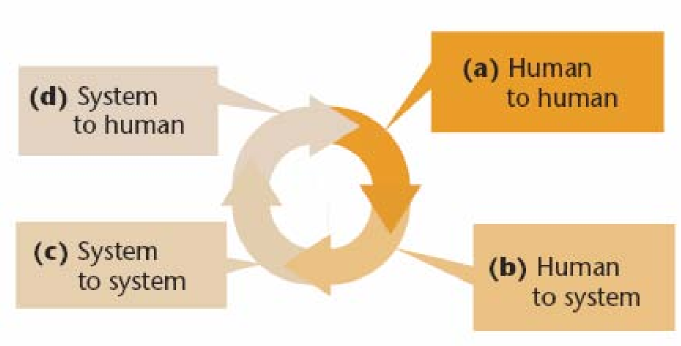
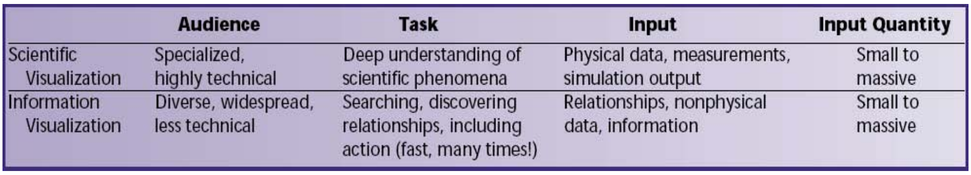


Figure 9. Visualization cycle (Duke et al. 2005).

3.1 Information visualization and scientific visualization

The two sub-categories of visualization are information visualization and scientific visualization. As defined by Mackinlay (2000), scientific visualization focuses primarily on physical data such as the human body, the earth and so on, and information visualization focuses on abstract, non-physical data such as text and hierarchies. Gershon & Eick (1998) have also considered information visualization as a discipline which tries to discover new visual metaphors to represent such information that has no natural and obvious physical representation. Another distinctive factor for these two categories is the target group. As Gershon & Eick (1998) have mentioned, scientific visualization generally serves highly trained scientists, while information visualization approaches have a more diverse user community, with different levels of education, backgrounds, capabilities, and needs. In Figure 10, the main differences between the two visualization areas are presented.



	Audience	Task	Input	Input Quantity
Scientific Visualization	Specialized, highly technical	Deep understanding of scientific phenomena	Physical data, measurements, simulation output	Small to massive
Information Visualization	Diverse, widespread, less technical	Searching, discovering relationships, including action (fast, many times!)	Relationships, nonphysical data, information	Small to massive

Figure 10. Information visualization compared to scientific visualization (Gershon & Eick 1998).

The approach constructed in this study contains aspects of both visualization categories, as the approach not only visualizes abstract textual OWL files, which is typical of information visualization, but also aims to graphically present physical environments and structures, which is characteristic of scientific visualization. In addition, the constructed approach aims to serve both specialized, highly technical users and users with a lower level of background education on ontologies and networked home environments.

Besides discriminating between visualizations based on their audiences, tasks and inputs, different kinds of graphical representation can be categorized according to exploited techniques and presentation methods. In the following section two- and three-dimensional visualizations are introduced.

3.2 2D/3D visualizations

The main idea of visualization is to transform data into a graphical form and usually this is carried out by using computer graphics. There are a number of classifications to categorize the diverse ways to use computer graphics, and the first classification is by the type (dimensionality) of object to be presented and the kind of picture to be produced. The pictures can be purely symbolic (2D graphs) or realistic (representations of real objects). Of course, it is also possible that the same object can be represented in a variety of ways. For example, geographical maps can be presented using 2D or 3D graphics. Software tools creating 3D graphics were relatively rare until the late eighties, since 3D software is far more complex than 2D software and 3D graphics require a great deal of computing power. However, during recent years there has been explosive growth in 3D applications; nowadays they are as common as 2D applications. (Foley et al. 1990, p. 7–21.)

Figure 11 presents an example of a 2D visualization (the left-hand picture) and a 3D visualization (the right-hand picture). As can be seen, in a 2D visualization the environment is shown from directly above, while 3D displays integrate all three dimensions of space into a single display (Smallman et al. 2001). In this example the 3D visualization is populated with realistic icons.

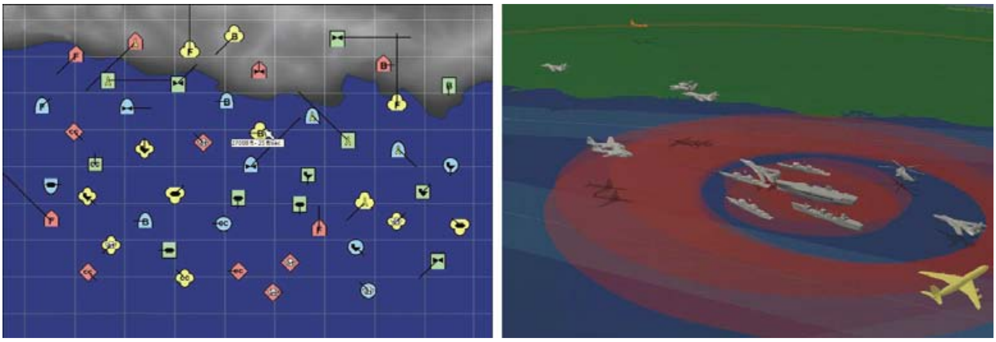


Figure 11. An example of 2D and 3D view (Smallman et al. 2001).

There is no universally applicable truth as to which presentation type is better, 2D or 3D. Springmeyer et al. (1992) mentioned in their study that 2D views are usually used to establish precise relationships, whereas 3D ones are used to present ideas to others and to gain a qualitative understanding. According to several studies, 2D views are considered better for seeing the details of a particular part and navigating or measuring distances precisely (St. John et al. 2001; Tory et. al. 2006). Also, Smallman et al. (2001) have said that the ambiguity of 2D displays is confined to the z dimension alone and that locations in the x and y dimensions are represented faithfully, whereas the ambiguity of 3D displays is because of the distortion of distances and angles inherent in a perspective. Tory et al. (2006) have also listed some of the disadvantages of 3D displays and mentioned that precise navigation and positioning are not possible with 3D displays, except in specific circumstances. Furthermore, it is difficult to display 3D spatial data on a 2D monitor in a way that clearly shows both the overall 3D shape of the object and detailed distances between landmarks.

On the other hand, 3D displays have certain benefits compared to 2D displays. For example, three-dimensional displays are said to be good for gaining an overview of a 3D space, understanding a 3D shape, and navigating approximately in 3D (Wanger et al. 1992). It has also been shown that, since our retinal images are perspective projections of the world, 3D displays may be more ecologically plausible than 2D displays and thus require less interpretive effort. Users also prefer 3D displays simply because of their familiarity and easy feel. With a 3D display necessary information is readily available and easily interpretable (Smallman et al. 2001). 3D displays enable the user to visualize spaces more realistically, and as Luymes (2000) has stated, the realism itself breeds the expectation of accuracy, reliability and authority in the representation, especially when considering computer visualizations which aspire to simulate real environments.

Both of these visualization types have their advantages and disadvantages. Which type should be used depends on the problem domain. Different styles are effective for different situations. As stated, 2D displays are better suited to focused spatial tasks while 3D displays are more impressive, give a better general picture of the object and are useful for understanding a 3D shape. Tory et al. (2004) have stated that since 3D and 2D views serve different purposes, having both visible may benefit certain tasks such as orienting and positioning objects relative to one another. Thus, it might be a good idea to exploit both types of visualization in a graphical presentation. However, one must not forget that although new software makes it easier to produce visualizations, it will be important not to use these new capabilities indiscriminately – only when they are appropriate and convey information effectively. After all, the basis of effective visualizations is always the understanding of the user’s need. (Gershon & Eick 1998.)

3.3 The influence of computer games

In recent years the rapid development of computer game graphics has had a huge influence on scientific and information visualization. The computer games industry has come up with new approaches to presenting data on computer screens in an illustrative and impressive way and succeeded in effectively exploiting different kinds of visualization techniques. The rapid financial growth of the computer games market has made it the driving force in the development of consumer graphics applications and hardware. Popular game environments have also spanned many classifications such as ‘first-person shooter’ games and ‘God games’ that let players create for example a city or even an entire world. Regardless of the type of game, each game type facilitates the development of visual thinking concepts. Thinking visually – in three dimensions – benefits the sense of wonder and user interaction connected with the application of scientific and information visualization technologies. (Rhyne 2000.)

One basic representation of three-dimensional spaces is the isometric perspective, which allows the player to have a general view of the game world at a glance (Fernández-Vara et al. 2005). By using isometric projection, spatial relationships between objects can be seen within wide environments. The computer games ‘The Sims’ and ‘SimCity’ are representatives of such visualization approaches, where the world is presented from an isometric view. These games provide an interactive laboratory for visualizing urban planning and local community involvement (Rhyne 2000). Nowadays many of the approaches implemented for educational or scientific purposes are inspired by computer games. Figure 12 gives an example of such a case. The left-hand picture is a screenshot from the game SimCity and the right-hand picture is a screenshot from an educational program called Virtual U, which simulates running a university. The graphical appearance of the game SimCity was an inspiration for the graphical appearance of the Virtual U, which can easily be seen from the screenshots presented in Figure 12.

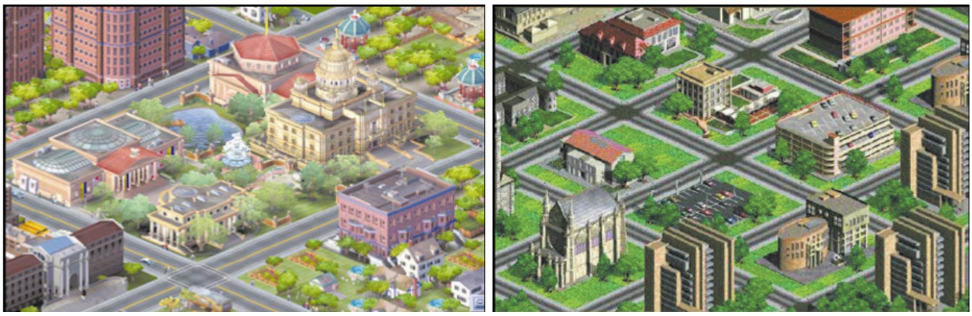


Figure 12. Screenshots from SimCity and Virtual U (Rhyne 2000; Macedonia 2000).

Characteristic of many computer games is that the user is much more than an observer just watching the details of a visualized world. One of the biggest advantages of computer games is their high level of interactive features, which have also spread to scientific and information visualization approaches. (Rhyne 2002).

3.4 Interactive visualization and graphics

Interactive visualization and interactive graphics are both related to man-machine communication. As Chen (2003) has defined it, visualization employs graphics to make pictures that give us insight into certain abstract data and symbols. The pictures may directly portray the description of the data, or present the content of the data in a completely innovative form. According to the visual information-seeking mantra, users should be able to first get an overview of the data, then zoom and filter, and finally obtain details on demand (Shneiderman 1996). So, in addition to just viewing the data, users want to interact with the data in order to gain a better understanding of it. Multiple methods of creating interaction between the user and the graphical presentation have been established to solve this.

Interactive graphics is one of the most natural means of communicating with a computer. Our well-developed two- and three-dimensionally oriented eye-brain pattern-recognition mechanism allows us to perceive and process many types of data rapidly and efficiently if the data are presented pictorially. With interactive graphics we are largely liberated from the tedium and frustration of looking for patterns and trends by scanning many pages of linear text. (Foley & Van Dam 1982, p. 5.) Furthermore, McDonald (1982) has stated that interactive graphics combine the human talents of perception of patterns and judgment using the full context of a problem with a machine's ability to perform rapid, accurate computation. Interactive graphics combine the best features of the interactivity of textual communication with graphical communication. Foley & Van Dam (1982, p. 5) have identified two different kinds of dynamics: motion dynamics and update dynamics. With motion dynamics objects can be moved and rotated with respect to a stationary observer. Equivalently, the objects can remain stationary and the viewer can move around them, pan to select the portion in view and zoom in or out for more or less detail. Update dynamics refers to an actual change of shape, position, or other properties of the objects being viewed. Dynamic interactive graphics offer us a large number of user-controllable modes with which to encode and communicate information: for example the two- or three-dimensional shape of objects and the time variations of different properties.

Interactive visualization allows us to visualize the results or presentation on the fly from different perspectives and thus helps us to understand the results better (Chen et al. 1996). Johnson et al. (1999) have emphasised the possibility of presenting "what-if" questions with the methods of interactive visualization. The ability to edit select values, or to change parameters, resolution or representation, and to see their effects helps scientists to understand the data and to test different scenarios. Scientists want to drive the scientific discovery process by interacting with their data. This interaction usually requires a graphical user interface in order to reveal the multiple or dynamic forms, layers or levels of detail. Interactive visualization can give doctors easier, quicker, and less expensive ways to see the inside of a patient's heart from different angles and layers, and determine whether and to what extent a patient may have a heart disease, for example. At its best, interactive visualization enables all aspects of the modelling and simulation process to be steered, controlled, manipulated, or modified graphically with a single program. (Chen et al. 1996.) Besides just providing efficient access to the data, Johnson et al. (1999) have mentioned that another important aspect of interactive visualization is to provide efficient algorithms for the presentation of data. However, these algorithms are out of the scope of this study.

In conclusion, interactive visualization and graphics deal with providing users with graphical methods of interacting with the data being visualized. This can include multiple different methods of viewing the data from different perspectives and from different angles in order to gain an understanding of the data. In addition, interactive visualization and graphics provide methods of editing different aspects of the visualization and, as Johnson et al. (1999) have mentioned, even of directly manipulating the data through the visualization.

3.5 The assessment of four existing ontology visualization approaches

As mentioned, ontologies can be difficult to understand. Looking at an OWL ontology for the first time can be overwhelming. However, in order to be able to exploit ontological data, we must understand it. Information visualization by definition is the process of turning abstract data into a visual shape easily understood by the user, making it possible for him/her to generate new knowledge about the relations between the data (Spence 2001). As discussed at the beginning of this chapter, visualization of data is perhaps the most effective way to help human beings to gain insight into data, thanks to the unique capabilities of the human visual system. Visualization can also be the key to better understanding of the data contained by ontologies. A visual version of an ontology allows users to visually follow a concept to its nearest neighbours or analyze the overall space for interesting related or unrelated concepts. (O'Leary 1998).

Because of the effectiveness of the visualization, several approaches to visualizing ontologies have been developed. Most of these visualization approaches are embedded in tools that support the development process of ontologies (for example Protégé). The intended users of these tools are ontology engineers that need to get an insight into the complexity of the ontology. Therefore, these tools employ schema visualization techniques that primarily focus on the structure of the ontology, i.e. its concepts and their relationships. (Fluit et al. 2002.) In order to gain an insight into the state of the art in the field of ontology visualization, four visualization approaches are described in detail and assessed in this study. The approaches selected are typical representatives of ontology visualization approaches.

3.5.1 Cluster Map

The Cluster Map ontology visualization approach is intended to bridge the gap between complex semantic structures and their simple, intuitive user-oriented presentation. Cluster Map focuses on visualizing instances and their classifications according to concepts, instead of just the general structure of the ontology. It is well-suited to visualizing ontologies that describe a domain through a set of classes and their hierarchical relationships. In addition, it provides a means of comparing and querying ontologies. (Fluit et al. 2002.) Figure 13 shows an example of a Cluster Map visualization. In the presented graph a collection of job offers, organized into an ontology, have been visualized. Each small yellow sphere represents an instance (a job offer) and the classes are represented as rounded rectangles. Balloon-shaped edges connect instances to their most specific classes and instances with the same class membership are grouped in clusters.

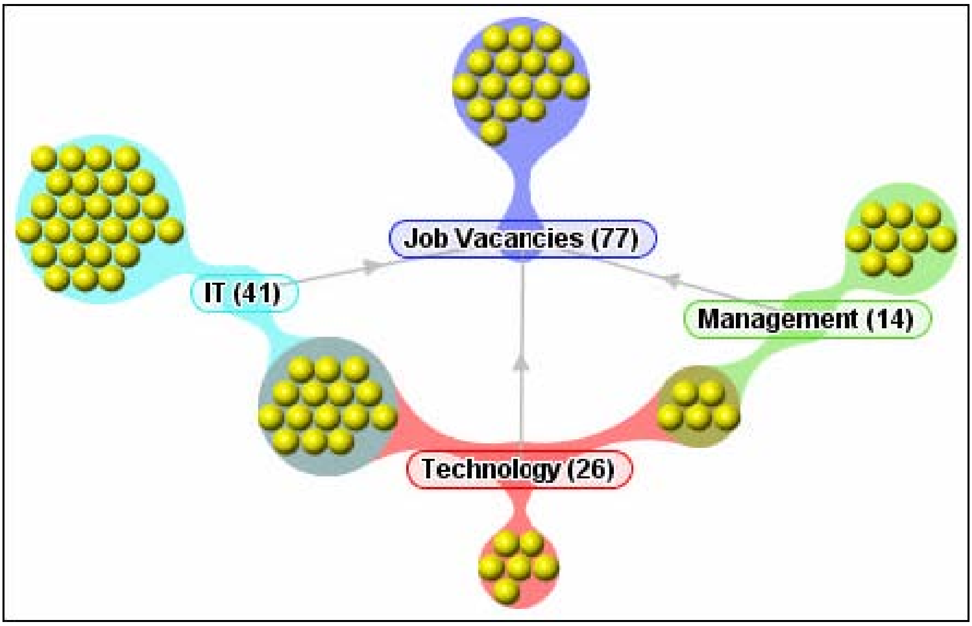


Figure 13. An example of a Cluster Map (Fluit et al. 2002).

The value of this approach lies in its expressiveness. The selected visualization technique makes it possible to easily detect classes and their relationships as well as which items belong to one or multiple classes. By visualizing overlapping classes through shared instances, users can clearly see how the classes relate to each other because of the instances they share. The cardinalities of classes and clusters are also visible. In a Cluster Map visualization, geometric closeness is related to semantic closeness. Classes are semantically close if they share many objects, and in the Cluster Map graph, the more objects two classes share, the closer they are represented. This approach also offers a means to configure the visualization. Classes can be added and removed from the graph and details about individual objects, such as name and URI, can be obtained. Depending on user preference, a double-click on a class either results in the class being expanded or collapsed, or in the whole visualization being replaced with a visualization of its subclasses, providing a way to semantically zoom in on the information. (Fluit et al. 2002.)

Cluster Map has its own unique way of visualizing ontologies. The approach is especially effective in representing the semantic closeness between instances and classes, since geographical closeness is linked to semantic closeness. In addition, Cluster Map reduces visual complexity by grouping instances into clusters. A disadvantage of Cluster Map is that it displays class hierarchical relations only while hiding any relations at the instance level, which may be a drawback in some cases. (Alani 2003.) Cluster Map also offers possibilities to query ontologies. Query formulating in Cluster Map is simply a matter of clicking on checkboxes. All classes are shown in a list and the user can select which classes should be inserted in the query result visualization. The approach enables visual and simple query formulation, but whether this method allows for enough expressivity remains open to question. An average user will probably want to do more than just search for classes. (Ouwerkerk & Stuckenschmidt 2003.)

A big advantage of Cluster Map is its ability to illustratively show query results. Usually, an incomplete query statement gives either a ‘‘no matches’’ message or a long list of partial matches as an answer. In such cases the user gets neither a clear overview of the results nor suggestions for further exploration (Fluit et al. 2002). Cluster Map tackles this problem by visualizing the query answer in such a way that the user is able to analyse alternative solutions. Figure 14 shows an example of such query result visualization. Although none of the query results (yellow spheres) fulfil all the query requirements (3 stars, 2 rooms, 4 persons, located in Loire), the visualization allows the user to see which results fulfil the requirements partially and thus obtain a better insight into the current situation.

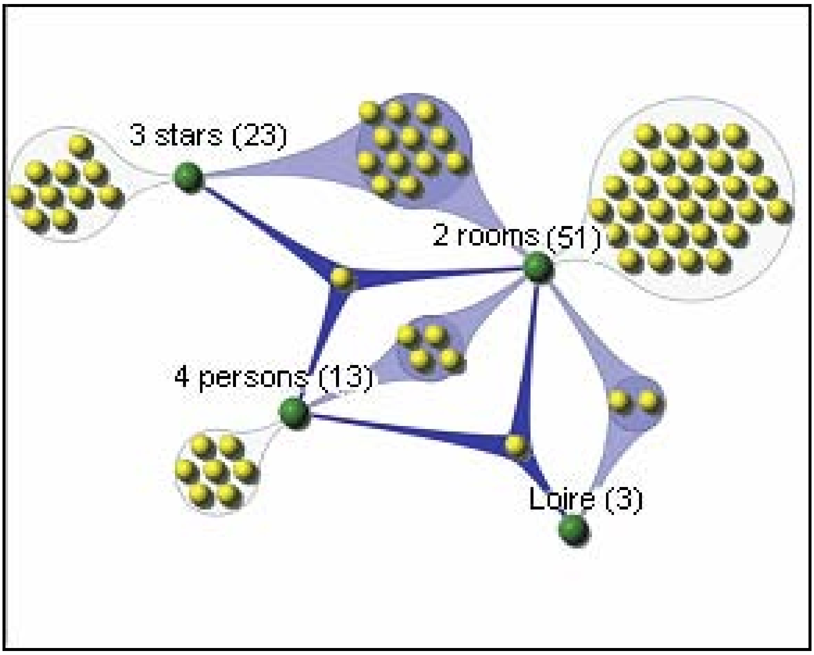


Figure 14. Cluster Map view showing query results (Fluit et al. 2002).

3.5.2 Jambalaya

Jambalaya is a visualization plug-in for the Protégé ontology editor. It uses the SHriMP (Simple Hierarchical Multi-Perspective) 2D visualization, which employs a nested graph view and the concept of nested interchangeable views, combined with geometric, fisheye and semantic zooming. In Jambalaya, classes and instances are represented as nodes in a graph. Instance nodes are distinguished from class nodes using different colours. In Jambalaya, nested nodes are used to depict subclass relationships between classes, as well as instance-of relationships between classes and instances. Role relations between classes or instances are represented using directed links between the related nodes. (Katifori et al. 2006.) Figure 15 presents an example of a Jambalaya visualization. In this view, classes are nested inside their superclasses with instances nested in their instantiating classes. As can be seen, some of the nodes are very small and can only be viewed when the user zooms in. All of the slot dependencies are also shown in this view. (Storey et al. 2001.)

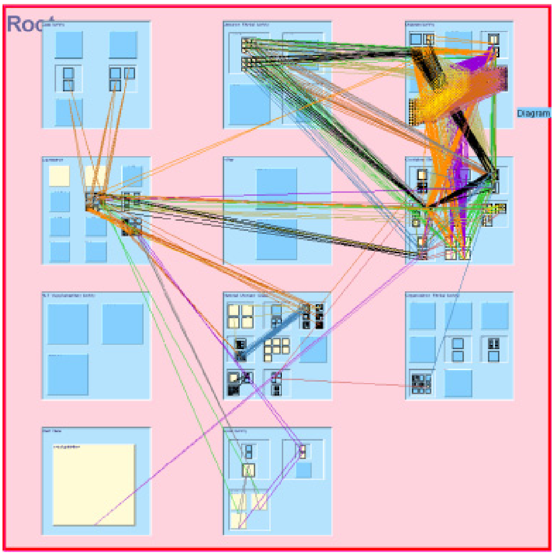


Figure 15. An example of a Jambalaya view (Storey et al. 2001).

A user is able to navigate ontologies using Jambalaya in several ways. For example, when a user selects a class or an instance, the view zooms to the selected class or instance node. When a node is double-clicked, a form is opened with accurate information about the selected node. The visualization also offers navigation buttons such as “back” and “search” which assist in navigation. Jambalaya also contains a search feature, in which whole ontologies or limited parts of them can be searched. (Storey et al. 2001.)

The biggest advantage of Jambalaya is its extensive support for navigation. For example, Jambalaya offers semantic zooming, which means that as the magnification of an object changes, different types of information about the object are shown (Perlin & Fox 1993). In Jambalaya this means that when a user zooms closer, the level of visible details of classes and instances increases. Jambalaya is not focused on one particular visualization technique; it provides a variety of views for users to select from (Ernst et al. 2003). For example, Jambalaya offers a fisheye view of a hierarchical tree which gives an overall picture of the visualized ontology. And by clicking on relevant classes and/or instances, the user can approach a relevant item and get additional information (Ouwerkerk & Stuckenschmidt 2003.) Like most ontology visualization tools, Jambalaya offers little or no support for editing tasks (Ernst et al. 2003).

Jambalaya contains a more advanced keyword search than many of the other ontology visualization tools, allowing the user to search the whole ontology (classes and instances alike) or limit the search scope by specifying the type of searched item (Katifori et al. 2006). Query formulating in Jambalaya is simply a question of entering a simple or complex search pattern, after which a list of relevant results is returned. If the returned list is large, the results can become overwhelming for the user since no visual clues as to where the result belongs in the ontology are presented. This may be a problem when the results are not very clear and more information is needed to decide whether the results are relevant or not. Jambalaya also offers another possibility to make queries. The user can select a desired class or instance from a list of object names, and the corresponding object will be highlighted in the graph. With this method the visualization does help a lot because users can always see where they are in the ontology. A downside of this approach is that the user will have to be familiar with the ontology to know where to click and find the desired result. (Ouwerkerk & Stuckenschmidt 2003.) Having two separate query mechanisms is a good idea as the user can choose whether he/she wants to quickly get some basic information about the ontology by using the graphical query alternative or more sophisticated information by using the more complicated query alternative which is based on search patterns.

3.5.3 OntoViz

Like Jambalaya, OntoViz is also a visualization plug-in for Protégé. OntoViz uses a simple 2D graph visualization method to present the classes and relationships of an ontology. With OntoViz it is possible to visualize the attribute slots, and inheritance and role relations of each class contained by the visualized ontology. OntoViz offers such configuration operations as selecting which classes and instances should be included in a visualization, specifying colours for nodes and edges, and zooming. (Sintek 2003.) Figure 16 shows an example of an OntoViz visualization in which classes and their inheritance relations, instances, and attribute slots can be seen. On the left side of the view is a configuration panel, in which the user can select which elements of the given ontology are shown in the visualization.

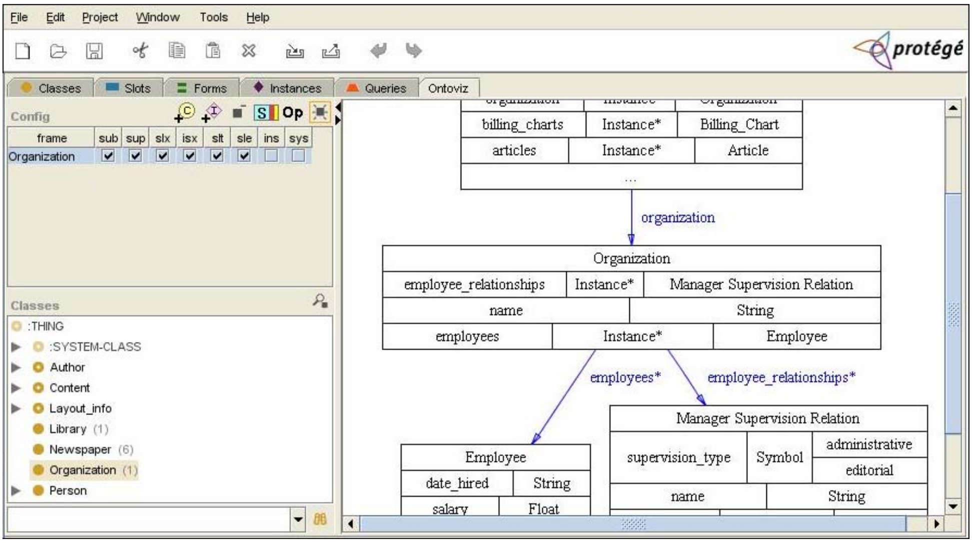


Figure 16. An example of an OntoViz view (Sintek 2003).

The interaction features are restricted to panning and simple zooming. The zooming offered by OntoViz supports only navigation tasks for top down browsing and no semantic zooming features are offered. OntoViz has one layout, which can only be structured based on inheritance relationships. The advantage of OntoViz is that it permits the visualization of several disconnected graphs at once. However, OntoViz is not suitable for visualizing large ontologies, since the visualization does not scale beyond a few hundred entities. In addition, OntoViz does not allow the browsing of multiple relationships. (Ernst et al. 2003.)

Katifori et al. (2006) have evaluated the advantages and disadvantages of OntoViz by studying queries posed to a test group. In general, OntoViz received very negative reactions. For example, all users commented on the lack of interaction and had problems with navigation. Users did not like the fact that they had to drag the scrollbars to navigate. Many of the users found the presentation “poor” and “chaotic” and some users complained about the fact that labels in the visualization are not fully visible, which forces users to guess their meaning. Another big disadvantage of OntoViz was that it did not offer any kind of search tool.

In conclusion it can be stated that OntoViz has many deficiencies, and thus is not the best possible choice for ontology visualization. Problems occur especially when visualizing large, complex ontologies that would require extensive interaction operations to be fully comprehended. However, for smaller ontologies, OntoViz could be an adequate visualization tool despite its deficiencies.

3.5.4 TGVizTab

The final ontology visualization tool assessed here is TGVizTab, which is also an extension for Protégé. The motivation behind TGVizTab was to develop a lightweight ontology visualisation tool that caters for common ontology features and meets some of the special requirements for visualising such network structures. TGVizTab is based on TouchGraph, which is an open source Java environment for the creation and navigation of interactive network graphs and which applies a spring-layout technique, where nodes repel each other while edges (connections) attract. This results in the placement of semantically similar nodes closer to each other. TGVizTab is intended to be a generic, dynamic and customisable ontology visualization tool. (Alani 2003.)

TGVizTab enables multiple ways of configuring visualization. Users have full control over the colour and visibility of each relation. Users can define which relations in a graph are shown and which are hidden. It is also possible to select a certain instance or class to act as the graph's focal node. Furthermore, a double-click on a node in the visualization causes the graph to recreate itself around the clicked node. Right-clicking a node brings up a menu which contains options for hiding, expanding and collapsing individual nodes and viewing their description forms. In TGVizTab it is also possible to determine the maximum number of edges that a node is allowed to have to stay visible. With this feature users are able to control the visibility and expansion of heavily connected nodes. This approach also allows users to visualize their ontologies by interactively navigating connected sub-graphs. Finally, a zooming feature is offered, in which edges are stretched or shrunk when zooming in or out, respectively. Figure 17 shows an example of a TGVizTab view. (Alani 2003.)

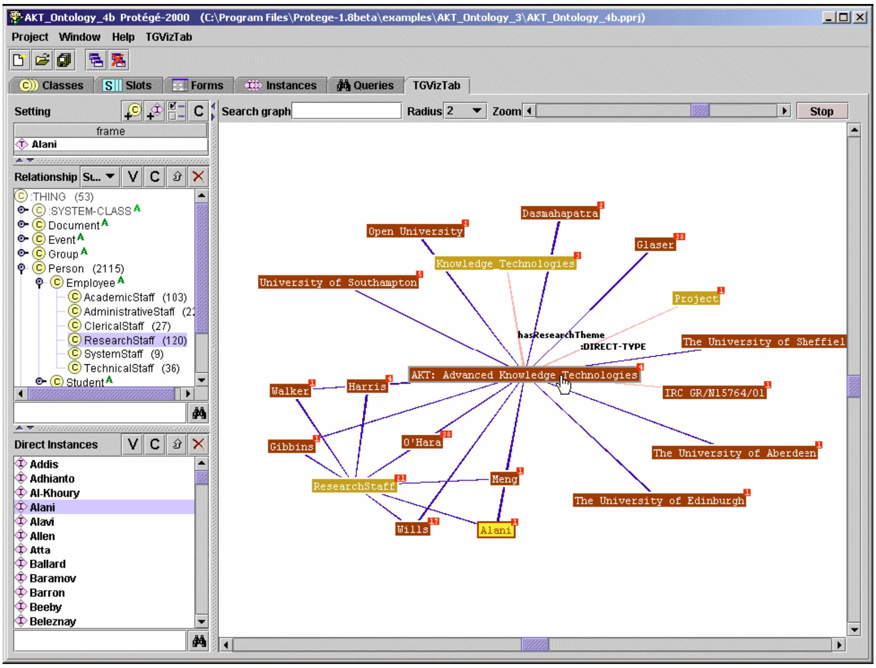


Figure 17. An example of a TGVizTab view (Alani 2003).

As seen in Figure 17, the view in TGVizTab is moderately unorganized. There is no hierarchical order between entities which makes top-down exploration difficult. Also, the relationships, especially non-structured ones, are hard to discover. However, graphs created with TGVizTab are typically readily apparent for users and excellent for incremental browsing of the ontology. The used layout enables the graph layout to be automatically rearranged in a way which makes it as readable as possible, by minimizing the crossing of edges and the overlapping of nodes. There are numerous ways to edit the visualization. For example, certain nodes and arcs can easily be turned off for an exploration of interest. (Ernst et al. 2003.) Nodes also move and adjust to user commands and the user may also expand and retract the nodes, as well as rotate the graph and change the zoom level. Thus, it can be said that the biggest advantage of this approach is its multiple interaction features. However, similarly to many other ontology visualization tools, TGVizTab lacks an effective and extensive search tool. Querying in TGVizTab is implemented by using an instance browser and a keyword search. The keyword search is available for locating classes and instances and can be applied only for information that is already visible in the respective window. (Katifori et al. 2006.)

3.5.5 Summary

As became clear, all evaluated ontology visualization approaches were graph based. Ontologies were visualized by representing their entities with nodes and the relationships between entities with arcs. Cluster Map and Jambalaya had slightly different approaches to graph visualization. In Cluster Map, graph nodes were not only added for single concepts but also for intersections between concepts. Concepts and concept intersections were represented as clusters of elements. In this way Cluster Map provided a view which shows how instances are spread through a set of classes. Jambalaya visualized ontologies in a graph in which the semantics of a relationship can be represented through graphical containment instead of edges.

In general, graphical editing operations were quite restricted. Only TGVizTab offered real possibilities to graphically edit the visualization. In other cases the interaction possibilities were restricted to navigation and the selection of features to be visualized. A zoom feature was provided in all approaches. Cluster Map and Jambalaya provided a semantic zoom, in which the amount of visible details increased or decreased as the magnification of an object changed. In the other two approaches the zoom was implemented in a more traditional manner.

Considering how important a role queries play in data analysis, the search features in most of the approaches were surprisingly modest. The only approach that contained at least somewhat effective search operations was Jambalaya. Jambalaya offered two distinguished query methods to answer the needs of different users. OntoViz did not offer any kind of search operations, which was a clear disadvantage. The most important results of the assessment of four existing ontology visualization approaches are summarized in Table 1.

Table 1. Results of assessment of four ontology visualization approaches.

	Graph-based visualization	**Query formulating**	**Graphical editing**	**Semantic zooming**
Cluster Map	yes	poor	no	yes
Jambalaya	yes	yes	no	yes
OntoViz	yes	no	no	no
TGVizTab	yes	poor	yes	no

All ontology visualization approaches evaluated here were intended to serve general purposes. They were all limited to presenting the abstract structural relations of ontology classes and their instances, and not concentrated on any specific domain. In addition, all assessed approaches were based on graph visualization algorithms. This is probably because of the wide range of areas of application that can be covered with graph-based visualizations. For example, file hierarchies, organizational charts and taxonomies that portray the relations between species can be effectively visualized with graphs. A simple way to determine the applicability of graph visualization is to consider whether there is an inherent relation in the data elements to be visualized? If the answer is yes, then the data can be represented by the nodes of a graph, with the edges representing the relations. If the answer is no, then some other visualization method might be more appropriate (Herman et al. 2000).

However, as Wehrend & Lewis (1990) have said, the chosen visualization technique should be relevant to the given problem and support the user's goal in viewing the representation. Although graph visualization is an effective technique for many problem domains, it is not an adequate technique to visualize all kinds of data. As discussed in Section 3.3, scientific and information visualization have been strongly influenced by new visualization techniques, inherited mostly from the computer game industry. For example, approaches that aim to visualize different kinds of physical environments and spatial relations between entities have successfully adopted visualization techniques used mostly in computer games. Thus, when designing new visualization applications, a wide variety of different visualization technologies should be taken into consideration when selecting the best possible visualization technique for the given problem domain.

4. The construction process of an ontology visualization tool

The main idea of constructive research is to create something new on the basis of existing research knowledge (Järvinen & Järvinen 2000, p. 102). As mentioned before, the target of this study was to produce an interactive ontology visualization approach. With this approach it should be possible to visualize, edit and even create ontologies, particularly in the domain of networked home environments. The construction work was started by getting familiar with the existing theory base. This literary review was carried out in order to gain insight into the given domain and to get some perspective on how the visualization should be implemented and which interaction possibilities should be included. The assessment of four existing ontology visualization approaches aimed at producing ideas for the design and implementation processes. During the early stages of the construction process, it was also decided to name the approach "VantagePoint".

The implementation phase was divided into two sub-processes: the specification process and the implementation process. As mentioned in Chapter 1, the construction process was incremental in nature and therefore both sub-processes were carried out in parallel. However, the tasks related to the specification process always preceded the tasks related to the implementation process. In this way it could be assured that every action that took place was planned beforehand. The parallel completion of the sub-processes was chosen because of the unique nature of the constructed approach. As Järvinen & Järvinen (2000, p. 114–115) have mentioned, the parallel completion of the specification process and the implementation process is often used in such situations where something completely new is being created. In these kinds of situations people in general find it difficult to imagine something that has never existed. Thus, it was considered beneficial to receive feedback after each implemented prototype version and to update specifications accordingly. Every prototype also provided an opportunity to compare the current state to the target state, and to decide on the most necessary actions to be taken next.

In this chapter the specification and the implementation processes are described in more detail. The specification process is divided into three sub-tasks: defining the requirements, defining the components, and defining the architecture of the approach. The implementation process is examined by describing the software tools and technologies used in the implementation work, and comparing the first implemented prototype to the latest one. The comparison is carried out to illustrate the evolution of the application during the construction process. Finally, the most important interaction operations of the latest prototype version are explained in a detailed fashion.

4.1 Specification process

Järvinen & Järvinen (2000, p. 106) have mentioned that the goal of the specification process in a construction study is to produce a description of the target state (see Section 1.2.2). In this study the specification process is divided into three distinct phases, and a detailed description of each phase is presented in this section. First, the most important requirements of the constructed approach are defined and why these requirements must be fulfilled is also considered. Second, the different components of the approach are explained and third, the architecture of the constructed approach is explained. These descriptions also include the diagrams and tables resulting from each phase

4.1.1 Requirements

There were some general requirements for VantagePoint that were clear from the beginning of the construction work. For example, it was obvious from the start that the application should be interactive. Users should be offered multiple methods of communicating with the visualized data. Another requirement was that VantagePoint should make ontologies more concrete, interesting and, above all, easier to comprehend. VantagePoint should also take into consideration the characteristics of ontologies that describe networked home environments (the essential role of contextual information, a relatively small number of classes and thus a simple class hierarchy, etc.). The final initial requirement was that with VantagePoint it should be possible not only to visualize ontologies, but also to create semantic models of networked home environments through graphical editing operations.

VantagePoint was aimed to help at least two different kinds of user groups. First, it was designed to help people who are not familiar with ontologies and the OWL language. VantagePoint should shorten the gap from beginner to intermediate OWL ontology reader by visualizing ontologies in a realistic manner. Second, VantagePoint was supposed to work as a virtual test laboratory for application developers that need ontologies and contextual data in their work by allowing the developers to see the operations as in real life and to better notice practical errors without expensive test laboratories.

No official requirement specification documents were written for VantagePoint, since the construction process was explorative by nature and it was clear that the requirements would evolve as the construction process progressed. However, some more specific requirements were defined and they are presented in Table 2.

Table 2. The most important requirements of VantagePoint.

Requirement	Explanation
R01	VantagePoint is able to present contextual data and spatial relationships effectively.
R02	VantagePoint is able to realistically present the networked home environment described by an ontology.
R03	Relevant entities in the visualization can be represented with 3D icons and with 2D symbols.
R04	VantagePoint offers multiple operations for editing the visualization. Users should be able to at least add, delete, rotate and move entities. Users are able not only to edit the visualization, but to directly manipulate the ontological data through visualization.
R05	With VantagePoint it is possible to graphically create semantic models of environments by exploiting the operations defined in R04. VantagePoint generates the OWL files that are necessary for storing the models.
R06	VantagePoint offers the possibility to view the visualization from multiple angles and with different perspectives. In this way users should be able to get a good overall view of the environment and also perform accurate editing operations.
R07	VantagePoint offers a possibility to get additional information about visualized entities.
R08	VantagePoint offers extensive and effective query possibilities. Users should be able to execute queries both graphically and textually.
R09	VantagePoint enables the printing of the visualized model in a textual form.
R10	VantagePoint is easy to use. Usability issues are taken into consideration in the design process of the user interface.

As discussed in Section 2.6, ontologies of the networked home environment domain describe the contextual environment among other things. Thus, it was defined in R01 that it would be important to present contextual information and spatial relationships effectively in the visualization. Requirement R02 came from the supervisors of the construction work, who thought it was important that the graphical appearance of the visualization was realistic and that it would impress users and encourage people to start using it. In addition, as discussed in Section 3.2, realism breeds the expectation of accuracy, reliability and authority in the representation, especially when considering computer visualizations which aspire to simulate real environments. This argument goes also for requirement R03, since realistic icons enhance the authentic feeling and make the visualization more decorative. According to Marr & Nishihara (1978), realistic three-dimensional icons enhance identification because the iconic mapping from objects to their physical shapes is direct and more intuitive. However, Smallman et al. (2001) have disagreed with Marr & Nishihara and stated that 2D symbols can be identified more accurately and faster than 3D icons, and that 3D icons may in fact cause confusion among users. Thus, it was decided that in VantagePoint both 3D icons and 2D symbols are supported.

As became clear in Section 3.5.5, the lack of interactive elements was the biggest disadvantage in the assessed ontology visualization tools. Thus, interactivity was considered as an important design principle for VantagePoint. Requirements R04–R07 were produced to ensure that the interactivity is realized in a best possible manner in VantagePoint. As mentioned in Section 3.4, interactive elements help us to understand the visualization better. With interactive operations it is possible to present "what-if" questions and to test what the consequences of different scenarios are. Thus, VantagePoint was equipped with various editing operations such as adding, moving and deleting of instances. These operations were defined not just to enable the editing of the visualization itself, but also to give direct access to the underlying ontological data behind the visualization. This means that whenever editing operations are made in the visualization, the ontological data changes accordingly. Requirement R05 declared that VantagePoint should enable users to create their own semantic models by using the graphical editing operations defined in R04. By fulfilling requirement R05, VantagePoint is extended from an ontology visualization tool to an ontology creation tool, which opens whole new opportunities for end users.

In Section 3.2, it was mentioned that the ability to view the data from different perspectives and from different angles enhances understanding of the data. To realize this, a requirement for multiple angles and perspectives was added to VantagePoint (R07). By defining this requirement, it was assured that users would get a good overall picture of the data, as well as being able to perform accurate editing operations.

The next requirement (R08) was to ensure that VantagePoint offers extensive possibilities to perform queries. As was concluded in Section 3.5.2, it is often a good idea to offer multiple ways to perform queries. Thus, it was decided that VantagePoint should offer two distinct options for query construction – a graphical query and a textual query. With the graphical query option inexperienced users should also be able to execute queries without having any particular knowledge of the ontology query languages RDQL/SPARQL. With the textual query option, advanced users should be able to define their own query statements without any constraints.

Also, requirements R09 and R10 are closely attached to interaction issues. According to requirement R09, users should be able to print the visualized ontology in a textual form. The possibility to print ontologies was added to help users to gain an insight into how the effects made in the visualization change the ontological data, which was intended to support the pedagogical goals of VantagePoint. Requirement R10 was defined in order to ensure that usability issues are taken into consideration during the construction process. According to Nielsen (1993, p. 26), usability is associated with five attributes:

- Learnability: The system should be easy to learn so that users can rapidly start getting some work done with the system.
- Efficiency: The system should be efficient to use in order to ensure a high level of productivity.
- Memorability: The system should be easy to remember so the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again.
- Errors: The system should have a low error rate, so that users make few errors when using the system, and so that if they do make errors they can easily recover from them.

– Satisfaction: The system should be pleasant to use.

These attributes were adopted as a guideline when designing the user interface and the user interaction operations of VantagePoint.

4.1.2 The logical architecture of VantagePoint

VantagePoint is comprised of multiple components that are presented in Figure 18. In this chapter these components are explained.

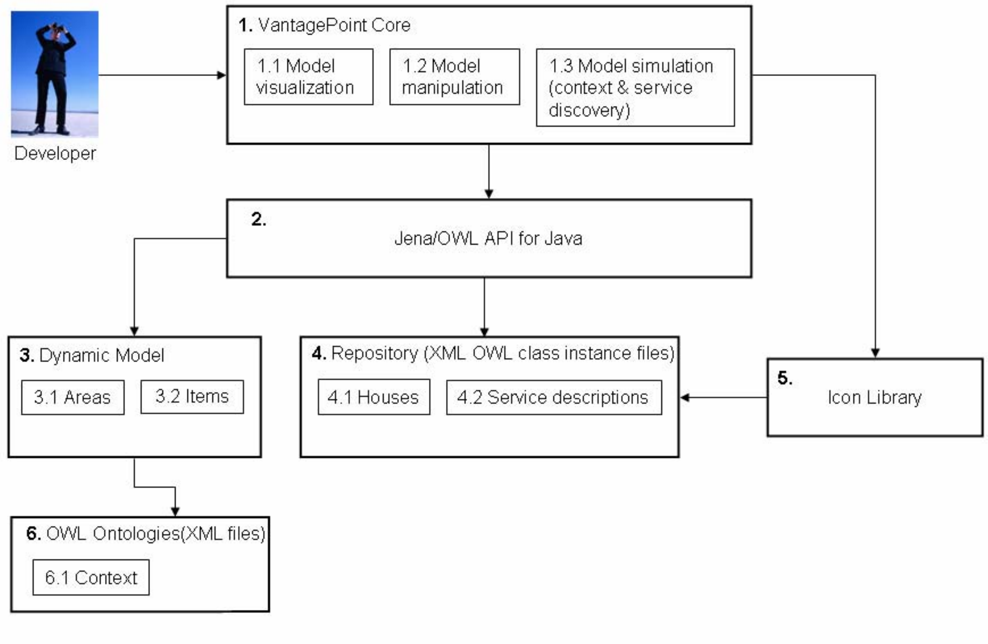


Figure 18. The components of VantagePoint.

VantagePoint Core contains the core functions of the software. As discussed above, VantagePoint forms semantic models of networked home environments and the *Model visualization* component is responsible for the visualizing of these models. It reads OWL files and searches for individuals that belong to predefined *VisualComponent* classes. Only individuals that belong to the *VisualComponent* class are visualized, all other data that is irrelevant or impossible to visualize are ignored. The *VisualComponent* class is divided into two separate subclasses - *Item* and *Area*. All things visualized fall into these two classes. If the ontology does not contain the class *VisualComponent*, it cannot be visualized with VantagePoint. In such cases a blank visualization is created and the user can print the loaded model as a text file, make queries to it or add new instances. If a visualized ontology contains instances that have defectively defined properties and cannot thus be visualized, these instances are gathered into a defective components list.

The *Model manipulation* component enables changes to be made to the visual representation of the model. The manipulation consists of editing operations such as removing, adding, moving and rotating of instances. The moving and rotating operations edit individuals' properties. For example, location properties are changed by moving instances in the visualization and the 'contains' property is automatically calculated from the new locations of the moved instances. When the user removes an area from the visualization, VantagePoint automatically removes all the instances that were contained by this area. This is implemented by utilizing the reasoners offered by Jena (see Section 2.4). The last manipulation operation is the adding operation. All types of visual components can be added to a model with VantagePoint. All the editing, removing and adding operations always change both the visualization and the dynamic model.

The *Model simulation* component defines ways of implementing context and service discovery operations. In practise, these operations are carried out by utilizing the SPARQL and RDQL query interfaces provided by VantagePoint.

As mentioned in Section 2.4, Jena is a Java framework that provides a programmatic environment for RDF, OWL, RDQL and SPARQL. The **Jena/OWL API for Java** component is used in VantagePoint to manage and query OWL ontologies. Jena provides a predefined OntModel class, which in VantagePoint is used for encapsulating OWL ontologies. The class OntModel supports for the kinds of objects expected to be in an ontology are: classes, properties, and individuals (Carroll et al. 2004).

VantagePoint offers tools to retrieve models from a disk, manage them dynamically and store them back on the disk. The Dynamic Model component enables the dynamic management of models. When a model is being graphically manipulated, the changes are simultaneously added both to the visualization of the model and the model itself. However, these changes are not permanent until the model is saved.

The **Dynamic Model** holds two sub-components; *Areas* and *Items*. All visible entities in a VantagePoint visualization belong either to class 'Area' or the class 'Item'. Areas are used to represent different kinds of spaces, like rooms and hallways in the house model. In the visualization, areas are polygons defined by a set of points. Items represent either devices or persons and they are represented with realistic 3D icons or 2D symbols that are supposed to make items recognizable.

In VantagePoint, an instance of the class 'Item' contains a 'represents' property, which is a URI that points to the data that describes what this particular item represents. The data attached to items are descriptions of devices and their services or descriptions of persons and their current state of mind. Items can also appear without any attached data. These kinds of items are just decorations to make the house environment look more real and to provide a more authentic feeling.

The **Repository** component contains models of houses and services that are stored in OWL files. The house models are actually OWL ontologies containing instances of the classes 'Area' and 'Item'. The ontology that holds the model of the house may have data that does not concern VantagePoint in the matter of visualizing. This kind of data is simply ignored when visualizing but it is available for querying.

The *Service descriptions* sub-component contains the service descriptions for different home appliances that are described using the OWL-S service description language (see Section 2.3) and can be found in networked home environments. This component is also an extension point of VantagePoint, as device manufactures are able to add new service descriptions to the service description repository.

The **Icon library** component defines a basic library of icons. The icon library contains both symbolic 2D icons and realistic 3D icons for devices and persons.

The icon library is stored in a text file, which contains the URLs of the icon files (PNG images) and description of the icons.

The **OWL ontologies** component contains an ontology called WorldModel.owl, which defines the context of the VantagePoint world and holds the class and property definitions of that context. Figure 19 presents the context model used by VantagePoint.

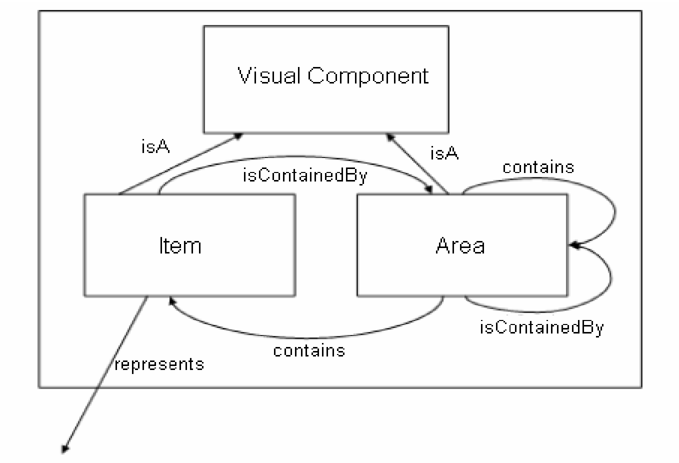


Figure 19. WorlModel.owl – ontology of the VantagePoint world.

As can be seen in the chart, WorldModel.owl contains one super-class, 'VisualComponent' and two sub-classes: 'Area' and 'Item'. As discussed before, the instances of the class 'Item' have a property called 'represents', which means that the instances can represent some external data, which is usually an ontology. When the 'represents' attribute is empty, an item is thought of as a "dummy item" that has no service descriptions or other kind of functionality. The instances of the class 'Area' do not have the 'represents' relationships. Areas can contain items and an item has to be contained by some area. An area can also contain other areas and it can be contained in another area. However an area does not necessarily have to be contained in another area.

4.1.3 The architectural structure of VantagePoint

The Model-View-Controller (MVC) user interface framework was selected as a starting point for the architectural design of VantagePoint. The Model-View-Controller architecture is a widely-used architectural approach for interactive applications. It divides functionality between objects involved in maintaining and presenting data to minimize the degree of coupling between the objects (Krasner & Pope 1988). In the Model-View-Controller architecture, objects of different classes take over the operations related to the application domain (the model), the display of the application's state (the view), and the user interaction with the model and the view (the controller) (Singh et al. 2002, p. 384). The basic idea is that the model notifies views when it changes and provides the ability for the view to query the model about its state, the view renders the contents of a model, and the controller defines application behaviour (Krasner & Pope 1988). The structure of the Model-View-Controller architecture is presented in Figure 20.

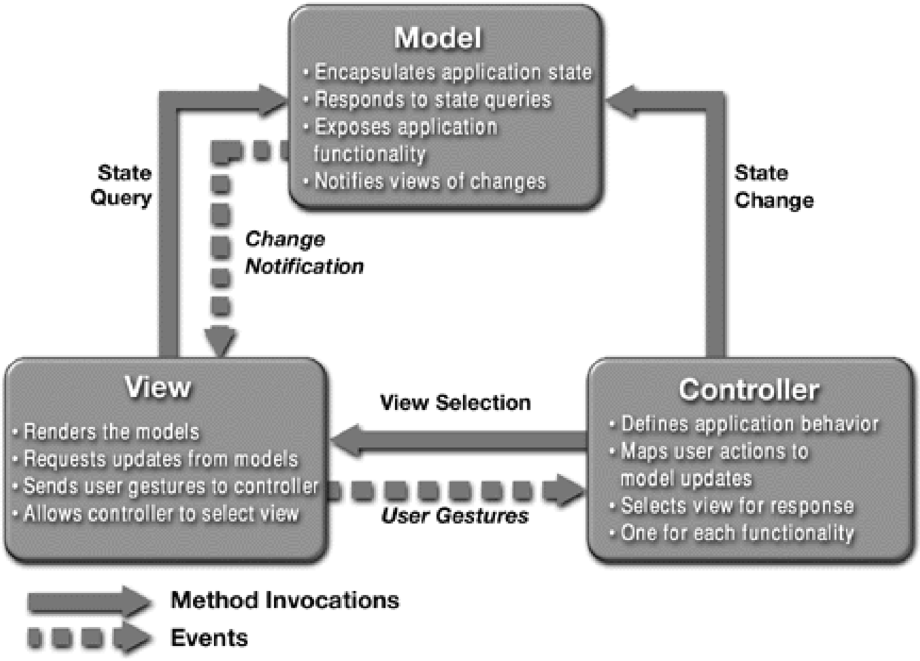


Figure 20. The Model-View-Controller architecture (Singh et al. 2002).

The basic interaction cycle in the Model-View-Controller architecture starts when the user performs an input action and the view sends these user gestures to the controller. The controller's responsibility is to dispatch user requests and to select views for presentation. The controller also interprets user inputs and maps them into actions to be performed by the model. The model notifies views when it changes and provides the ability for the view to query the model about its state. It also provides the ability for the controller to access application functionality encapsulated by the model. A view renders the content of a model. It accesses data from the model and specifies how that data should be presented. The view also updates data presentation when the model changes. Finally, a new cycle is started when the view forwards new user inputs to the controller. (Singh et al. 2002 p. 384; Krasner & Pope 1988.)

The Model-View-Controller architecture has some indisputable advantages. Krasner & Pope (1988) have stated that, especially when building interactive applications, modularity of components has enormous benefits. Isolating functional units from each other as much as possible makes it easier for the application designer to understand and modify each particular unit, without having to know everything about the other units. This three-way division of an application entails separating the parts that represent the model of the underlying application domain from the way the model is presented to the user and from the way the user interacts with it. Singh et al. (2002, p. 384) have mentioned that separating responsibilities among model, view, and controller objects reduces code duplication and makes applications easier to maintain. It also makes handling data easier, whether adding new data sources or changing data presentation, because business logic is kept separate from data.

The Model-View-Controller framework is also the basis of the architectural design of VantagePoint. A more accurate structure of VantagePoint is presented in the form of a class diagram in Figure 21.

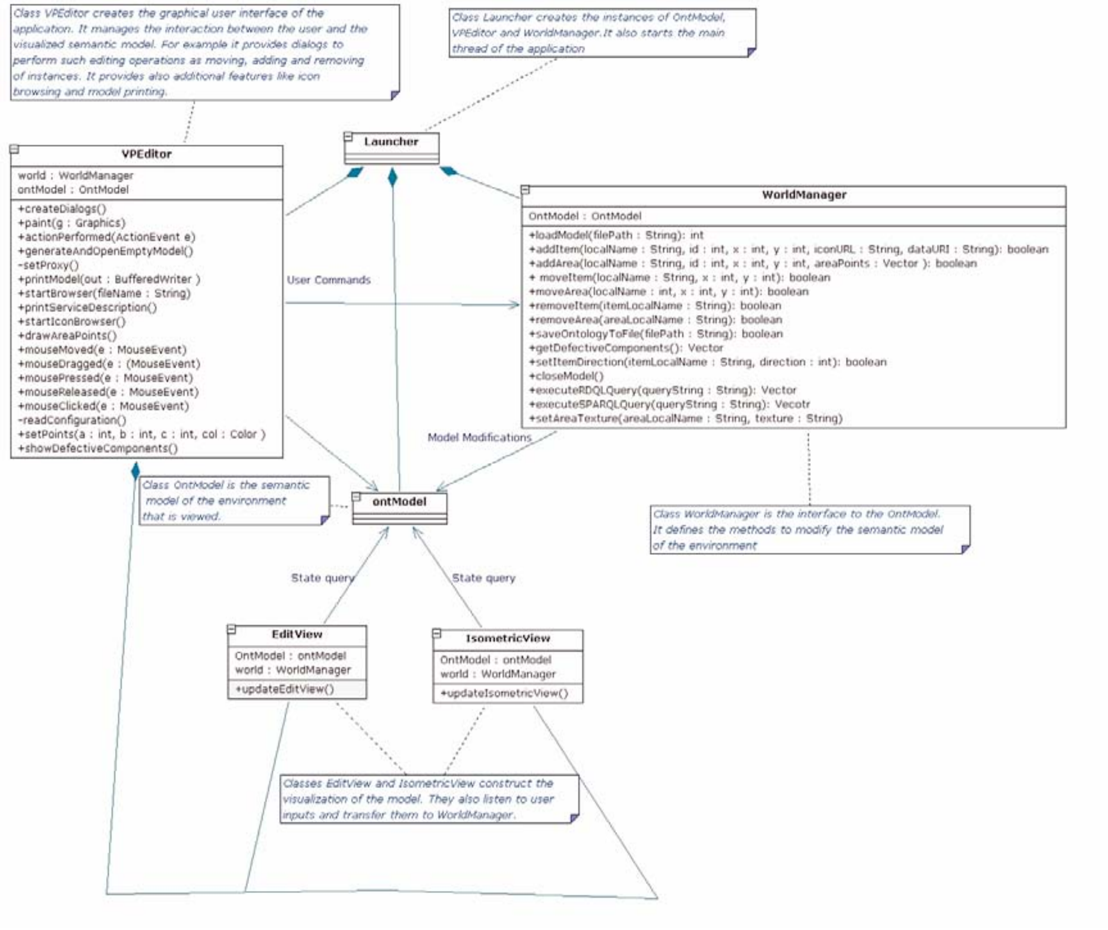


Figure 21. A diagram of the core classes, attributes and methods of VantagePoint.

The class diagram of VantagePoint contains six classes. Each class is explained in more detail below.

OntModel: This class corresponds to the "model" class in the Model-View-Control architecture. The OntModel class in VantagePoint stores the semantic model of the environment and it is adopted from the class library provided by Jena (see Section 2.4). OntModel is an ontology interface that provides a convenient API for working with ontology models. It supports the kinds of objects expected to be in an ontology: classes, properties, and individuals (Carroll et al. 2004). The OntModel class in VantagePoint works slightly differently than the model class in the MVC architecture presented in Figure 20. In VantagePoint the model class does not notify the views when it changes. Instead, the class VPEditor updates the views as it deems it necessary. In this way, the change notification routines do not have to be implemented into the OntModel class.

WorldManager: This class corresponds to the “controller” class in the Model-View-Control architecture. It defines methods of controlling and manipulating the model. For example, it offers operations to add or remove instances from models, save and open models, and query models. Similarly to MVC’s controller class, it maps the user actions to model updates.

EditView/IsometricView: These classes correspond to the “view” class in the Model-View-Control architecture. The view classes build up the visualization straight from the model when necessary. These classes also implement the methods of ‘observing’ to users’ gestures when they perform editing operations

VPEditor: This class initializes the graphical user interface and acts as a listener for the views. The VPEditor class does not fit straightforwardly into the MVC architecture, but it can be considered as a hybrid of the view and controller classes, because it on one hand uses the class WorldManager as an interface to the OntModel, but on the other hand is responsible for selecting and updating the views, and presenting information about the model to the users.

VPLauncher: This class launches VantagePoint and creates the instances of the components contained in VantagePoint. The main idea behind this class is to gather all the components together and clarify the initialization of the application

To illustrate the basic functionality of the approach, a detailed description of VantagePoint’s interaction cycle is presented in Figure 22.

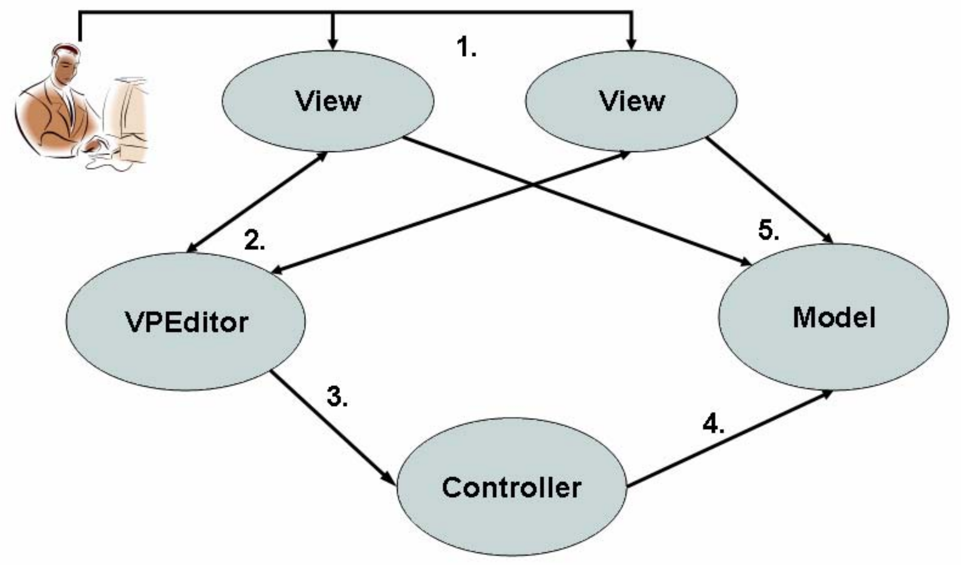


Figure 22. A standard interaction cycle of VantagePoint.

1. The basic interaction cycle starts when the user performs an editing operation in some of the views. For example, the user may change the location of an item by dragging it to another position.
2. The VPEditor class 'listens' to the views and tracks the changes occurring in them.
3. VPEditor forwards the necessary information about the editing operation to the controller class.
4. The controller class (WorldManager) acts as an interface to the model (OntModel) and changes the model according to the editing operation.
5. Finally, VPEditor calls the views' update methods to set the views in sync with the current state of the model.

As discussed above, the user interface framework of VantagePoint differs slightly from the MVC architecture shown in Figure 20. The biggest difference is the role of the model class. In VantagePoint, the OntModel class is adopted from the class library provided by Jena and it corresponds to the model class of the MVC architecture. During the construction process it was considered better not to implement any new functionality in the OntModel class, but to leave it as it was. Instead, a new class called VPEditor was created. The VPEditor class was added to implement some of the functionalities that would normally belong to the model class in the MVC architecture.

Furthermore, it was decided to implement the user interface elements of the application in the VPEditor class. In this way the WorldManager class, which acts as a controller class in VantagePoint architecture, could be maintained as a simple interface to the model. This was considered to increase the versatility and the reusability of the approach.

4.2 The Implementation process

The implementation process was carried out on the basis of the specification process. In the implementation process the following questions were presented: “Is the approach feasible?”; “Is the approach feasible with the given resources?”; and, “How can the approach be implemented?” The goal of the implementation process was to realize the target state by producing a software tool which fulfils the requirements set in the specification process

The implementation process was started by getting acquainted with the Jena framework. As discussed in Section 2.4, Jena is a powerful tool to manage ontologies and it supports the ontology query languages RDQL and SPARQL. Another well-studied technique was the Swing library, which is a graphical user interface toolkit for Java. Swing was utilized to create the graphical elements of the approach, such as dialogs, panels and buttons. The actual programming work was carried out using Eclipse, which is an open-source-based development platform (http://www.eclipse.org/).

The implementation work was started by creating the first prototype of the approach. The goal of the first prototype was to illustrate the graphical appearance, the user interface and the interaction methods of the approach. A screenshot from the first prototype is presented in Figure 23.

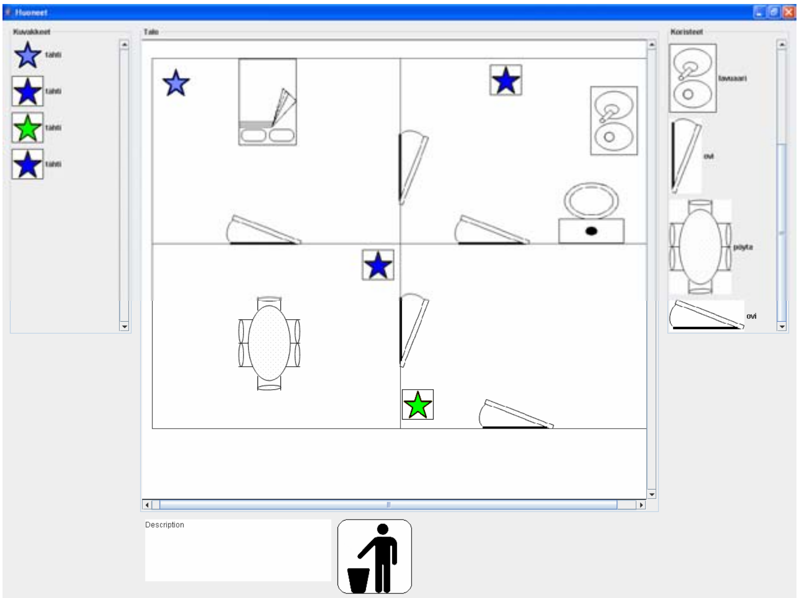
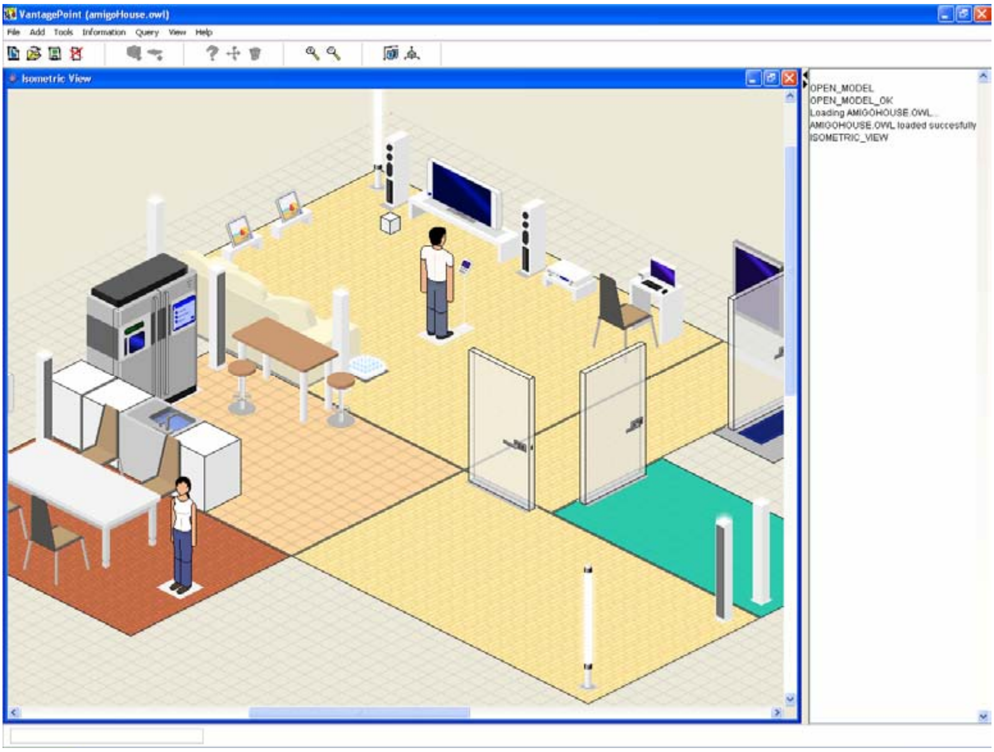


Figure 23. A screenshot from the first prototype of VantagePoint.

At this point the visualization could only present environments from a bird's eye perspective. No ontologies were involved; instead a sample house environment with four rooms was hard-coded in and visualized every time the application was launched. Also, the interaction possibilities were quite restricted as users were only able to add or remove items through drag-and-drop operations. None of the items had ontologies attached; they were just icons on the screen. As can be seen, the graphical appearance and the user interface were still quite primitive. The construction work of the first prototype was aimed at producing new ideas for further development work and to practise skills related to graphical programming and the efficient exploitation of Jena.

As discussed at the beginning of this chapter, the implementation work took place in parallel with the specification work. After every implemented prototype, the specification process was returned and the specifications were updated according to the feedback received. Gradually the application evolved as the ontologies became involved and as the graphical appearance and the user interaction procedures became more sophisticated. In the most recent prototype versions, the application was able to three-dimensionally visualize networked home environment models defined by ontologies. The interaction possibilities were also much more advanced than in the very first prototypes. Eventually, it was possible to build semantic models of networked home environments by defining areas, devices and persons with the graphical interaction operations offered by VantagePoint. A screenshot from the latest prototype version of the approach is presented in Figure 24.



```
OPEN_MODEL
OPEN_MODEL_OK
Loading AMIGOHOUSE.OWL...
AMIGOHOUSE.OWL loaded succesfully
ISOMETRIC_VIEW
```

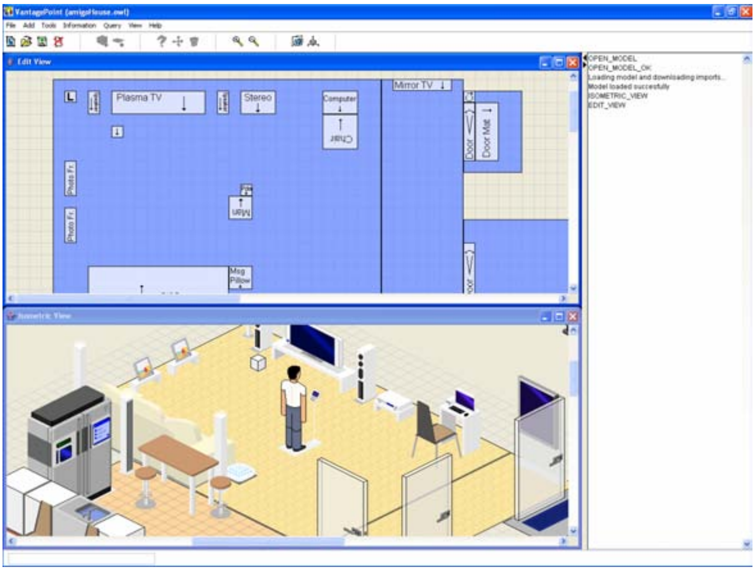
Figure 24. A screenshot from the latest prototype version of VantagePoint.

The resources available for this study did not enable any kind of testing work to be performed. For instance, it would have been interesting to test how effectively VantagePoint is able to visualize extremely large ontologies. Currently, the biggest ontologies visualized with VantagePoint have contained approximately 10 000 RDF- triplets. In addition, various usability tests would have given valuable information about the learnability and functionality of the graphical user interface of VantagePoint.

4.2.1 The editing operations provided by VantagePoint

As discussed in Section 4.1.1, one of the most important requirements of VantagePoint was to offer extensive interaction possibilities to the user. As became clear in the assessment of existing ontology visualization tools, interaction possibilities were in general underexploited, which was considered to be a clear disadvantage (see Section 3.5). Therefore, special attention was given in the construction work of VantagePoint to the interaction between the user and the visualization.

In the Section 4.1.1, it was concluded that VantagePoint should offer **multiple views** and thus enable users to see the visualization from multiple angles and with different perspectives. Therefore, two distinct views were implemented into VantagePoint. The edit view is a 2D 'ground plan' view of the ontology that has been visualized. The purpose of the edit view is to enable more accurate editing operations. As became clear in Section 3.2, 2D views are considered better for navigating and measuring distances precisely, establishing precise relationships and performing spatial positioning. The isometric view was implemented to visualize ontologies in a more impressive way. In this view the environment is presented from an isometric projection which should offer a better general view of the house. As mentioned in Section 3.2, three-dimensional displays are said to be good for gaining an overview of a 3D space, understanding 3D shape, and navigating approximately. However, the isometric view does not offer as accurate editing and adding operations as the edit view. Figure 25 presents a screenshot from VantagePoint where both views are simultaneously visible.



```
OPEN_MODEL
OPEN_MODEL_OK
Loading model and downloading imports...
Model loaded successfully
ISOMETRIC_VIEW
EDIT_VIEW
```

Figure 25. A VantagePoint screenshot where both views visible.

The exploitation of both visualization types, 2D and 3D, was considered to be a necessity, since VantagePoint was required not only to visualize ontologies, but also to offer means to create and edit them. Both views have their own special purposes and both views are needed: the edit view for editing and the isometric view for gaining an overview. As was mentioned in Section 3.2, having both visualization types in the same approach is desirable, as long as it is appropriate. As can be seen from Figure 25, the appearance of the edit view is somewhat rough. Items are represented with symbols, which include a textual description of the item, and an arrow indicating the current direction of the item. The edit view is also presented from a bird's eye perspective, which does not exploit the three-dimensional visualisation. Instead, it enables the possibility to accurately create areas with exact measurements and locate items in their correct positions.

In Section 3.3 the fact that the rapid development of the computer game industry has had a huge influence on scientific and information visualization was discussed. Many approaches implemented for educational or scientific purposes have been inspired by computer games. In many cases this has also proven to be an advantage, as these approaches have been able to visualize data in a more illustrative and impressive manner. In VantagePoint, the isometric view has been strongly influenced by the graphical appearance of the game 'The Sims'. This can be easily seen by examining Figure 26, in which a screenshot from both approaches is presented. As mentioned in Section 3.3, the isometric projection allows users to have a general view of the visualized world at a glance. It was also stated that by using the isometric projection, spatial relationships between objects can be seen within wide environments. Therefore, it was also considered to be a good idea to exploit the graphical appearance of The Sims in VantagePoint.

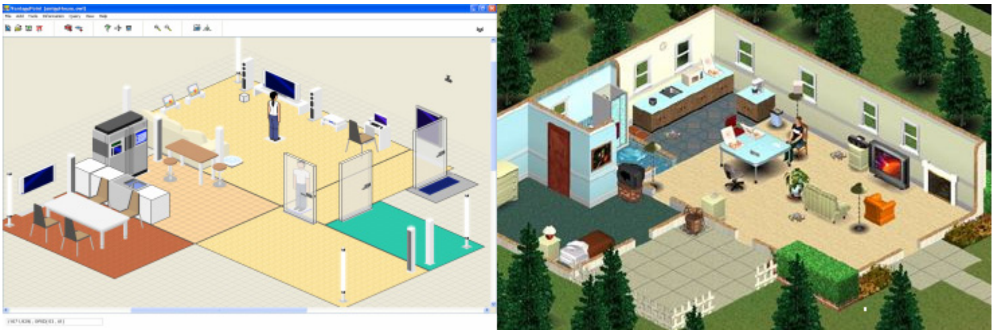


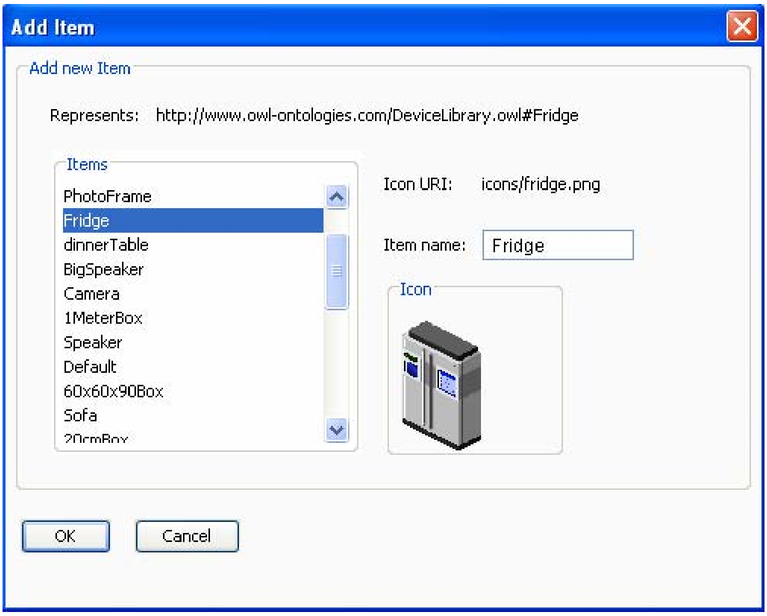
Figure 26. The isometric view of VantagePoint vs. a screenshot from the game The Sims [1].

VantagePoint offers operations for **adding** and **removing** of instances. As mentioned in Section 4.1.1, users should be able to directly manipulate the ontological data through the interaction operations defined for VantagePoint. Thus, when the user adds an instance to the visualization it is also added to the semantic model. Similarly, as instances are deleted from the visualization, they are also removed from the model.

VantagePoint provides the possibility to graphically add items and areas to the model. The dialog box for adding items is presented in Figure 27.

---

[1] The screenshot from http://www.application-systems.de/sims/screenshots.html



Represents: http://www.owl-ontologies.com/DeviceLibrary.owl#Fridge

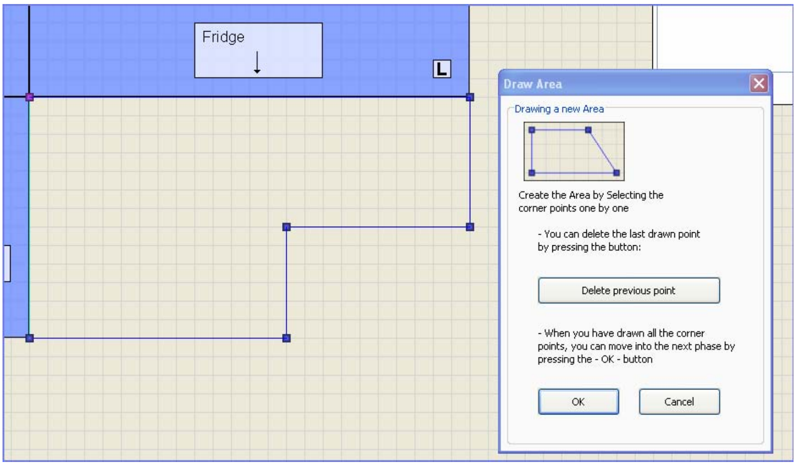
Icon URI: icons/fridge.png

Item name: Fridge

Figure 27. The "add item" operation.

The adding operation is started by selecting an item from the text list which is located in the left margin of the dialog box. The icon that represents the selected item can be seen in the lower right corner of the dialog box. The first text field describes which ontology file (service description) the item is attached to. The second field shows the URI of the icon that is used to represent the item in the visualization. The name of the item can be written into the third field. Once the item has been named and the "OK" button has been pressed, the location for the item can be determined by dragging it to a desired position.

As mentioned above, users are also able to add areas into the model. This is carried out by using the operation presented in Figure 28.



Create the Area by Selecting the corner points one by one

- You can delete the last drawn point by pressing the button:

- When you have drawn all the corner points, you can move into the next phase by pressing the - OK - button

Figure 28. The "add area" operation.

The adding of areas is performed by assigning the corner points of the area to the design space (see Figure 28). Once the desired points have been selected (three at least), the area must be named. It is also possible to determine a floor material for the area. Different floor materials are represented with different textures in the isometric view (see Figure 24).

Delete operations are performed simply by selecting the desired instance and pressing the 'garbage can' button in the control panel. As the selected instance disappears from the screen, it is also removed from the ontological model. It is worth remembering that if an area is deleted, all areas and instances that were contained by this area are also deleted.

In addition, to the operations described above, VantagePoint offers interaction operations such as **moving of instances, printing of models** and **getting additional information about instances**. In VantagePoint all visualized elements are movable and the moving operation is executed simply by dragging an instance to a new location. The printing operation enables the visualized model to be printed in a textual form. In this way it is possible to examine the structure of the OWL file and see how the changes made in the visualization have affected the model. In addition, the service descriptions attached to different items can be printed. The final interaction operation described here is called "getting additional information about instances". This operation enables certain additional information about selected instances to be quickly obtained without constructing any queries. An example of this operation is shown in Figure 29. As can be seen, the additional information contains such data elements as the name, the ID, the size, the position, the URI of the icon and the containment relationship of the selected instance.

Name: fridge
ID: 3
Width: 7
Height: 3
Relative Grid X: 0
Relative Grid Y: 4
Icon URL: icons/fridge.png
Is contained by: http://www.owl-ontologies.com/MyHouse.owl#kitchen

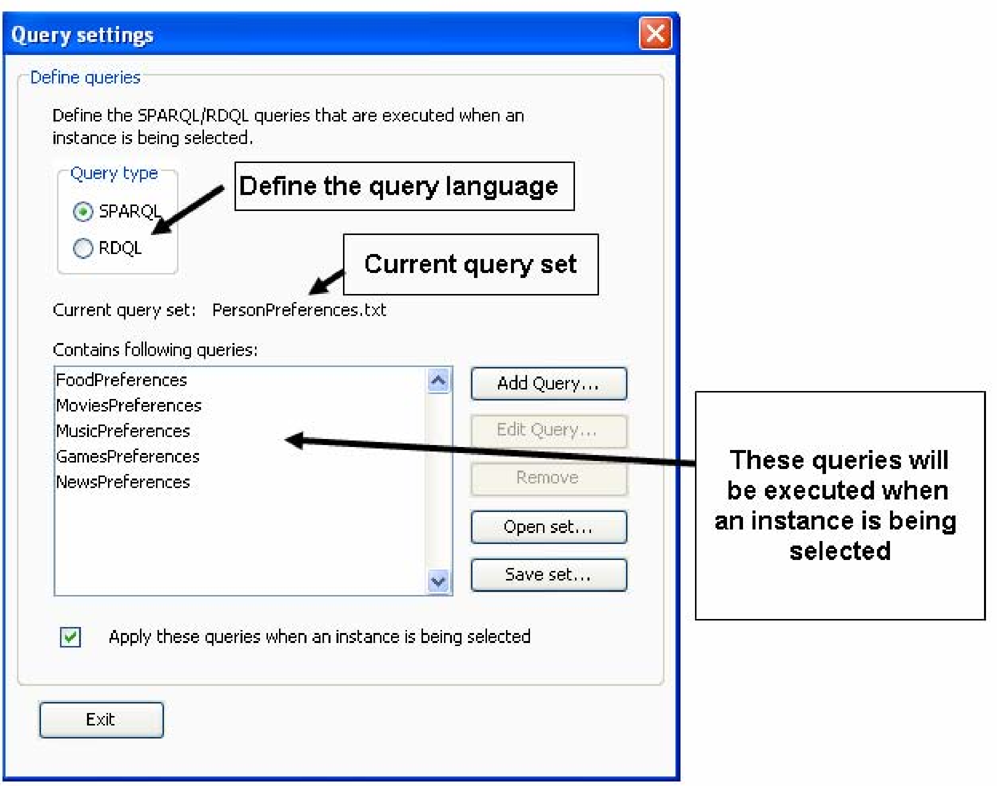
Figure 29. The "Instance information" dialog.

4.2.2 Query construction in VantagePoint

One of the most important features of VantagePoint is its extensive support for **query construction**. VantagePoint supports two ontology query languages: RDQL and SPARQL (see Section 2.5). Queries can be used for multiple purposes. As mentioned in Section 2.6, with ontologies it is possible to describe all entities related to networked home environments such as applications, users, devices and services, to name just a few. However, all of this information cannot be included in the visualization and therefore queries are needed to provide access to this "hidden data". An example of data that cannot be visualized is the service descriptions attached to different items. These service descriptions are described with the OWL-S language (see Section 2.3), and they define, for example, what services a certain item is offering, what the inputs, outputs, preconditions and effects of these services are and how these services can be exploited. To be able to access this information, queries are needed.

In Section 3.5.5, it was discussed that in the existing visualization approaches the query construction features were either lacking completely or too restrictive. The only exception was Jambalaya, which offered two distinguished query methods to answer the needs of different users. In the construction work of VantagePoint the extensive support for query construction was considered to be a very important feature. Queries are an effective way of retrieving data from ontologies, and with queries it is possible to access even data that could not be visualized. It was decided to equip VantagePoint with two different query methods: a graphical query and a free query.

Graphical querying in VantagePoint means that users can define queries that will be executed when an instance is being clicked on in the visualization. By means of graphical querying it possible to retrieve information about, for example, what services are offered in a certain area. The queries that will be executed when an instance is being clicked can be defined in the query settings dialog box, which is presented in Figure 30.



Define the SPARQL/RDQL queries that are executed when an instance is being selected.

Query type

● SPARQL

○ RDQL

Define the query language

Current query set

Current query set: PersonPreferences.txt

Contains following queries:

- FoodPreferences
- MoviesPreferences
- MusicPreferences
- GamesPreferences
- NewsPreferences

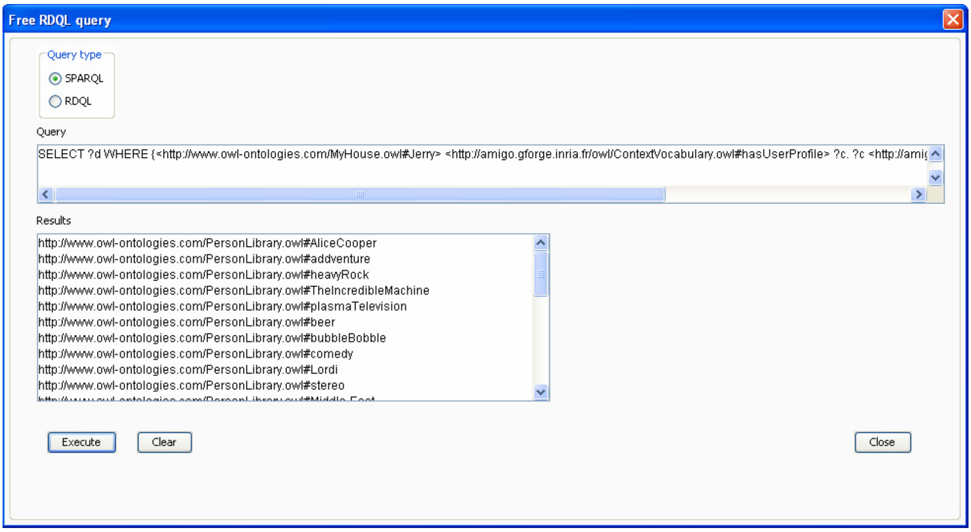
These queries will be executed when an instance is being selected

☑ Apply these queries when an instance is being selected

Figure 30. A dialog box to define queries executed in graphical querying.

As can be seen from Figure 30, it is first possible to select the query language that will be used in the queries (RDQL/SPARQL). Next, the user is informed about the query set that is currently open. The query set includes a collection of queries that will be executed when an instance is being clicked. These queries are listed in the text area below. Individual queries can be edited, removed or added. To be able to define own queries or query sets, the knowledge of RDQL or SPARQL is required. However, the query sets can be saved in text file to be reused later and thus the same queries can be executed through graphical user interface with minimal knowledge about the query languages needed. The results of the executed queries will appear to the information area, which is located in the right margin of the user interface.

In the free query, the queries to be executed are not restricted in any way. With the free query it is possible to retrieve any kind of information from the model, even data that could not have been visualized. As presented in Figure 31, the free query is constructed by writing the query statement in to the upper text area and pressing the execute button. Results will appear to the lower text area.



```
SELECT ?d WHERE {<http://www.owl-ontologies.com/MyHouse.owl#Jerry> <http://amigo.gforge.inria.fr/owl/ContextVocabulary.owl#hasUserProfile> ?c. ?c <http://ami
```

```
http://www.owl-ontologies.com/PersonLibrary.owl#AliceCooper
http://www.owl-ontologies.com/PersonLibrary.owl#addventure
http://www.owl-ontologies.com/PersonLibrary.owl#heavyRock
http://www.owl-ontologies.com/PersonLibrary.owl#TheIncredibleMachine
http://www.owl-ontologies.com/PersonLibrary.owl#plasmaTelevision
http://www.owl-ontologies.com/PersonLibrary.owl#beer
http://www.owl-ontologies.com/PersonLibrary.owl#bubbleBobble
http://www.owl-ontologies.com/PersonLibrary.owl#comedy
http://www.owl-ontologies.com/PersonLibrary.owl#Lordi
http://www.owl-ontologies.com/PersonLibrary.owl#stereo
```

Figure 31. The dialog box for defining free queries.

These two query construction options were considered to be adequate for the purposes of VantagePoint end users. With these query methods it is possible to access any data contained by the loaded model. A big advantage is that VantagePoint provides a possibility to execute even complex queries through a simple graphical interface without any knowledge of the ontology query languages RDQL or SPARQL. In addition, VantagePoint offers an option for advanced users to define queries that are not restricted in any way and thereby retrieve any kind of information from the loaded model.

5. The validation

The term validation conveys a sense that a scientific effort must be justified in some logical, objective, and algorithmic way (Kleindorfer et al. 1998). Adrion et al. (1982) have defined validation as the determination of the correctness of the software produced from a development project with respect to the user needs and requirements. There is no one right way to perform validation. Different kinds of systems require different validation techniques and objectives. For example, Adrion et al. (1982) have stated that a program the malfunction of which would have severe consequences justifies greater effort in their validation. Thus, software used in the control of airplane landings requires higher confidence in its proper functioning than does a car pool locator program. In this chapter a proper validation concept for VantagePoint is defined. The validation concept was developed considering the characteristics of the constructed approach as well as the resources available in this study. At the end of the chapter the results of the validation are presented and analyzed.

As mentioned earlier, one of the key requirements for VantagePoint was to offer the possibility to create semantic models of networked environments through graphical editing operations. The resultant models are described by OWL files that define rooms, devices, services and persons contained by the environment. In this way VantagePoint allows the semantic modelling and interactive simulation of physical real-world environments. Kleijnen (1995) has mentioned that the validation is concerned with determining whether the conceptual simulation model is an accurate representation of the system under study. In this case the validation is carried out to clarify if it is possible to model an existing real-world intelligent environment with VantagePoint. Another goal of this validation is to find out how effectively the query interface provided by VantagePoint allows users to obtain particularly service-related information from the semantic model. This validation concept was selected because, until now, only imaginary environments had been modelled with VantagePoint and thus there were no experiences concerning how the approach and its operations would perform in an authentic end-user scenario. Although the validation concept was quite small-scale, it was considered to be adequate for the purposes of VantagePoint and feasible with the resources available in this study.

Kleijnen (1995) has stated that sometimes the most challenging difficulty in simulation validation is to obtain relevant data. Without relevant data it is usually difficult to determine how accurate the simulation is, since there is no possibility to compare the simulation to the original system. This is often the case when simulating abstract phenomena. In this study the problem of obtaining relevant data was solved by selecting an existing environment for the validation. To be more specific, the selected environment was an intelligent home laboratory owned by a partner of the project Amigo. Unfortunately, the project partner considers some information related to this laboratory to be confidential, and therefore the name of the partner and the location of the laboratory cannot be revealed in this study. Yet this environment was considered to be suitable for the validation since it contained all the characteristics of an intelligent home environment, including rooms, devices and services. In addition, an accurate modelling of this environment was possible because specific enough information was provided for the purposes of the Amigo project. This information contained photos of the environment, measurements of the rooms and the locations of the devices providing services. A ground plan of the home laboratory is presented in Figure 32. As can be seen, the ground plan includes the measurements of the rooms and the positions of the items that provide services.

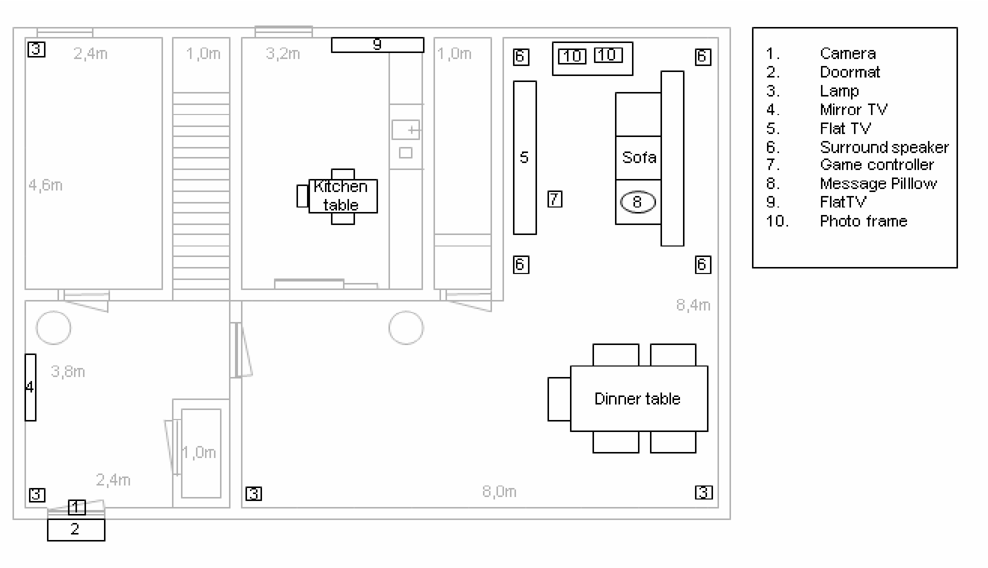


Figure 32. A ground plan of the intelligent home environment used in the validation. (Amigo project material.)

5.1 The completion of the validation and the acceptability criteria

Adrion et al. (1982) have stated that careful planning is critical to successful validation. Therefore, also in this study, the validation was designed beforehand and the validation process was divided into six steps, which are described below.

Step 1: Selecting the environment to be used in the validation. As discussed before, this is one of the most critical phases in the validation process. To be able to perform a successful validation, an adequate amount of specific information about the environment used in the validation must be available.

Step 2: Modelling the physical structure of the environment used in the validation. This is carried out by drawing the rooms of the intelligent home laboratory with VantagePoint. The result of this phase is an OWL file defining the physical structure of the environment.

Step 3: Creating the 2D symbols and the 3D icons for the items that will be included in the visualization of the environment. Once the symbols and the icons are drawn using a graphical editor application, they are added to the icon library. After this the icons are available in VantagePoint's "add item" dialog box.

Step 4: Defining service descriptions for the devices. The authentic service descriptions of the devices are not available for this study, and thus they must be created. This is carried out by defining imaginary service descriptions using the OWL-S service description language. Validation-wise it makes no difference if the service descriptions are authentic or imaginary, as long as they are described with the OWL-S language and can be found through the pre-defined service queries.

Step 5: Adding the devices and furniture to the model. This is carried out by using the "add item" operation of VantagePoint, in which the items are named and dragged into their proper positions. The result of this step is an OWL file which defines, as well as the physical structure of the environment, the devices and the services contained in the environment.

Step 6: Adding a person to the model. The person represents an imaginary resident of the house. As mentioned, an important objective of this validation is to test whether the services contained in different rooms can be found by the means provided by VantagePoint. A simple service discovery test is executed by moving the person from one room to another and observing if VantagePoint is able to find the services contained in these rooms.

In addition to careful planning, clearly stated objectives are also critical to successful validation (Adrion et al. 1982). Kleijnen (1995) has mentioned that the validation cannot be assumed to result in a perfect model, since the perfect model would be the real system itself. Instead, the model should be "good enough", which depends always on the goal of the model. In the case of VantagePoint, "good enough" is specified through the four acceptability criteria that are described below.

Criterion 1: With VantagePoint it is possible to accurately model the physical structure of the environment used in the validation. This consists of modelling different rooms and other areas with their exact measurements and correct shapes.

Criterion 2: With VantagePoint it is possible to model the items that are located in the environment selected for the validation. The application allows these items to be positioned in their exact locations and to be represented with illustrative icons.

Criterion 3: VantagePoint provides a means to access various service descriptions through the items shown in the visualization.

Criterion 4: VantagePoint enables simple service discovery. The application allows the user to obtain information about the services contained in different areas of the environment.

5.2 The results of the validation

As mentioned in the previous section, the validation was carried out by following the six steps that were defined beforehand. In addition, four acceptability criteria were defined in order to assess the success of the validation. In this chapter the results of the validation are explained by evaluating the completion of the validation tasks against the four acceptability criteria described in the previous chapter. In addition, the deficiencies of VantagePoint that were revealed during the validation process are presented in this chapter.

The modelling of the physical environment was completed successfully. The drawing operation worked smoothly and allowed adding areas with the correct measurements and desired shapes. In the visualization, a grid divided the design space into 20-centimeter-long sections, which was a big advantage in the area creation process. The grid helped the drawing task by allowing areas to be created with exact measurements and locations. The areas contained by the environment were all square in shape and thus easy to model. However, VantagePoint would also offer the possibility to create areas in other shapes, but the accuracy of this operation could not be tested within this validation.

The adding of the devices and furniture to the model was also perceived to be a relatively straightforward operation. After the new icons and service descriptions were added into the icon library and the new service descriptions into the service description repository, the items could easily be included in the semantic model by using the “add item” operation offered by VantagePoint. The grid with an accuracy of 20 centimetres was proven to enable an adequate enough positioning of items. In addition, VantagePoint offered an operation to set the directions of the items. The only fault noticed during the phase of adding the devices and furniture was that when an item was located on top of another item, it could not be shown properly in the isometric view. However, this had no great semantic effect on the simulation, since according to the context model presented in Section 4.1.2, items can not be contained by other items in the VantagePoint world. Thus, it made no semantic difference if the pillow that was located on top of the sofa in the ground plan presented in Figure 32 had to be located next to the sofa to ensure its visibility in the 3D visualization. In addition, as mentioned, this problem occurred only in the isometric view; in the edit view it was possible to represent items on top of other items. The problem with the isometric view was caused because of the algorithm used to create the 3D visualization. This fault will hopefully be corrected by the next prototype version. The visualization of the semantic model of the environment is presented in Figure 33, in which both the edit view and the isometric view are shown.

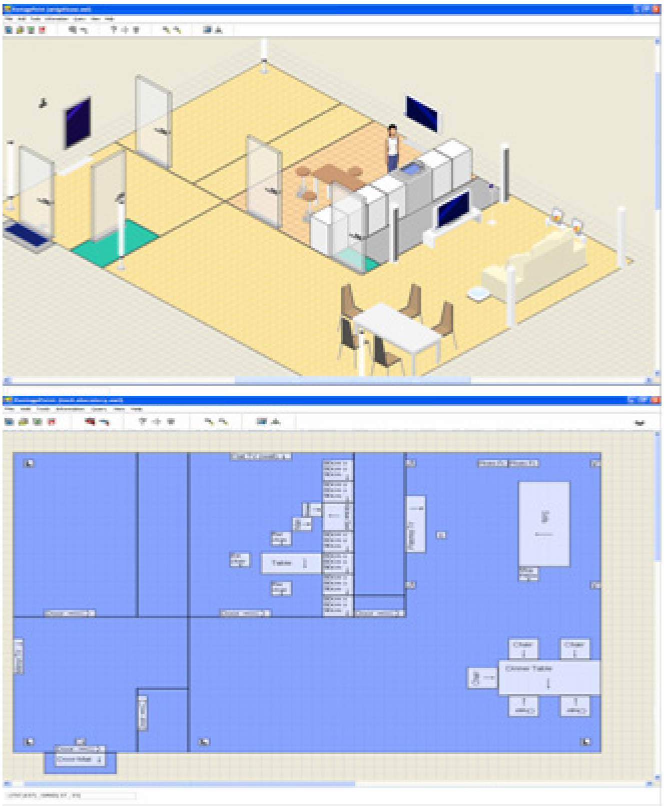
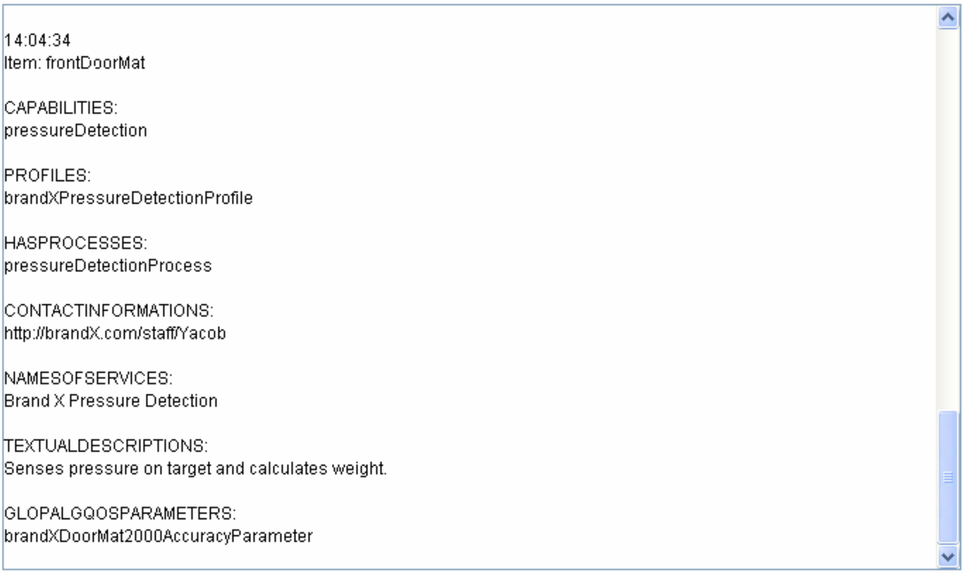


Figure 33. Two screenshots from VantagePoint presenting the visualization of the environment used in the validation.

Once the items were added to the model, the next phase was to test whether information about the service descriptions attached to different devices could be retrieved through the operations offered by VantagePoint. This test was implemented by defining query sets that contained queries to retrieve detailed information about different service descriptions. These queries were executed through the graphical query interface by clicking on the items in the visualization. As can be seen from Figure 34, the queries and the graphical query interface worked well, and were effectively able to retrieve information from the

OWL-S service descriptions. An example of such OWL-S service description can be found in Appendix A.



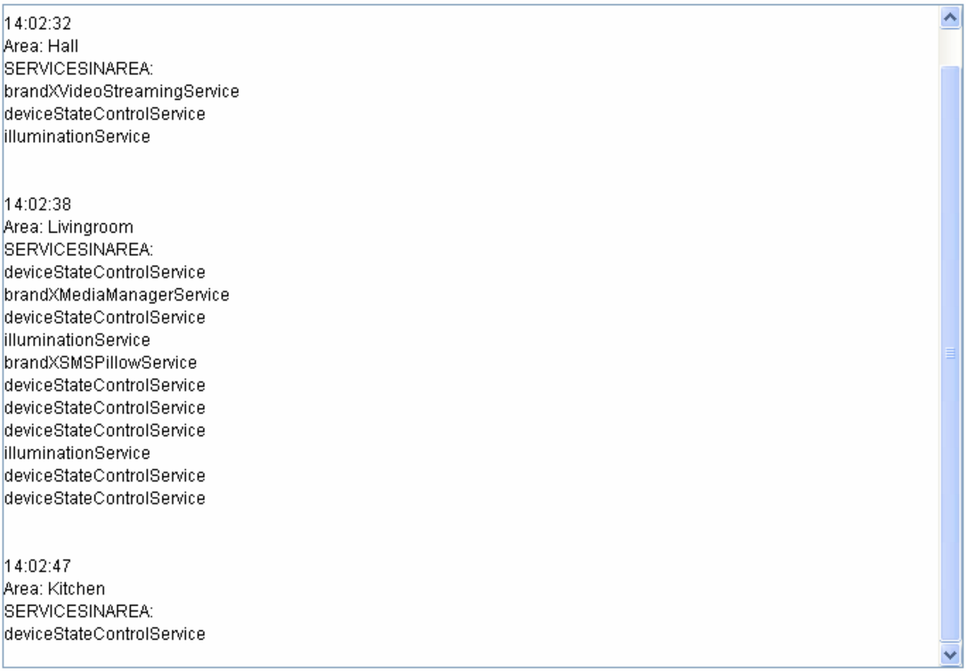
```
14:04:34
Item: frontDoorMat

CAPABILITIES:
pressureDetection

PROFILES:
brandXPressureDetectionProfile

HASPROCESSES:
pressureDetectionProcess

CONTACTINFORMATIONS:
http://brandX.com/staff/Yacob

NAMESOFSERVICES:
Brand X Pressure Detection

TEXTUALDESCRIPTIONS:
Senses pressure on target and calculates weight.

GLOPALGQOSPARAMETERS:
brandXDoorMat2000AccuracyParameter
```

Figure 34. A sample query result from a service description attached to a doormat.

According to the fourth acceptability criterion, VantagePoint should be able to perform a simplified service discovery by printing the services contained in each area of the environment. The service discovery was implemented by adding a method to VantagePoint code that is executed every time the occupant of the house is moved into another room. This method uses the query interface provided by VantagePoint and executes a query which should return all the services contained in the particular room that the person enters. Finally, this method prints the detected services to the screen.

As mentioned in Section 4.1.1, VantagePoint should support people who need ontologies and contextual data in their work. For example, device manufactures who are creating new service descriptions for their devices may find it beneficial to be able to test their services by using the query interfaces provided by VantagePoint. A simplified service discovery is an example of such a situation, where new devices are added into the model, and it is tested whether these devices can be found.

The simple service discovery worked faultlessly and in Figure 35 the results of queries that were executed to perform the service discovery are presented. As can be seen, in this case the person has moved first from the hall to the living room, and then from the living room to the kitchen.



```
14:02:32
Area: Hall
SERVICESINAREA:
brandXVideoStreamingService
deviceStateControlService
illuminationService


14:02:38
Area: Livingroom
SERVICESINAREA:
deviceStateControlService
brandXMediaManagerService
deviceStateControlService
illuminationService
brandXSMSPillowService
deviceStateControlService
deviceStateControlService
deviceStateControlService
illuminationService
deviceStateControlService
deviceStateControlService


14:02:47
Area: Kitchen
SERVICESINAREA:
deviceStateControlService
```

Figure 35. Query results as the person moved from one room to another.

Although the service discovery tested in this validation proved to be successful, it was quite restricted. The service discovery was only able to track services to the accuracy of room level. This may cause problems if more precise service discovery was required or the modelled environment contained extremely large areas. Also, in many cases the discovered services should be ranked according to user preferences, for example. This would ensure that the most suitable services for different situations could be discovered more easily. There have been some preliminary test approaches developed to perform a more sophisticated service discovery, but they had not been integrated into VantagePoint at the time of this study.

The validation also revealed some flaws in the interaction operations of VantagePoint. For example, when performing moving, adding and deleting operations, a five-second delay occurred. This delay did not prevent the execution of these operations, but it started to cause frustration as operations were executed repeatedly. However, the delay was not considered to be a significant problem as these operations are usually executed only when an environment is being modelled, which is normally carried out only once. In addition, these delays were not caused by VantagePoint itself, but by the reasoning engine exploited in VantagePoint and provided by Jena. Another deficiency that was discovered was the lack of an undo operation. During the modelling of the environment, the accuracy requirements caused mistakes when performing some of the operations. However, VantagePoint offered no possibility to cancel the unsuccessful operations, which was considered to be a clear disadvantage. Finally, the lack of a zoom facility was perceived to hinder the efficient modelling of the given environment. Especially in the edit view, there would have been a clear need to see both an overall picture of the environment and a more detailed vision of the environment when performing accurate editing operations. In the isometric view the lack of a zoom was not that obvious. All of these faults will hopefully be repaired by the next prototype version. In particular, the lack of undo and zoom operations is considered a significant deficiency and designs for adding these operations into VantagePoint have already been made.

Despite these deficiencies, the validation in general can be considered to be successful. VantagePoint was able to accurately model an existing intelligent environment and to find the services contained in this environment, and thus passed the acceptance criteria defined in the previous chapter. With the exception of the problems mentioned in the previous paragraph, the interaction operations also worked faultlessly and effectively. In addition it must not be forgotten that VantagePoint is still a prototype and under continuous development work. In conclusion, VantagePoint proved to be a functional approach for creating, visualizing and editing semantic models that describe intelligent home environments.

Besides this validation, VantagePoint has been also evaluated in the Amigo project review, which was held in Eindhoven, in the Netherlands on 28–29 November 2006. In general the reviewers regarded VantagePoint as an interesting and useful tool, which will surely evoke interest among researchers. Some excerpts from the Amigo project review report, relating to VantagePoint, can be found in Appendix C.

6. Conclusion and discussion

This study is concluded in this chapter. The research problem and questions derived from it are answered. In addition, the achievements and the limitations of this study are discussed. At the end of this chapter, possible research leads for the future are suggested.

This study aimed at the construction of a prototype software tool which is able to interactively visualize ontologies, particularly in the domain of networked home environments. This tool was implemented in a construction process which was divided into two sub-processes: specification and implementation. In the specification process, a description of the target state was produced, and in the implementation process this state was realized. Overall, the construction process was successful and the work proceeded without any significant problems or delays. At the end of the study, the implemented approach was validated in a test case.

The research problem of the study was defined as the following:

> **How can ontologies be visualized, particularly in the domain of networked home environments, and how should interaction between the visualization approach and the user be implemented?**

The research problem was solved by answering the following three research questions:

1. *What special requirements does the networked home domain set for the visualization approach?*
2. *How should context information be presented and managed in the visualization?*
3. *How should the interaction be implemented and managed?*

The first question was answered by examining the theory of networked home environments. The theory suggested that a networked home environment comprises rooms, devices, services and persons. In addition, it was revealed that these kinds of environments can best be modelled by using ontologies and describing the models with the OWL ontology language. On the basis of the theory, some general requirements for the visualization approach were conducted. For example, it was defined that the visualization should illustratively represent the basic elements of networked home environments, including rooms, devices and persons. In addition, the visualization should be able to read ontological data described by the OWL language, and visualize it. Finally, the visualization should provide a means to manage, save and retrieve the environment models described by ontologies.

The domain of networked home environments also required that the constructed approach should be able to effectively represent contextual information in the visualization. For example, the importance of spatial relationships is highlighted in this particular domain and, therefore, the traditional graph visualization algorithms that are mainly focused on representing the abstract relationships between classes were not considered to be the best possible solutions for this particular problem. Instead, in order to provide a more believable and illustrative view, it was decided to implement the approach to visualizing data in a more realistic manner. The items, for example, were visualized using realistic three-dimensional icons to make the iconic mapping from objects to their physical shapes more direct and intuitive. In addition, two distinct views were created. The two-dimensional edit view was a “ground plan” view of the environment being visualized. This view enabled effective and accurate discovery of the spatial relationships between different elements, perception of the exact positions of various instances, and accurate editing operations. The isometric view was created to visualize environments from an isometric perspective. This 3D view enabled to obtain a better overall picture of an environment, and it was assumed to impress the users and encourage people to start using the constructed approach.

The third research problem was about interaction. The interaction between the user and the visualization was considered to be an important feature, because the approach was supposed to be able not only to visualize semantic models of environments, but also to help users to create them. The interaction was realized by giving users extensive possibilities to manipulate the visualization through multiple drag-and-drop and drawing operations. With these operations it was possible, for example, to add, move and remove different elements from the visualization. In addition, extensive query construction possibilities enabled any kind of information to be extracted from the visualized model. Through these versatile interaction possibilities, users were also assumed to gain a better insight into ontologies and extend their possibilities to manipulate the underlying ontological models.

It was intended to solve the main research problem of the study by answering these sub-questions. The answer to the main question was concretized by implementing an application called VantagePoint, which was supposed to act as a proof-of-concept of how these problems can be solved. On the basis of these questions, a well-defined target state for this study was defined. When comparing the target state with the current state of the study, it can be seen that the target state was realized quite well. VantagePoint fulfils its requirements by effectively visualizing ontologies of the given domain field. In addition, the suitability of VantagePoint for also modelling existing real-world environments was proved through the validation, which was performed after the construction process.

VantagePoint was constructed because there was a clear need for such an application. As was seen in Section 3.5, all the assessed ontology visualization approaches were quite similar, as they used the same visualization technique and as they were all domain independent. However, as mentioned in Section 3.5.5, the chosen visualization technique should be relevant to the given problem and support the user’s goal in viewing the representation. VantagePoint fills a certain niche that is not supported by the other ontology visualization and development tools and it serves a broad user group. VantagePoint models and visualizes networked home environments described by ontologies in a convenient way and effectively represents different physical structures and spatial relationships between entities. In addition, it enables the attachment of various data elements to the visualized instances and provides access to these data elements through well-defined query interfaces. Currently, there are no similar approaches available.

VantagePoint serves multiple purposes. To begin with, it provides easy access to the complex world of ontologies and OWL language. It enables the building of semantic models of various environments without requiring any particular knowledge of ontologies. Thus, VantagePoint helps people who are not familiar with ontologies and the OWL language. VantagePoint shortens the gap from beginner to intermediate OWL ontology reader by visualizing instances in their actual locations and making OWL ontologies more interesting and concrete, and above all easier to comprehend.

Perhaps the main user group of VantagePoint is application developers who need ontologies and contextual data in their work. For example, developers, who create service descriptions for different devices, are able to create semantic models of different environments and thus simulate abstract or existing environments with VantagePoint. By populating the models with illustrative 3D icons, the service-hosting devices can be concretized and located in their exact positions. With VantagePoint, various changes in context can be seen in a more illustrative manner than observing changes in raw OWL files. Developers are able to test and demonstrate the service descriptions in different scenarios. VantagePoint works as a virtual test laboratory allowing the developers to see operations as in real life and notice practical errors better without expensive test laboratories. A concrete usage scenario for VantagePoint is simple service discovery, in which the correctness of OWL-S service descriptions can be tested by observing whether VantagePoint is able to find these services in different situations.

Because VantagePoint was implemented as open source software, application developers are able to add their own functionality and/or plug-ins that extend the functionality of the application or modify it to better serve their own particular purposes. In that case, the core functionality of VantagePoint is to act as an interface between ontological models and external applications and provide such services as managing and storing context information, providing query interfaces, visualizing the model and providing graphical editing operations to manipulate the model.

Although VantagePoint was originally designed to fulfil the requirements set by the domain of networked home environments, it could be modified to also visualize ontologies of other domains. An example of such a case could be the modelling of organizational structures. With VantagePoint, different organizational structures could be easily and effectively modelled and visualized. Areas could represent different departments of an organization and items could represent employees. Employees could be represented using real photographs which would enhance identification. In addition, additional information could be attached to employees that could be accessed through various query and printing operations. This would enable the execution of queries about employees that have certain kind of expertise, for example. Also, various changes in organizational structures could be easily updated through graphical editing operations.

VantagePoint does set some limitations on the ontologies that can be visualized. As mentioned in Section 4.1.2, only individuals that belong to the predefined VisualComponent class can be visualized. In addition, these individuals must contain certain properties to be able to be visualized. So VantagePoint cannot be characterized as a domain-independent ontology visualization approach, as it sets quite strict requirements for ontologies that can be visualized. However, this is not considered as a disadvantage, since there are currently several domain-independent ontology visualization approaches available and there is no point in reinventing the wheel. VantagePoint fills its own niche and is useful for its end users.

The validation revealed some deficiencies in VantagePoint. For example, it was discovered that the delay related to some editing operations causes frustration in an actual use situation and in the 3D visualization it is impossible to represent items that are situated on top of other items. In addition, the lack of zoom and undo operations were considered to be clear disadvantages. However, it must not be forgotten that VantagePoint is still in a prototype phase and will evolve as new prototype versions are being released. Thus the problems that were revealed during the validation process will certainly be repaired in future versions of VantagePoint.

6.1 Achievements of the study

The main achievement of this study is a prototype ontology visualization tool particularly for the domain of networked home environments. The construction of such a tool was considered necessary because currently there are no similar approaches available and a clear need exists for such an application. The final phase of the study was the validation, in which an existing intelligent environment was modelled using the constructed approach. The results of the validation were an OWL description of the environment that was used in the validation and OWL-S service descriptions that were created for the purposes of the validation.

The construction process was iterative by its nature. This meant that the specification and implementation processes were partly performed in parallel. The approach was constructed according to the requirements defined in Section 4.1.1. Within these requirements the most critical principles considering the graphical appearance and user interaction of the approach were defined. In the next phase the different elements of the approach were divided into component clusters, which are described in Section 4.1.2. In addition, a class diagram was created and it is presented in Section 4.1.3.

The Model-View-Controller architecture was selected as the user interface framework of VantagePoint, because it was considered to suit well the given visualization problem. The Model-View-Controller architecture and the slightly modified Model-View-Controller framework used in VantagePoint are described in Section 4.1.3. The implementation process produced multiple prototype versions of the application. The first and the most recent prototype versions are presented in Section 4.2.

The final task was to validate the approach in a test case in which an existing networked home environment was modelled using the constructed approach. The validation was carried out by carefully planning the different steps of the process and determining suitable acceptability criteria. The result of the validation was a semantic model of the chosen environment. This model was compared to the acceptability criteria and a conclusion was reached as to whether the validation was successful or not. The validation and its results are described in detail in Chapter 5.

6.2 Limitations of the study

As mentioned, the ontology visualization approach constructed in this study is currently in a prototype phase. The resources available for this study did not enable any kind of testing work to be performed. For example, various performance and reliability tests would have produced valuable information about the behaviour of the approach in different situations. For instance, it would have been interesting to test how effectively VantagePoint is able to visualize extremely large ontologies that contain more than 10 000 RDF- triplets. In addition, various usability tests would have given valuable information about the learnability and functionality of the various user interaction operations and overall quality of the user interface. As the development work on the approach continues and as VantagePoint develops to a more mature level, multiple test procedures will surely be executed.

The validation was performed by the author of this study, who also participated closely in the design and implementation processes of the constructed approach. In many cases, the developers of different approaches tend to understate the deficiencies of their creations. Thus, the objectivity of the results of the validation could be considered debatable. However, there exists no clear motivation to skew the results, since the validated application is just a prototype and is not targeted at commercial markets. Therefore the results of the validation can be considered to be relatively reliable. The validation was also quite restricted, containing only one modelled environment. In most validations, more extensive material is used in order to get more reliable results. However, with the resources available in this study, this limited validation was considered to be adequate.

6.3 Future research leads

This study evoked numerous possible future research leads. As mentioned, the constructed approach is still in a prototype phase, requiring more prolonged research work. For example, an interesting research topic would be to find out how VantagePoint should be modified in order to be able to visualize ontologies of other domains? This could include, for example, adding new views and interaction methods to VantagePoint.

As mentioned, VantagePoint is targeted especially at application developers who need ontologies and contextual data in their work. A possible future research lead could be interviewing end users in order to get more detailed information on how VantagePoint could better support their work. Furthermore, VantagePoint could be extended by adding more dynamic elements to it. House residents could autonomously move around home environments according to predefined scenarios. This more dynamic environment would enable more authentic and extensive service discovery or composition testing. In addition, the dynamic elements would improve the ability of this approach to simulate real world phenomenon and environments.

Currently VantagePoint presents the results of different queries in a textual list. However, the visualization of query results would make them more illustrative and understandable. Therefore, a possible research lead would be to find a suitable visualization method for presenting the query results provided by VantagePoint.

In the future it would be interesting to research whether VantagePoint could be extended as context control system. This would mean that besides modelling and visualizing networked home environments, VantagePoint could provide a means to control real world physical environments. In consequence, VantagePoint could be used for example to turn different devices on or off, or control the heating system of a house.

References

Adrion, W.R., Branstad, M.A. & Cherniavsky, J.C. (1982). Validation, verification and testing of computer software. ACM Computing Surveys, Vol. 14, No. 2, 159–192.

Aiello, M., Marchese, M., Busetta, P. & Calabrese, G. (2005). Opening the Home: a Web Service Approach to Domotics. Proceedings of the IADIS International Conference on Applied Computing, Algarve, Portugal, 22–25 February 2005.

Alani, H. (2003). TGVizTab: an ontology visualisation extension for Protege. Proceedings of Knowledge Capture (K-Cap'03), Workshop on Visualization Information in Knowledge Engineering, Sanibel Island, Florida, USA, 25–26 October 2003.

Amigo (2005). Amigo short project description. Information Society Technologies, http://www.hitech-projects.com/euprojects/amigo/publications/IST-004182%20Amigo-IP%20short%20project%20description.pdf (Referenced April 1, 2007).

Aref, M. & Zhou, Z. (2005). The Ontology Web Language (OWL) for a Multi-Agent Understating System. IEEE 2005 International Conference on Integration of Knowledge Intensive Multi-Agent Systems: Modeling, Exploration, and Engineering, Waltham, Massachusetts, USA, 18–21 April 2005, 586–591.

Baymani, S. & Strifeldt, E. (2005). Carbonara – a semantically searchable distributed repository. Master's thesis, KTH Royal Institute of Technology.

Berners-Lee, T., Hendler, J. & Lassila, O. (2001). The Semantic Web. Scientific American, Vol. 284, No. 5, 28–37.

Boekhorst, F. (2002). Ambient intelligence, the next paradigm for consumer electronics: How will it affect silicon? In: Proceedings of the International Solid-Stole Circuits Conference (ISSCC '02), San Francisco, CA, USA, 4–6 February 2002.

Broekstra, J., Eberhart, A., Haase, P. & Volx, R. (2004). A Comparison of RDF Query Languages. Proceedings of the Third International Semantic Web Conference. Hiroshima, Japan, 7–11 November 2004.

Broens, T.H.F. (2004). Context-aware, Ontology based, Semantic Service Discovery. Masters thesis, University of Twente.

Brown, P.J., Bovey, J.D. & Chen, X. (1997). Context-aware applications: From the laboratory to the marketplace. IEEE Personal Communications, Vol. 4, No. 5, 58–64.

Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A. & Wilkinson K. (2004). Jena: Implementing the semantic web recommendations. In: Proceedings of the 13th international World Wide Web Conference on Alternate Track Papers and Posters, New York, NY, USA, 19–21 May 2004.

Chen, J.X. (2003). Guide to Graphics Software Tools. Berlin: Springer Verlag.

Chen, H., Perich, F., Finin, T. & Joshi. A. (2004). SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. Proceedings of First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services. Boston Massachusetts, USA, 22–26 April 2004, 258–267.

Chen, J.X., Rine, D. & Simon, H.D. (1996). Advancing Interactive Visualization and Computational Steering. IEEE Computational Science and Engineering, Vol. 3, No. 4, 13–17.

Chuah, M.C. & Roth, S.F. (1996). On the semantics of interactive visualizations. Proceedings of the Information Visualization, IEEE, San Francisco, USA, California, 28–29 October 1996, 29–36.

Davies, J., Studer, R. & Warren, P. (2006). Semantic Web Technologies: Trends and Research in Ontology-based Systems. West Sussex: Wiley Publishing.

Daymon, C. & Holloway, I. (2002). Qualitative Research Methods in Public Relations and Marketing Communications. London, United Kingdom: Routledge.

Dey, A.K. & Abowd, G.D. (2000). Towards a better understanding of context and context awareness. Proceedings of the CHI 2000 Workshop on The What, Who, Where, and How of Context-Awareness, The Hague, The Netherlands, 2–3 April 2000.

Doan, A., Madhavan, J., Domingos, P. & Halevy, A. Y. (2002). Learning to Map between Ontologies on the Semantic Web. Proceedings of the Eleventh International World Wide Web Conference, Honolulu, Hawaii, USA, 07–11 May 2002, 662–673.

Duke, D.J., Brodlie, K.W., Duce, D.A. & Herman, I. (2005). Do you see what I mean? Computer Graphics and Applications, IEEE, Vol. 25, No.3, 6–9.

Ensing, J. (2002). Software architecture for the support of context aware applications.
http://citeseer.ist.psu.edu/cache/papers/cs/30466/http:zSzzSzwww.extra.research.philips.comzSzpublzSzrepzSznl-urzSzur2002-841.pdf/ensing02software.pdf (Referenced April 16, 2007).

Ernst, N.A., Storey, M.-A. & Musen, M. (2003). Addressing cognitive issues in knowledge engineering with Jambalaya. Workshop on Visualization in Knowledge Engineering, Sanibel Island, Florida, USA, 25–26 October 2003, 26–30.

Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D.L. & Patel-Schneider, P.F. (2001). OIL: an ontology infrastructure for the Semantic Web. Intelligent Systems, IEEE, Vol. 16, No. 2, 38–45.

Fernández-Vara, C., Zagal, J. & Mateas, M. (2005). Evolution of Spatial Configurations in Videogames. Proceedings of the Digital Games Research Association (DiGRA) Conference, Vancouver, British Columbia, Canada, 16–20 June 2005.

Fluit, C., Sabou, M. & van Harmelen, F. (2002). Ontology-based Information Visualisation. In: Visualising the Semantic Web, Berlin: Springer Verlag.

Foley, J.D. & Van Dam, A. (1982). Fundamentals of Interactive Computer Graphics. Boston: Addison-Wesley.

Foley, J.D., van Dam, A., Feiner, S. & Hughes, J. (1990). Computer Graphics: Principles and Practice. 2nd ed., Boston: Addison-Wesley.

Georgantas, N., Mokhtar S.B., Tartanoglu, F. & Issarny, V. (2005). Semantics-aware services for the mobile computing environment. Lecture Notes in Computer Science, 3549, 1–35.

Gershon, N. & Eick, S.G. (1998). Guest Editors‘ Introduction: Information Visualization. The Next Frontier. Journal of Intelligent Information Systems, Vol. 11, No. 3, 199–204.

Gruber, T. (1993). A translation approach to portable ontologies. Knowledge Acquisition, Vol. 5, No. 2, 199–220.

Gu, T., Wang, X.H., Pung, H.K. & Zhang, D.Q. (2004). Ontology Based Context Modeling and Reasoning using OWL. In: Proceedings of the 2004 Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS2004), San Diego, California, USA, 18–21 January.

Guarino, N. (1997). Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In: Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology. Berlin: Springer Verlag, 139–170.

Guarino, N. (1998). Formal ontology and information systems. Formal Ontology in Information Systems, Amsterdam: IOS Press, 3–15.

Harris, S. (2005) SPARQL Query Processing with Conventional Relational Database Systems, International Workshop on Scalable Semantic Web Knowledge Base Systems, New York City, USA, 20 November 2005, 235–244.

Hart, C. (1998). Doing a Literature Review. London: Sage.

Herman, I., Melancon, G. & Marshall, M. (2000). Graph visualization and navigation in information visualization: a survey. IEEE Transactions on Visualization and Computer Graphics, Vol. 6, 24–43.

Hesselman, C., Tokmakoff, A., Pawar, P. & Iacob, S. (2006). Discovery and Composition of Services for Context-Aware Systems. Proceedings of the 1st European Conference on Smart Sensing and Context (EuroSCC'06), Enschede, The Netherlands, 25–27 October 2006.

Johnson, C., Parker, S.G., Hansen, C., Kindlmann, G.L. & Livnat, Y. (1999). Interactive Simulation and Visualization. IEEE Computer, Vol. 32, No. 12, 59–65.

Järvinen, P. & Järvinen A. (2000). Tutkimustyön metodeista. Tampere: Opinpajan kirja.

Kalaoja, J., Kantorovitch, J., Carro, S., Miranda, M.J., Ramos, Á. & Parra, J. (2006). The Vocabulary Ontology Engineering. 8th International Conference on Enterprise Information Systems, Paphos, Cyprus, 23–27 May 2006.

Katifori, A., Torou, E., Halatsis, C., Lepouras, G. & Vassilakis, C. (2006). A Comparative Study of Four Ontology Visualization Techniques in Protege: Experiment Setup and Preliminary Results. Proceedings of the 10th International Conference on Information Visualisation, London, UK, 5 –7 July 2006, 417–423.

Kleijnen, J.P.C. (1995). Verification and Validation of Simulation Models. European Journal of Operational Research, Vol. 82, 145–162.

Kleindorfer, G.B., O'Neill, L. & Ganeshan, R. (1998). Validation in Simulation: Various Positions in the Philosophy of Science. Management Science, Vol. 44, No. 8, 1087–1099.

Krasner, G.E. & Pope, S.T. (1988). A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 system. Journal of Object Oriented Programming, Vol. 1, No. 3, pp. 26–49.

Lanzenberger, M. & Sampson, J. (2006). AlViz – A Tool for Visual Ontology Alignment. Proceedings of the 10th International Conference on Information Visualisation, London, UK, 5–7 July 2006, 430–440.

Lassila, O. & Swick, R. (2000). Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, World Wide Web Consortium, Boston, www.w3.org/TR/REC-rdf-syntax (Referenced March 13, 2007).

Lehti, P. & Fankhauser, P. (2004). XML data integration with OWL: Experiences and challenges. In: Proceedings of Symposium on Applications and the Internet, Tokyo, Japan, 26–30 January 2004, 160–170.

Lijun, X., Yao, Z. & Bin, S. (2006). Ontology Construction for Scientific Visualization. First International Multi-Symposiums on Computer and Computational Sciences, Vol. 1, 778–784.

Luymes, D.T. (2000). The rhetoric of visual simulation in forest design: some research directions. In: Sheppard S.R.J. & Harshaw, H.W. (Eds.) Forests and Landscapes: Linking Ecology, Sustainability, and Aesthetics. IUFRO Research Series, No. 6. Wallingford, UK: CABI Publishing, 191–204.

Macedonia, M. (2000). Using technology and innovation to simulate daily life. Computer, Vol. 33, No.4, 110–112.

Mackinlay, J.D. (2000). Opportunities for information visualization. Computer Graphics and Applications, IEEE, Vol. 20, No. 1, 22–23.

Marr, D. & Nishihara, K. (1978). Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes. Proceedings of the Royal Society of London. Series B, Biological Sciences, Vol. 200, No. 1140, 269–294.

Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N. & Sycara, K. (2004). Bringing semantics to Web Services: The OWL-S approach. Proceedings of First International Workshop on Semantic Web Services and

Web Process Composition (SWSWPC 2004), San Diego, California, USA, 6 – 9 July 2004.

Masuoka, R., Labrou, Y., Parsia, B. & Sirin, E. (2003). Ontology-Enabled Pervasive Computing Applications. IEEE Intelligent Systems, Vol. 18, No. 5, 68–72.

McBride, B. (2002). Jena: A semantic web toolkit. IEEE Internet Computing, Vol. 6, No. 6, 55–59.

McDonald, J.A. (1982). Interactive Graphics for Data Analysis. Doctoral dissertation, Stanford University, Stanford, California, USA.

McGuinness, D. & van Harmelen, F.D. (2004). OWL Web Ontology Language Overview, W3C Recommendation. http://www.w3.org/TR/2004/REC-owl-features-20040210/ (Referenced March 15, 2007).

Mokhtar, S.B., Georgantas, N. & Issarny, V. (2005). Ad hoc composition of user tasks in pervasive computing environments. In: Proceedings of the 4th Workshop on Software Composition (SC'05), Edinburgh, Scotland, 9 April 2005.

Mutton, P. & Golbeck, J. (2003). Visualization of Semantic Metadata and Ontologies. Proceedings of the Seventh International Conference on Information Visualization, London, UK, 16–18 July 2003, 300–305.

Nielsen, J. (1993). Usability Engineering. Boston, MA: Academic Press.

O'Leary, D.E. (1998). Using AI in Knowledge Management: Knowledge Bases and Ontologies. IEEE Intelligent Systems, Vol. 13, No. 3, 34–39.

Ouwerkerk, A. & Stuckenschmidt, H. (2003). Visualizing RDF Data for P2P Information Sharing. In: Proceedings of the workshop on Visualizing Information in Knowledge Engineering, VIKE'03, Sanibel Island, FL, USA, 6 July 2004.

Perez, J., Arenas, M. & Gutierrez, C. (2006). Semantics and complexity of SPARQL. In: Proceedings of the 5th International Semantic Web Conference (ISWC-06), Athens, GA, USA, 5–9 November 2006.

Perlin, K. & Fox, D. (1993). Pad: an alternative approach to the computer interface. Proceedings of ACM SigGraph ’93, Anaheim, CA, USA, 1–6 August 1993.

Petriu, E.M., Georganas, N.D., Petriu, D.C., Makrakis, D. & Groza, V.Z., (2000). Sensor-based information appliances. IEEE Instrumentation and Measurement Magazine Vol. 3, No. 4, 31–35.

Prud’hommeaux, E. & Seaborne, A. (2004). SPARQL Query Language for RDF. W3C Working Draft, http://www.w3.org/TR/rdf-sparql-query/, (Referenced 13. April 2007).

Ranganathan, A., McGrath, R.E., Campbell, R. & Mickunas, D.M. (2003). Ontologies in a pervasive computing environment. Workshop on Ontologies in Distributed Systems, IJCAI 2003, Acapulco, Mexico, 9–10 August 2003.

Rhyne, T.-M. (2000). Computer games' influence on scientific and information visualization, Computer, Vol. 33, No. 12, 154–159.

Rhyne, T.-M. (2002). Computer Games and Scientific Visualization. Communications of the Association for Computing Machinery (CACM), Vol. 45, No. 7, 40–44.

Ringbauer, B., Heidmann, F. & Biesterfeldt, J. (2003). When a House Controls its Master, Universal Design for Smart Living Environments. In: Jacko, J. & Stephanidis, C. (Eds.) Proceedings of 10th Int. Conf. on Human-Computer Interaction (HCI International 2003), Vol. 1, Heraklion, Greece, 22–27 June 2003.

Rocker, C., Streitz, N., Jance, M., & Portolan, N. (2005). User Requirements for intelligent Home Environments: A Scenario Driven Approach and Empirical Cross Cultural Study. Smart Objects and Ambient Intelligence, SOC-EUSAI, Grenoble, France, 12–14 October 2005.

Schilit, B., Adams, N. & Want, R. (1994). Context-aware computing applications. In: IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, California, USA, 8–9 December 1994.

Seaborne, A. (2004) RDQL - A Query Language for RDF, W3C Submission http://www.w3.org/Submission/RDQL/ (Referenced 13. April 2007).

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy of information visualizations, Proc. IEEE Symposium on Visual Languages '96, Los Alamitos, California, USA, 3–6 September 1996, 336–343.

Singh, I., Stearns, B. & Johnson, M. (2002). Designing Enterprise Applications with the J2EE Platform. Second edition, Boston: Addison-Wesley.

Sintek, M. (2003). Ontoviz tab: Visualizing Protégé ontologies. http://protege.cim3.net/cgi-bin/wiki.pl?OntoViz, (Referenced 2. April 2007).

Smallman, H., St. John, M., Oonk, H., & Cowen, M. (2001). Information Availability in 2D and 3D Displays. IEEE Computer Graphics and Applications, Vol. 21, No. 5, 51–57.

Spence, R. (2001). Information Visualization. Boston: Addison-Wesley.

Springmeyer, R.R., Blattner, M.M. & Marx, N.L. (1992). A Characterization of the Scientific Data Analysis Process. Proceedings of IEEE Visualization '92, Boston, Massachusetts, USA, 19–23 October 1992.

Staab, S., Erdmann, M., Mädche, A. & Decker, S. (2000.) An extensible approach for modeling ontologies in RDF(S). First Workshop on the Semantic Web at the Fourth European Conference on Digital Libraries, Lisbon, Portugal, 18–20 September 2000.

St. John, M., Cowen, M.B., Smallman, H.S & Oonk, H.M. (2001). The use of 2D and 3D displays for shape-understanding versus relative-position tasks. Human Factors, Vol. 43, No. 1, pp. 79–98.

Storey, M.A., Musen, M., Silva, J., Best, C., Ernst, N., Fergerson, R. & Noy, N. (2001). Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé. In: Workshop on Interactive Tools for Knowledge Capture, Victoria, B.C. Canada, 20 October 2001.

Tane, J., Schmitz, C. & Stumme, G. (2004). Semantic resource management for the web: an e-learning application. In: Proceedings of the 13th international World Wide Web Conference on Alternate Track Papers & Posters, New York, NY, USA, 19–21 May 2004.

Taniar, D. & Wenny, J. R. (2006). Web semantics and ontology, Hershey, PA, USA: Ideagroup Pub.

Tory, M., Kirkpatrick, A.E., Atkins M.S. & Möller, T. (2006). Visualization task performance with 2D, 3D, and combination displays. IEEE Transactions on Visualization and Computer Graphics, Vol. 12, No. 1, 2–13.

Tory, M., Möller, T., Atkins, M. S. & Kirkpatrick, A. E. (2004). Combining 2D and 3D views for orientation and relative position tasks. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vienna, Austria, 24–29 April 2004.

Uschoold, M. & Cruninger, M. (1996). Ontologies: Principles, Methods and Applications. Knowledge Engineering Review, Vol. 11, No. 2, 93–155.

Vallée, M., Ramparany, F. & Vercouter, L., (2005). Flexible composition of smart device services. In: The 2005 International Conference on Pervasive Systems and Computing, Las Vegas, Nevada, USA, 27–30 June 2005.

Venkatesh, A., Kruse, E. & Shih, E.C.-F. (2003). The Networked Home: An Analysis of Current Developments and Future Trends. Cognition, Technology and Work, Vol. 5, No. 1, 23–32.

Wanger, L.R., Ferwerda, J.A. & Greenberg, D.P. (1992). Perceiving spatial relationships in computer-generated images. IEEE Computer Graphics and Applications, Vol.12, No. 3, 44–58.

Weber, W., Braun, C., Glaser, R., Gsottberger, Y., Halik, M., Jung, S., Klauk, H., Lauterbach, C., Schmid, G., Shi, X., Sturm, T.F., Stromberg, G. & Zschieschang, U. (2003). Ambient intelligence – key technologies in the information age. In: 49th IEEE International Electron Devices Meeting, Washington, D.C, USA, 8–10 December 2003.

Wehrend, S. & Lewis, C. (1990). A problem-oriented classification of visualization techniques. Proceedings of the 1st conference on Visualization '90, San Francisco, California, USA, 23–26 October 1990.

Weiser, M. (1991). The computer for the 21st century. Scientific American, Vol. 265, No. 3, 94–104.

Wijk, J.J. (2005). The value of visualization. Proceedings of the IEEE Visualization (VIS'05), Minneapolis, MN, USA, 23–28 October 2005.

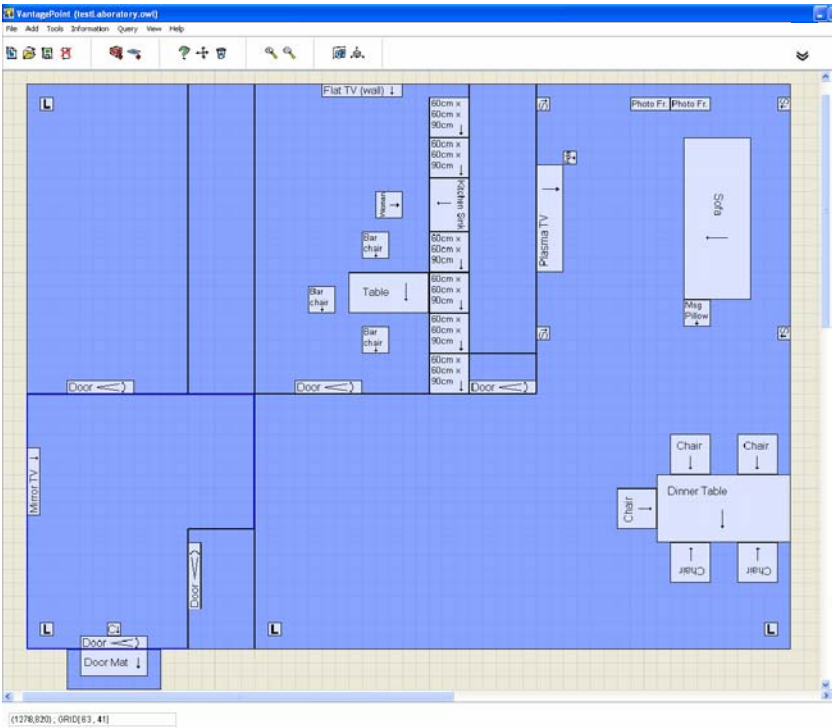
Zhang, Z. (2005). Ontology query languages for the semantic web: A performance evaluation. Master’s thesis, University of Georgia.

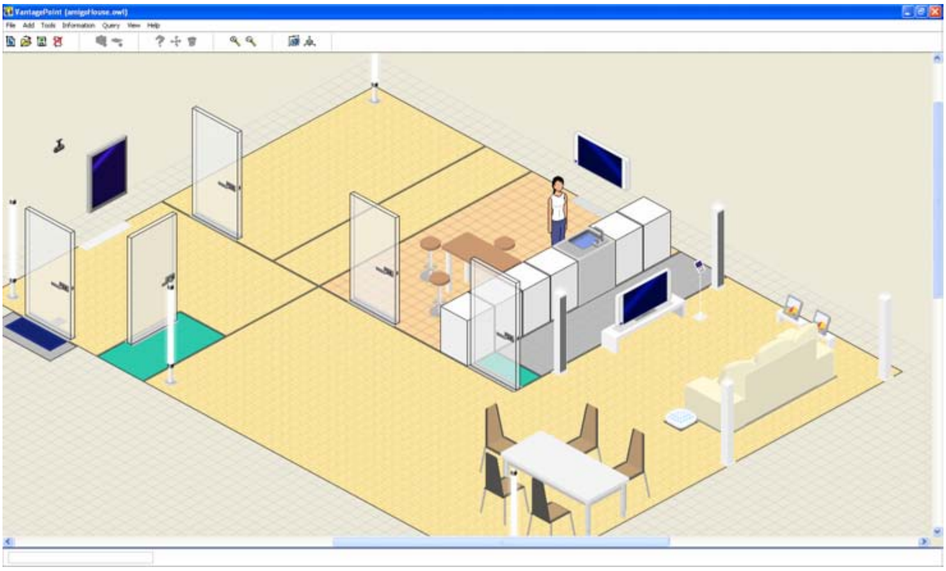
Appendix A: An example OWL-S service description

```
− <rdf:RDF xml:base="http://amigo.gforge.inria.fr/owl/BrandXPressureDetectionService.owl">
  − <owl:Ontology rdf:about="">
      <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Service.owl"/>
      <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Profile.owl"/>
      <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl"/>
      <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Grounding.owl"/>
    </owl:Ontology>
  − <!--

        ##############################################################
              Class definitions
        ##############################################################

    -->
  − <owl:Class rdf:ID="PressureDetectionProfile">
      <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Profile.owl#Profile"/>
      <rdfs:subClassOf rdf:resource="http://amigo.gforge.inria.fr/owl/Capabilities.owl#ContextDetection"/>
      <rdfs:subClassOf rdf:resource="http://amigo.gforge.inria.fr/owl/Capabilities.owl#HomeCareAndSafetySupport"/>
    </owl:Class>
  − <!--

        ##############################################################
              Service
        ##############################################################

    -->
    <!--  Service description  -->
  − <service:Service rdf:ID="brandXPressureDetectionService">
      <service:presents rdf:resource="#brandXPressureDetectionProfile"/>
      <service:describedBy rdf:resource="#pressureDetectionProcess"/>
    </service:Service>
    <!--  Profile description  -->
  − <PressureDetectionProfile rdf:ID="brandXPressureDetectionProfile">
      <service:presentedBy rdf:resource="#brandXPressureDetectionService"/>
      <profile:serviceName>Brand X Pressure Detection</profile:serviceName>
      <profile:textDescription>Senses pressure on target and calculates weight.</profile:textDescription>
    − <profile:contactInformation>
      − <rdf:Description rdf:about="http://brandX.com/staff/Yacob">
          <vCard:FN>Yacob Sato</vCard:FN>
        − <vCard:N rdf:parseType="Resource">
            <vCard:Family>Sato</vCard:Family>
            <vCard:Given>Yacob</vCard:Given>
          </vCard:N>
        − <vCard:EMAIL rdf:parseType="Resource">
            <rdf:value>MrY@brandX.com</rdf:value>
            <rdf:type rdf:resource="http://www.w3.org/2001/vcard-rdf/3.0#internet"/>
          </vCard:EMAIL>
        − <vCard:ORG rdf:parseType="Resource">
            <vCard:Orgname>BrandX ltd</vCard:Orgname>
            <vCard:Orgunit>Customer Relationships</vCard:Orgunit>
          </vCard:ORG>
        </rdf:Description>
      </profile:contactInformation>
      <profile:has_process rdf:resource="#pressureDetectionProcess"/>
    − <profile:serviceCategory>
      − <profile:ServiceCategory>
          <profile:categoryName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Capabilities.owl</profile:categoryName>
          <profile:taxonomy rdf:datatype="http://www.w3.org/2001/XMLSchema#string">http://amigo.gforge.inria.fr/owl/Capabilities.owl</profile:taxonomy>
          <profile:value rdf:resource="http://amigo.gforge.inria.fr/owl/Capabilities.owl#ContextDetection"/>
          <profile:value rdf:resource="http://amigo.gforge.inria.fr/owl/Capabilities.owl#HomeCareAndSafetySupport"/>
        </profile:ServiceCategory>
      </profile:serviceCategory>
    − <AmigoS:hasGlobalQoSParameter>
      − <AmigoS:ProvidedQoSParameter rdf:ID="brandXDoorMat2000AccuracyParameter">
        − <AmigoS:hasQoSType>
            <AmigoS:QoSType rdf:ID="accuracy"/>
          </AmigoS:hasQoSType>
          <AmigoS:Nature>static</AmigoS:Nature>
        − <AmigoS:hasMetric>
          − <AmigoS:Metric rdf:ID="brandXDoorMat2000Accuracy">
              <AmigoS:Value>0.5</AmigoS:Value>
              <AmigoS:MetricType>decimal</AmigoS:MetricType>
            − <AmigoS:hasUnit>
```

```
               <AmigoS:Unit rdf:ID="kilos"/>
             </AmigoS:hasUnit>
           </AmigoS:Metric>
         </AmigoS:hasMetric>
       </AmigoS:ProvidedQoSParameter>
     </AmigoS:hasGlobalQoSParameter>
   − <AmigoS:hasCapability>
     − <AmigoS:ProvidedCapability rdf:ID="pressureDetection">
       − <AmigoS:hasConversation>
         − <process:AtomicProcess rdf:ID="pressureDetectionProcess">
           − <process:hasInput>
            − <process:Input rdf:ID="pressureSensorValue">
                <process:parameterType/>
              </process:Input>
            </process:hasInput>
          − <process:hasOutput>
            − <process:Output rdf:ID="weightValue">
                <process:parameterType/>
              </process:Output>
            </process:hasOutput>
          </process:AtomicProcess>
        </AmigoS:hasConversation>
      </AmigoS:ProvidedCapability>
    </AmigoS:hasCapability>
  </PressureDetectionProfile>
</rdf:RDF>
```

Appendix B: The visualization and the OWL description of the environment used in validation





```
– <rdf:RDF>
  – <owl:Ontology rdf:about="http://www.owl-ontologies.com/MyHouse.owl">
      <owl:imports rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl"/>
    </owl:Ontology>
  – <rdf:Description rdf:about="http://www.owl-ontologies.com/MyHouse.owl#frontDoorMat">
      <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/Domotics.owl#MeasuringSensor
      </rdf:type>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
      </rdf:type>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
      </rdf:type>
      <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#brandXPressureDetectionService"/>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
      </rdf:type>
      <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:relativeX>
      <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/doorMat.png</WorldModel:iconURL>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/Domotics.owl#Sensor
      </rdf:type>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</WorldModel:relativeY>
      <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
      <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
      <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/Devices.owl#Device
      </rdf:type>
      <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
    – <WorldModel:isContainedBy>
      – <WorldModel:Area rdf:about="http://www.owl-ontologies.com/MyHouse.owl#porch">
          <WorldModel:contains rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#frontDoorMat"/>
          <WorldModel:texture rdf:datatype="http://www.w3.org/2001/XMLSchema#string">CEMENT</WorldModel:texture>
          <WorldModel:points rdf:datatype="http://www.w3.org/2001/XMLSchema#string">(0,0);(140,0);(140,60);(0,60)</WorldModel:points>
          <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</WorldModel:Id>
         <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">900</WorldModel:relativeY>
         <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">120</WorldModel:relativeX>
        </WorldModel:Area>
     </WorldModel:isContainedBy>
     <rdf:type rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#PressureSensor"/>
   – <rdf:type>
       http://www.owl-ontologies.com/DeviceLibrary.owl#PressureSensor
     </rdf:type>
   – <rdf:type>
       http://amigo.gforge.inria.fr/owl/Domotics.owl#DomoticDevice
     </rdf:type>
     <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
   – <WorldModel:represents>
       http://www.owl-ontologies.com/DeviceLibrary.owl#brandXDoorMat2000
     </WorldModel:represents>
     <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
   </rdf:Description>
 – <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#kitchenDesk3">
     <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">260</WorldModel:relativeX>
     <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">280</WorldModel:relativeY>
     <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
     <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/60x60x90Box.png</WorldModel:iconURL>
     <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
   – <WorldModel:isContainedBy>
     – <WorldModel:Area rdf:about="http://www.owl-ontologies.com/MyHouse.owl#Kitchen">
       – <WorldModel:contains>
         – <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#barChair2">
             <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">360</WorldModel:relativeY>
             <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
             <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/barChair.png</WorldModel:iconURL>
```

```
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">160</WorldModel:relativeX>
        <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
      </WorldModel:Item>
    </WorldModel:contains>
  − <WorldModel:contains>
    − <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#kitchenDesk5">
        <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">260</WorldModel:relativeX>
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/60x60x90Box.png</WorldModel:iconURL>
        <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
        <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">80</WorldModel:relativeY>
      </WorldModel:Item>
    </WorldModel:contains>
    <WorldModel:contains rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#kitchenDesk3"/>
    <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">400</WorldModel:relativeX>
  − <WorldModel:contains>
    − <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#kitchenDesk4">
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">260</WorldModel:relativeX>
        <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">220</WorldModel:relativeY>
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/60x60x90Box.png</WorldModel:iconURL>
        <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
        <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
      </WorldModel:Item>
    </WorldModel:contains>
  − <WorldModel:contains>
    − <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#kitchenDesk2">
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">260</WorldModel:relativeX>
        <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">340</WorldModel:relativeY>
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/60x60x90Box.png</WorldModel:iconURL>
        <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
        <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
      </WorldModel:Item>
    </WorldModel:contains>
  − <WorldModel:contains>
    − <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#kitchenSink">
        <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
        <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">140</WorldModel:relativeY>
        <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/kitchenSink.png</WorldModel:iconURL>
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">260</WorldModel:relativeX>
        <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">WEST</WorldModel:direction>
      </WorldModel:Item>
    </WorldModel:contains>
    <WorldModel:texture rdf:datatype="http://www.w3.org/2001/XMLSchema#string">CERAMIC</WorldModel:texture>
    <WorldModel:points rdf:datatype="http://www.w3.org/2001/XMLSchema#string">(0,0);(320,0);(320,460);(0,460)</WorldModel:points>
  − <WorldModel:contains>
    − <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#kitchenTable">
        <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">280</WorldModel:relativeY>
        <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
        <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/table.png</WorldModel:iconURL>
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">140</WorldModel:relativeX>
        <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
      </WorldModel:Item>
    </WorldModel:contains>
    <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">60</WorldModel:relativeY>
  − <WorldModel:contains>
    − <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#kitchenDoor">
        <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/door1.png</WorldModel:iconURL>
        <WorldModel:height rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</WorldModel:height>
        <WorldModel:width rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</WorldModel:width>
        <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
        <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">440</WorldModel:relativeY>
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">60</WorldModel:relativeX>
        <WorldModel:depth rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:depth>
```

```
      </WorldModel:Item>
    </WorldModel:contains>
  - <WorldModel:contains>
    - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#kitchenDesk6">
        <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">260</WorldModel:relativeX>
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/60x60x90Box.png</WorldModel:iconURL>
        <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
        <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:relativeY>
      </WorldModel:Item>
    </WorldModel:contains>
  - <WorldModel:contains>
    - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#tyjt">
        <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">EAST</WorldModel:direction>
        <j.3:hasUserProfile rdf:resource="http://www.owl-ontologies.com/PersonLibrary.owl#MariaUser"/>
        <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      - <rdf:type>
          http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
        </rdf:type>
        <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
        <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
        <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">160</WorldModel:relativeY>
        <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/Amigo.owl#Person"/>
        <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Person</rdf:type>
      - <WorldModel:represents>
          http://www.owl-ontologies.com/PersonLibrary.owl#MariaPerson
        </WorldModel:represents>
      - <rdf:type>
          http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
        </rdf:type>
        <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/womanFrame.png</WorldModel:iconURL>
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">180</WorldModel:relativeX>
        <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
      </WorldModel:Item>
    </WorldModel:contains>
  - <WorldModel:contains>
    - <rdf:Description rdf:about="http://www.owl-ontologies.com/MyHouse.owl#KitchenWallTV">
        <rdf:type>-19a2404d:1105395f2b6:-7fad</rdf:type>
        <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
      - <WorldModel:represents>
          http://www.owl-ontologies.com/DeviceLibrary.owl#brandXWallTV
        </WorldModel:represents>
      - <rdf:type>
          http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#VideoDevice
        </rdf:type>
        <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
      - <rdf:type>
          http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#ConsumerElectronicsDevice
        </rdf:type>
        <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
      - <rdf:type>
          http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
        </rdf:type>
        <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</WorldModel:relativeX>
        <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</WorldModel:relativeY>
        <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
        <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/flatTVWall.png</WorldModel:iconURL>
        <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
      - <rdf:type>
          http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
        </rdf:type>
        <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#VideoRenderDevice"/>
      - <rdf:type>
          http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#VideoRenderDevice
        </rdf:type>
        <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
```

```
      <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
      <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
      <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
      <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#deviceStateControlService"/>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
      </rdf:type>
      <rdf:type>-19a2404d:1105395f2b6:-7faa</rdf:type>
    </rdf:Description>
  </WorldModel:contains>
- <WorldModel:contains>
  - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#kitchenDesk1">
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">260</WorldModel:relativeX>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">400</WorldModel:relativeY>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/60x60x90Box.png</WorldModel:iconURL>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
    </WorldModel:Item>
  </WorldModel:contains>
- <WorldModel:contains>
  - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#barChair3">
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/barChair.png</WorldModel:iconURL>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">300</WorldModel:relativeY>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">80</WorldModel:relativeX>
    </WorldModel:Item>
  </WorldModel:contains>
  <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</WorldModel:Id>
- <WorldModel:contains>
  - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#barChair1">
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">220</WorldModel:relativeY>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/barChair.png</WorldModel:iconURL>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">160</WorldModel:relativeX>
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
    </WorldModel:Item>
      </WorldModel:contains>
    - <WorldModel:contains>
      - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#barChair3">
          <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
          <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
          <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/barChair.png</WorldModel:iconURL>
          <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
          <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">300</WorldModel:relativeY>
          <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">80</WorldModel:relativeX>
        </WorldModel:Item>
      </WorldModel:contains>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</WorldModel:Id>
    - <WorldModel:contains>
      - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#barChair1">
          <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">220</WorldModel:relativeY>
          <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
          <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/barChair.png</WorldModel:iconURL>
          <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
          <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">160</WorldModel:relativeX>
          <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Kitchen"/>
        </WorldModel:Item>
      </WorldModel:contains>
    </WorldModel:Area>
  </WorldModel:isContainedBy>
/WorldModel:Item>
rdf:Description rdf:about="http://www.owl-ontologies.com/MyHouse.owl#LivingRoomPlasmaTV">
  <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">120</WorldModel:relativeY>
  <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">420</WorldModel:relativeX>
  <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#deviceStateControlService"/>
- <rdf:type>
    http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
```

```
  </rdf:type>
  <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">EAST</WorldModel:direction>
  <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
  <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
- <rdf:type>
    http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
  </rdf:type>
- <WorldModel:represents>
    http://www.owl-ontologies.com/DeviceLibrary.owl#BrandXPlasmaTV
  </WorldModel:represents>
  <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
  <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/plasmaTable.png</WorldModel:iconURL>
  <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
- <WorldModel:isContainedBy>
  - <WorldModel:Area rdf:about="http://www.owl-ontologies.com/MyHouse.owl#Livingroom">
    - <WorldModel:contains>
      - <j.1:StillImageRenderDevice rdf:about="http://www.owl-ontologies.com/MyHouse.owl#photoFrame1">
        - <WorldModel:represents>
            http://www.owl-ontologies.com/DeviceLibrary.owl#PhotoFrame
          </WorldModel:represents>
          <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
          <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">560</WorldModel:relativeX>
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#StillImageRenderDevice
          </rdf:type>
          <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
          <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
          </rdf:type>
          <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:relativeY>
          <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/photoFrame.png</WorldModel:iconURL>
          <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
          <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
          <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
          <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
          </rdf:type>
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#StillImageDevice
          </rdf:type>
          <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
          <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
          <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#ConsumerElectronicsDevice
          </rdf:type>
          <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#deviceStateControlService"/>
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
          </rdf:type>
          <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        </j.1:StillImageRenderDevice>
      </WorldModel:contains>
    - <WorldModel:contains>
      - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#dinnerChair3">
          <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">EAST</WorldModel:direction>
          <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">540</WorldModel:relativeX>
          <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/chair.png</WorldModel:iconURL>
          <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
          <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">600</WorldModel:relativeY>
          <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
        </WorldModel:Item>
      </WorldModel:contains>
      <WorldModel:texture rdf:datatype="http://www.w3.org/2001/XMLSchema#string">BIRCH_PARQUET</WorldModel:texture>
    - <WorldModel:contains>
      - <rdf:Description rdf:about="http://www.owl-ontologies.com/MyHouse.owl#gameController">
          <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
          <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
          <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/pda.png</WorldModel:iconURL>
```

```
    <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
    <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">460</WorldModel:relativeX>
    <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
    <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
  − <rdf:type>
      http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
    </rdf:type>
    <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</WorldModel:relativeY>
  − <WorldModel:represents>
      http://www.owl-ontologies.com/DeviceLibrary.owl#brandXComputer
    </WorldModel:represents>
    <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
    <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
    <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/Devices.owl#DesktopPC"/>
  − <rdf:type>
      http://amigo.gforge.inria.fr/owl/Devices.owl#DesktopPC
    </rdf:type>
    <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
  − <rdf:type>
      http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
    </rdf:type>
  − <rdf:type>
      http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
    </rdf:type>
    <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#brandXMediaManagerService"/>
    <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">EAST</WorldModel:direction>
  − <rdf:type>
      http://amigo.gforge.inria.fr/owl/Devices.owl#ProcessingUnit
    </rdf:type>
  − <rdf:type>
      http://amigo.gforge.inria.fr/owl/Devices.owl#Device
    </rdf:type>
    <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
    <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
  </rdf:Description>
</WorldModel:contains>
<WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">400</WorldModel:relativeX>
− <WorldModel:contains>
  − <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#dinnerChair4">
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">NORTH</WorldModel:direction>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">620</WorldModel:relativeX>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/chair.png</WorldModel:iconURL>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">680</WorldModel:relativeY>
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
    </WorldModel:Item>
  </WorldModel:contains>
− <WorldModel:contains>
  − <rdf:Description rdf:about="http://www.owl-ontologies.com/MyHouse.owl#leftRearSpeaker">
      <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#deviceStateControlService"/>
      <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
      <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
      <rdf:type>-19a2404d:1105395f2b6:-7fad</rdf:type>
    − <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#ConsumerElectronicsDevice
      </rdf:type>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">780</WorldModel:relativeX>
    − <WorldModel:represents>
        http://www.owl-ontologies.com/DeviceLibrary.owl#brandXSpeaker2000
      </WorldModel:represents>
    − <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
      </rdf:type>
    − <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
      </rdf:type>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
      <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
      <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
      <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
```

```
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">360</WorldModel:relativeY>
      <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">WEST</WorldModel:direction>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#AudioRenderDevice
      </rdf:type>
      <rdf:type>-19a2404d:1105395f2b6:-7faa</rdf:type>
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#AudioDevice
      </rdf:type>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/speakerSmall.png</WorldModel:iconURL>
      <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#Speaker"/>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#Speaker
      </rdf:type>
      <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
      </rdf:type>
    </rdf:Description>
  </WorldModel:contains>
- <WorldModel:contains>
  - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#dinnerTable">
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">600</WorldModel:relativeX>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/dinnerTable.png</WorldModel:iconURL>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">580</WorldModel:relativeY>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
    </WorldModel:Item>
  </WorldModel:contains>
- <WorldModel:points rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    (0,460);(0,840);(800,840);(800,0);(420,0);(420,460)
  </WorldModel:points>
  <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">60</WorldModel:relativeY>
- <WorldModel:contains>
  - <Amigo:Device rdf:about="http://www.owl-ontologies.com/MyHouse.owl#lamp1">
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">800</WorldModel:relativeY>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
    - <WorldModel:represents>
        http://www.owl-ontologies.com/DeviceLibrary.owl#Lamp
      </WorldModel:represents>
    - <Amigo:deploysService>
        http://www.owl-ontologies.com/DeviceLibrary.owl#illuminationService
      </Amigo:deploysService>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">760</WorldModel:relativeX>
      <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/pilarLamp.png</WorldModel:iconURL>
    </Amigo:Device>
  </WorldModel:contains>
- <WorldModel:contains>
  - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#dinnerChair5">
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">NORTH</WorldModel:direction>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">720</WorldModel:relativeX>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/chair.png</WorldModel:iconURL>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">680</WorldModel:relativeY>
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
    </WorldModel:Item>
  </WorldModel:contains>
- <WorldModel:contains>
  - <rdf:Description rdf:about="http://www.owl-ontologies.com/MyHouse.owl#smsPillow">
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/Domotics.owl#MeasuringSensor
      </rdf:type>
      <rdf:type rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#PressureSensor"/>
    - <rdf:type>
        http://www.owl-ontologies.com/DeviceLibrary.owl#PressureSensor
```

```
      </rdf:type>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/Domotics.owl#Sensor
      </rdf:type>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/Domotics.owl#DomoticDevice
      </rdf:type>
      <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
      <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#brandXSMSPillowService"/>
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/messagePillow.png</WorldModel:iconURL>
      <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/Devices.owl#Device
      </rdf:type>
      <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
      <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">640</WorldModel:relativeX>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">320</WorldModel:relativeY>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
      </rdf:type>
    – <WorldModel:represents>
        http://www.owl-ontologies.com/DeviceLibrary.owl#brandXMessagePillow2000
      </WorldModel:represents>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
      </rdf:type>
      <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
      <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
      </rdf:type>
      <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
      <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
    </rdf:Description>
  </WorldModel:contains>
  <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</WorldModel:Id>
– <WorldModel:contains>
  – <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#sofa">
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
      <WorldModel:width rdf:datatype="http://www.w3.org/2001/XMLSchema#int">240</WorldModel:width>
      <WorldModel:height rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</WorldModel:height>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">80</WorldModel:relativeY>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">WEST</WorldModel:direction>
      <WorldModel:depth rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</WorldModel:depth>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">640</WorldModel:relativeX>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/sofa.png</WorldModel:iconURL>
    </WorldModel:Item>
  </WorldModel:contains>
  <WorldModel:contains rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#LivingRoomPlasmaTV"/>
– <WorldModel:contains>
  – <rdf:Description rdf:about="http://www.owl-ontologies.com/MyHouse.owl#rightFrontSpeaker">
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">EAST</WorldModel:direction>
      <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#AudioRenderDevice
      </rdf:type>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#AudioDevice
      </rdf:type>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#ConsumerElectronicsDevice
      </rdf:type>
      <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#Speaker"/>
    – <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#Speaker
      </rdf:type>
```

```
      <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
      <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
      <rdf:type>-19a2404d:1105395f2b6:-7faa</rdf:type>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
      </rdf:type>
      <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
      <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
      <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#deviceStateControlService"/>
      <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
      </rdf:type>
      <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">420</WorldModel:relativeX>
    - <WorldModel:represents>
        http://www.owl-ontologies.com/DeviceLibrary.owl#brandXSpeaker2000
      </WorldModel:represents>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:relativeY>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
      </rdf:type>
      <rdf:type>-19a2404d:1105395f2b6:-7fad</rdf:type>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/speakerSmall.png</WorldModel:iconURL>
    </rdf:Description>
  </WorldModel:contains>
- <WorldModel:contains>
  - <rdf:Description rdf:about="http://www.owl-ontologies.com/MyHouse.owl#rightRearSpeaker">
    - <WorldModel:represents>
        http://www.owl-ontologies.com/DeviceLibrary.owl#brandXSpeaker2000
      </WorldModel:represents>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/speakerSmall.png</WorldModel:iconURL>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">780</WorldModel:relativeX>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
      </rdf:type>
      <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#AudioRenderDevice
      </rdf:type>
      <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
      </rdf:type>
      <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
      <rdf:type>-19a2404d:1105395f2b6:-7fad</rdf:type>
      <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#Speaker"/>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#Speaker
      </rdf:type>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">WEST</WorldModel:direction>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#ConsumerElectronicsDevice
      </rdf:type>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:relativeY>
      <rdf:type>-19a2404d:1105395f2b6:-7faa</rdf:type>
      <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#deviceStateControlService"/>
      <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#AudioDevice
      </rdf:type>
      <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
      <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
      <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
      </rdf:type>
```

```
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
      <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
    </rdf:Description>
  </WorldModel:contains>
- <WorldModel:contains>
  - <Amigo:Device rdf:about="http://www.owl-ontologies.com/MyHouse.owl#lamp2">
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/pilarLamp.png</WorldModel:iconURL>
    - <Amigo:deploysService>
        http://www.owl-ontologies.com/DeviceLibrary.owl#illuminationService
      </Amigo:deploysService>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">800</WorldModel:relativeY>
    - <WorldModel:represents>
        http://www.owl-ontologies.com/DeviceLibrary.owl#Lamp
      </WorldModel:represents>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
      <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:relativeX>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
    </Amigo:Device>
  </WorldModel:contains>
- <WorldModel:contains>
  - <rdf:Description rdf:about="http://www.owl-ontologies.com/MyHouse.owl#leftFrontSpeaker">
      <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
      </rdf:type>
      <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
      <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/speakerSmall.png</WorldModel:iconURL>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#ConsumerElectronicsDevice
      </rdf:type>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
      </rdf:type>
      <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#Speaker"/>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#Speaker
      </rdf:type>
      <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">360</WorldModel:relativeY>
      <rdf:type>-19a2404d:1105395f2b6:-7faa</rdf:type>
    - <WorldModel:represents>
        http://www.owl-ontologies.com/DeviceLibrary.owl#brandXSpeaker2000
      </WorldModel:represents>
      <rdf:type>-19a2404d:1105395f2b6:-7fad</rdf:type>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">EAST</WorldModel:direction>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
      </rdf:type>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
      <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#deviceStateControlService"/>
      <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#AudioRenderDevice
      </rdf:type>
      <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
      <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
      <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
      <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
      <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
    - <rdf:type>
        http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#AudioDevice
      </rdf:type>
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">420</WorldModel:relativeX>
    </rdf:Description>
  </WorldModel:contains>
- <WorldModel:contains>
  - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#dinnerChair1">
      <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">720</WorldModel:relativeX>
      <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
```

```
          <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/chair.png</WorldModel:iconURL>
          <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
          <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">520</WorldModel:relativeY>
          <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
        </WorldModel:Item>
      </WorldModel:contains>
    - <WorldModel:contains>
      - <rdf:Description rdf:about="http://www.owl-ontologies.com/MyHouse.owl#photoFrame2">
          <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
          </rdf:type>
          <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
          <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
          <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
          <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/photoFrame.png</WorldModel:iconURL>
          <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
          </rdf:type>
          <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
          </rdf:type>
          <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
          <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:relativeY>
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#StillImageDevice
          </rdf:type>
          <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
          <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#ConsumerElectronicsDevice
          </rdf:type>
          <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">620</WorldModel:relativeX>
          <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#deviceStateControlService"/>
        - <WorldModel:represents>
            http://www.owl-ontologies.com/DeviceLibrary.owl#PhotoFrame
          </WorldModel:represents>
          <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#StillImageRenderDevice"/>
          - <rdf:type>
              http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#StillImageRenderDevice
            </rdf:type>
            <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
            <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
          </rdf:Description>
        </WorldModel:contains>
      - <WorldModel:contains>
        - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#dinnerChair2">
            <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">620</WorldModel:relativeX>
            <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
            <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/chair.png</WorldModel:iconURL>
            <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
            <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">520</WorldModel:relativeY>
            <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Livingroom"/>
          </WorldModel:Item>
        </WorldModel:contains>
      </WorldModel:Area>
    </WorldModel:isContainedBy>
    <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
  - <rdf:type>
      http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#ConsumerElectronicsDevice
    </rdf:type>
    <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
    <rdf:type>-19a2404d:1105395f2b6:-7faa</rdf:type>
  - <rdf:type>
      http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
    </rdf:type>
  - <rdf:type>
      http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#VideoDevice
    </rdf:type>
```

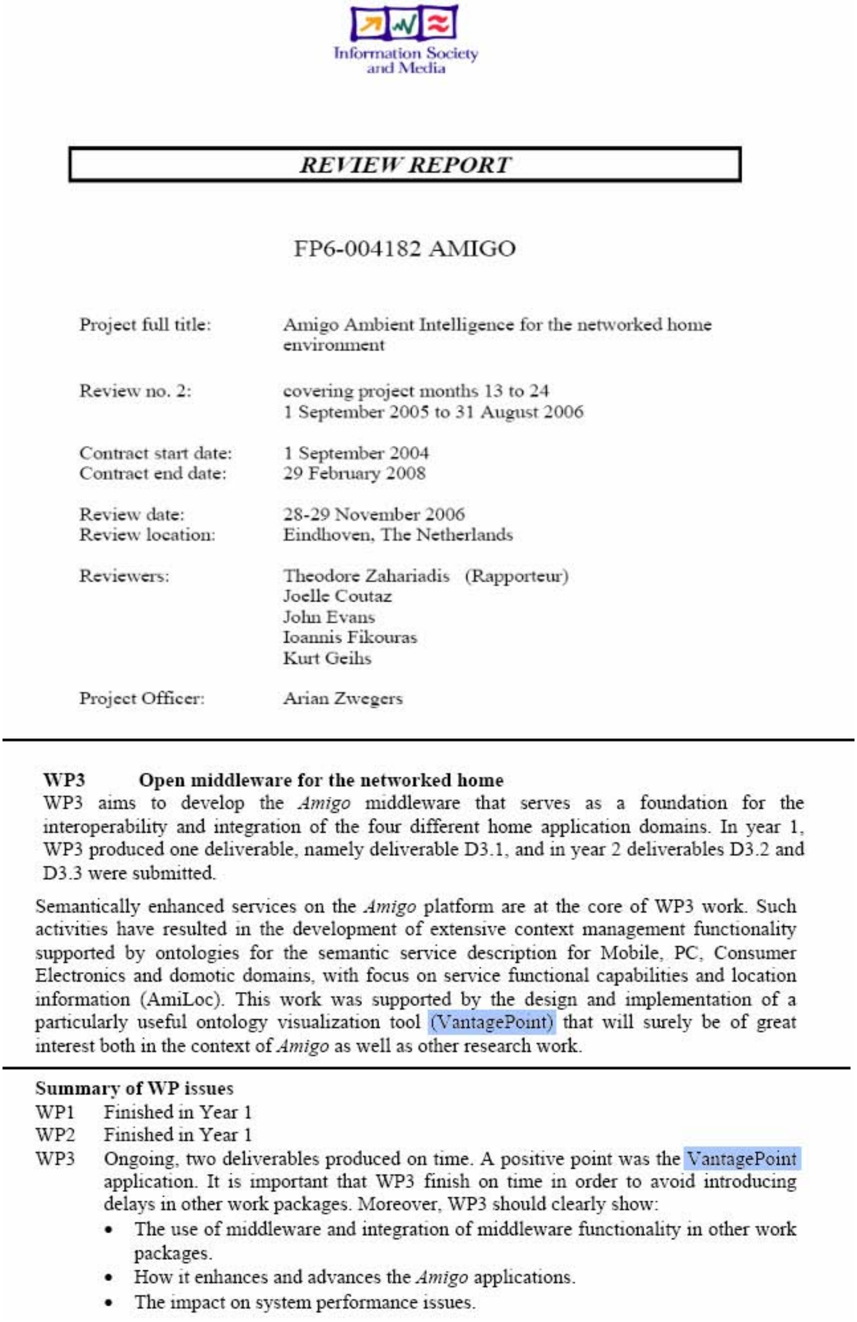
```
  – <rdf:type>
      http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#VideoRenderDevice
    </rdf:type>
    <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#VideoRenderDevice"/>
    <rdf:type>-19a2404d:1105395f2b6:-7fad</rdf:type>
    <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
    <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
    <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
  </rdf:Description>
- <WorldModel:Area rdf:about="http://www.owl-ontologies.com/MyHouse.owl#Stairway">
    <WorldModel:texture rdf:datatype="http://www.w3.org/2001/XMLSchema#string">BIRCH_PARQUET</WorldModel:texture>
    <WorldModel:points rdf:datatype="http://www.w3.org/2001/XMLSchema#string">(0,0);(100,0);(100,460);(0,460)</WorldModel:points>
    <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</WorldModel:Id>
    <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">60</WorldModel:relativeY>
    <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">300</WorldModel:relativeX>
  </WorldModel:Area>
- <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#door2">
    <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
  – <WorldModel:isContainedBy>
    – <WorldModel:Area rdf:about="http://www.owl-ontologies.com/MyHouse.owl#Room1">
        <WorldModel:texture rdf:datatype="http://www.w3.org/2001/XMLSchema#string">BIRCH_PARQUET</WorldModel:texture>
        <WorldModel:points rdf:datatype="http://www.w3.org/2001/XMLSchema#string">(0,460);(0,0);(240,0);(240,460)</WorldModel:points>
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</WorldModel:Id>
        <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">60</WorldModel:relativeY>
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">60</WorldModel:relativeX>
        <WorldModel:contains rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#door2"/>
      – <WorldModel:contains>
        – <Amigo:Device rdf:about="http://www.owl-ontologies.com/MyHouse.owl#lamp4">
          – <WorldModel:represents>
              http://www.owl-ontologies.com/DeviceLibrary.owl#Lamp
            </WorldModel:represents>
            <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:relativeY>
            <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
            <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:relativeX>
            <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Room1"/>
          – <Amigo:deploysService>
              http://www.owl-ontologies.com/DeviceLibrary.owl#illuminationService
            </Amigo:deploysService>
            <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
            <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/pilarLamp.png</WorldModel:iconURL>
            <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
          </Amigo:Device>
        </WorldModel:contains>
      </WorldModel:Area>
    </WorldModel:isContainedBy>
    <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">60</WorldModel:relativeX>
    <WorldModel:width rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</WorldModel:width>
    <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
    <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">440</WorldModel:relativeY>
    <WorldModel:depth rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:depth>
    <WorldModel:height rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</WorldModel:height>
    <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/door1.png</WorldModel:iconURL>
  </WorldModel:Item>
– <WorldModel:Area rdf:about="http://www.owl-ontologies.com/MyHouse.owl#Room4">
    <WorldModel:texture rdf:datatype="http://www.w3.org/2001/XMLSchema#string">CEMENT</WorldModel:texture>
    <WorldModel:points rdf:datatype="http://www.w3.org/2001/XMLSchema#string">(0,0);(100,0);(100,400);(0,400)</WorldModel:points>
    <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</WorldModel:Id>
    <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">60</WorldModel:relativeY>
    <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">720</WorldModel:relativeX>
  </WorldModel:Area>
– <rdf:Description rdf:about="http://www.owl-ontologies.com/MyHouse.owl#mirrorTV">
    <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/mirrorTV.png</WorldModel:iconURL>
  – <rdf:type>
      http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
    </rdf:type>
    <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
    <rdf:type>-19a2404d:1105395f2b6:-7faa</rdf:type>
    <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#VideoRenderDevice"/>
  – <rdf:type>
      http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#VideoRenderDevice
    </rdf:type>
```

```
    <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
  – <WorldModel:represents>
      http://www.owl-ontologies.com/DeviceLibrary.owl#brandXMirrorTV
    </WorldModel:represents>
  – <rdf:type>
      http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#ConsumerElectronicsDevice
    </rdf:type>
    <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">80</WorldModel:relativeY>
    <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">EAST</WorldModel:direction>
    <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
  – <rdf:type>
      http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#VideoDevice
    </rdf:type>
  – <WorldModel:isContainedBy>
    – <WorldModel:Area rdf:about="http://www.owl-ontologies.com/MyHouse.owl#Hall">
      – <WorldModel:contains>
        – <rdf:Description rdf:about="http://www.owl-ontologies.com/MyHouse.owl#frontDoorCamera">
            <rdf:type>http://www.w3.org/2000/01/rdf-schema#Resource</rdf:type>
            <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">120</WorldModel:relativeX>
            <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
            <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
            <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
            <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Hall"/>
          – <rdf:type>
              http://amigo.gforge.inria.fr/owl/Domotics.owl#PrecenseDetector
            </rdf:type>
          – <rdf:type>
              http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#VideoCaptureDevice
            </rdf:type>
            <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#brandXVideoStreamingService"/>
            <rdf:type rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#DigitalSecurityCamera"/>
          – <rdf:type>
              http://www.owl-ontologies.com/DeviceLibrary.owl#DigitalSecurityCamera
            </rdf:type>
            <rdf:type>-19a2404d:1105395f2b6:-7fad</rdf:type>
          – <rdf:type>
              http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
            </rdf:type>
          – <rdf:type>
              http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#ConsumerElectronicsDevice
            </rdf:type>
          – <rdf:type>
              http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#VideoDevice
            </rdf:type>
          – <rdf:type>
              http://amigo.gforge.inria.fr/owl/Devices.owl#Device
            </rdf:type>
          – <rdf:type>
              http://amigo.gforge.inria.fr/owl/Domotics.owl#DomoticDevice
            </rdf:type>
          – <rdf:type>
              http://amigo.gforge.inria.fr/owl/Amigo.owl#PhysicalEntity
            </rdf:type>
          – <WorldModel:represents>
              http://www.owl-ontologies.com/DeviceLibrary.owl#brandXDigitalSecurityCamera2000
            </WorldModel:represents>
            <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
            <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
            <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
            <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
            <rdf:type>-19a2404d:1105395f2b6:-7fa6</rdf:type>
          – <rdf:type>
              http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
            </rdf:type>
          – <rdf:type>
              http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#StillImageCaptureDevice
            </rdf:type>
          – <rdf:type>
              http://amigo.gforge.inria.fr/owl/Domotics.owl#Sensor
            </rdf:type>
            <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">340</WorldModel:relativeY>
```

```
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/Domotics.owl#Detector
          </rdf:type>
          <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/camera.png</WorldModel:iconURL>
          <rdf:type>-19a2404d:1105395f2b6:-7faa</rdf:type>
          <rdf:type>-19a2404d:1105395f2b6:-7fa2</rdf:type>
        - <rdf:type>
            http://amigo.gforge.inria.fr/owl/ConsumerElectronics.owl#StillImageDevice
          </rdf:type>
        </rdf:Description>
      </WorldModel:contains>
    - <WorldModel:points rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        (0,0);(0,380);(240,380);(240,200);(340,200);(340,0)
      </WorldModel:points>
      <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">520</WorldModel:relativeY>
      <WorldModel:texture rdf:datatype="http://www.w3.org/2001/XMLSchema#string">BIRCH_PARQUET</WorldModel:texture>
      <WorldModel:contains rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#mirrorTV"/>
      <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</WorldModel:Id>
    - <WorldModel:contains>
      - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#frontDoor">
          <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">80</WorldModel:relativeX>
          <WorldModel:height rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</WorldModel:height>
          <WorldModel:depth rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:depth>
          <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
          <WorldModel:width rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</WorldModel:width>
          <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
          <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/door1.png</WorldModel:iconURL>
          <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">360</WorldModel:relativeY>
          <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Hall"/>
        </WorldModel:Item>
      </WorldModel:contains>
    - <WorldModel:contains>
      - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#lamp3">
          <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Hall"/>
          <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
          <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:relativeX>
          <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/pilarLamp.png</WorldModel:iconURL>
          <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
          <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/Amigo.owl#Device"/>
        - <WorldModel:represents>
            http://www.owl-ontologies.com/DeviceLibrary.owl#Lamp
           </WorldModel:represents>
           <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">340</WorldModel:relativeY>
         - <Amigo:deploysService>
              http://www.owl-ontologies.com/DeviceLibrary.owl#illuminationService
           </Amigo:deploysService>
         </WorldModel:Item>
       </WorldModel:contains>
       <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">60</WorldModel:relativeX>
     </WorldModel:Area>
   </WorldModel:isContainedBy>
 - <rdf:type>
     http://amigo.gforge.inria.fr/owl/Amigo.owl#ContextConcept
   </rdf:type>
   <rdf:type>-19a2404d:1105395f2b6:-7fad</rdf:type>
   <rdf:type rdf:resource="http://amigo.gforge.inria.fr/owl/WorldModel.owl#Item"/>
   <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
   <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</WorldModel:relativeX>
   <rdf:type>http://www.w3.org/2002/07/owl#Thing</rdf:type>
   <rdf:type>http://www.owl-ontologies.com/WorldModel.owl#Item</rdf:type>
 - <rdf:type>
     http://amigo.gforge.inria.fr/owl/Amigo.owl#AmigoConcept
   </rdf:type>
   <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Object</rdf:type>
   <Amigo:deploysService rdf:resource="http://www.owl-ontologies.com/DeviceLibrary.owl#deviceStateControlService"/>
   <rdf:type>http://amigo.gforge.inria.fr/owl/Amigo.owl#Device</rdf:type>
 </rdf:Description>
- <WorldModel:Area rdf:about="http://www.owl-ontologies.com/MyHouse.owl#Room3">
   <WorldModel:texture rdf:datatype="http://www.w3.org/2001/XMLSchema#string">DEFAULT</WorldModel:texture>
   <WorldModel:points rdf:datatype="http://www.w3.org/2001/XMLSchema#string">(0,0);(100,0);(100,60);(0,60)</WorldModel:points>
   <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</WorldModel:Id>
```

```
    <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">460</WorldModel:relativeY>
    <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">720</WorldModel:relativeX>
  - <WorldModel:contains>
    - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#door3">
        <WorldModel:width rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</WorldModel:width>
        <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">SOUTH</WorldModel:direction>
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/door1.png</WorldModel:iconURL>
        <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Room3"/>
        <WorldModel:depth rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:depth>
        <WorldModel:height rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</WorldModel:height>
        <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">40</WorldModel:relativeY>
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</WorldModel:relativeX>
      </WorldModel:Item>
    </WorldModel:contains>
  </WorldModel:Area>
- <WorldModel:Area rdf:about="http://www.owl-ontologies.com/MyHouse.owl#Room2">
    <WorldModel:texture rdf:datatype="http://www.w3.org/2001/XMLSchema#string">DEFAULT</WorldModel:texture>
    <WorldModel:points rdf:datatype="http://www.w3.org/2001/XMLSchema#string">(0,180);(100,180);(100,0);(0,0)</WorldModel:points>
    <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</WorldModel:Id>
    <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">720</WorldModel:relativeY>
    <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">300</WorldModel:relativeX>
  - <WorldModel:contains>
    - <WorldModel:Item rdf:about="http://www.owl-ontologies.com/MyHouse.owl#door1">
        <WorldModel:isContainedBy rdf:resource="http://www.owl-ontologies.com/MyHouse.owl#Room2"/>
        <WorldModel:relativeX rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</WorldModel:relativeX>
        <WorldModel:depth rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:depth>
        <WorldModel:height rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</WorldModel:height>
        <WorldModel:width rdf:datatype="http://www.w3.org/2001/XMLSchema#int">100</WorldModel:width>
        <WorldModel:Id rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</WorldModel:Id>
        <WorldModel:direction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">EAST</WorldModel:direction>
        <WorldModel:relativeY rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20</WorldModel:relativeY>
        <WorldModel:iconURL rdf:datatype="http://www.w3.org/2001/XMLSchema#string">icons/door1.png</WorldModel:iconURL>
      </WorldModel:Item>
    </WorldModel:contains>
  </WorldModel:Area>
</rdf:RDF>
```

Appendix C: Extracts from the review report



Information Society
and Media

REVIEW REPORT

FP6-004182 AMIGO

Project full title:	Amigo Ambient Intelligence for the networked home environment
Review no. 2:	covering project months 13 to 24 1 September 2005 to 31 August 2006
Contract start date:	1 September 2004
Contract end date:	29 February 2008
Review date:	28-29 November 2006
Review location:	Eindhoven, The Netherlands
Reviewers:	Theodore Zahariadis (Rapporteur) Joelle Coutaz John Evans Ioannis Fikouras Kurt Geihs
Project Officer:	Arian Zwegers

WP3 Open middleware for the networked home

WP3 aims to develop the *Amigo* middleware that serves as a foundation for the interoperability and integration of the four different home application domains. In year 1, WP3 produced one deliverable, namely deliverable D3.1, and in year 2 deliverables D3.2 and D3.3 were submitted.

Semantically enhanced services on the *Amigo* platform are at the core of WP3 work. Such activities have resulted in the development of extensive context management functionality supported by ontologies for the semantic service description for Mobile, PC, Consumer Electronics and domotic domains, with focus on service functional capabilities and location information (AmiLoc). This work was supported by the design and implementation of a particularly useful ontology visualization tool (VantagePoint) that will surely be of great interest both in the context of *Amigo* as well as other research work.

Summary of WP issues

WP1 Finished in Year 1

WP2 Finished in Year 1

WP3 Ongoing, two deliverables produced on time. A positive point was the VantagePoint application. It is important that WP3 finish on time in order to avoid introducing delays in other work packages. Moreover, WP3 should clearly show:

- The use of middleware and integration of middleware functionality in other work packages.
- How it enhances and advances the *Amigo* applications.
- The impact on system performance issues.

Series title, number and
report code of publication

VTT Publications 649
VTT-PUBS-649

Author(s)
Niskanen, Ilkka

Title

An interactive ontology visualization approach for the domain of networked home environments

Abstract

This study concentrates on the construction process of a software tool which is able to interactively visualize ontologies, particularly in the domain of networked home environments. Special attention in this study is given to the design and implementation of the interaction mechanisms between the user and the visualization. This study is carried out by using a constructive research method supported by literary review and a test case, in which the constructed approach is validated.

In networked home environments domestic devices can interact seamlessly with each other and with-in home and external networks, and users can easily manage home devices, both locally and remotely. Networked home environments are described by ontologies, which are controlled vocabularies that describe objects and the relations between them in a formal way. The Web Ontology Language (OWL) semantic mark-up language was created for publishing and sharing ontologies, and it provides mechanisms for creating all the components of an ontology: concepts, instances, relations and axioms.

The OWL language is complex and looking at an OWL ontology for the first time can be overwhelming. Several visualization approaches have been developed to enhance the understanding of ontologies, but these existing visualization approaches are domain independent, graph based and not suitable for visualizing ontologies that describe networked home environments. The prototype software tool constructed in this study addresses these shortcomings by visualizing ontologies in a domain-specific way. It provides extensive interaction possibilities to help users to communicate with the visualized data. This approach makes OWL ontologies more interesting and concrete, and above all easier to comprehend.

The main achievement of this study is an ontology visualization tool prototype called VantagePoint. The results of the validation indicate that with VantagePoint it is possible to accurately create and visualize semantic models of real-world networked home environments. Through the extensive query construction features it is possible to access any data contained by the models. Although the validation also reveals some minor deficiencies, it must not be forgotten that VantagePoint is still in a prototype phase and further research and development work will be undertaken.

ISBN
978-951-38-7033-1 (URL: http://www.vtt.fi/publications/index.jsp)

Series title and ISSN	Project number
VTT Publications 1455-0849 (URL: http://www.vtt.fi/publications/index.jsp)	489

Date	Language	Pages
July 2007	English, Finnish abstr.	112 p. + app. 19 p.

Name of project	Commissioned by
Amigo	

Keywords	Publisher
Ontology visualization, interactivity, domains, networked home environments, software tools, visualization tools, Web Ontology Language, semantic mark-up language, validation, VantagePoint	VTT Technical Research Centre of Finland P.O.Box 1000, FI-02044 VTT, Finland Phone internat. +358 20 722 4404 Fax +358 20 722 4374

Julkaisun sarja, numero ja raporttikoodi

VTT Publications 649
VTT-PUBS-649

Tekijä(t)

Niskanen, Ilkka

Nimeke

Ratkaisumalli verkottuneita kotiympäristöjä kuvaavien ontologioiden interaktiiviseen visualisointiin

Tiivistelmä

Tämän tutkimuksen tarkoituksena on rakentaa ohjelmistotyökalu, joka visualisoi verkottuneita kotiympäristöjä kuvaavia ontologioita. Tutkimuksessa keskitytään erityisesti käyttäjien ja visualisoidun datan väliseen vuorovaikutukseen. Tutkimus toteutetaan käyttämällä konstruktiivista tutkimusotetta, jota tukee kirjallisuuskatsaus sekä validointi.

Verkottuneessa kotiympäristössä laitteet ovat saumattomassa vuorovaikutuksessa sekä toistensa että kodin ulkopuolisten verkkojen kanssa. Käyttäjät voivat helposti hallita laitteita niin paikallisesti kuin etäältä käsin. Verkottuneet kotiympäristöt mallinnetaan ontologioilla, jotka kuvaavat formaalilla tavalla ympäristössä olevia objekteja sekä niiden välisiä suhteita. OWL-kieli mahdollistaa ontologioiden julkaisemisen ja jakamisen sekä tarjoaa mekanismit ontologioiden sisältämien komponenttien kuvaamiseen

Työskentely ontologioiden kanssa on monimutkaista ja etenkin aloittelevilla OWL-kielen lukijoilla on usein suuria ymmärtämisvaikeuksia. Datan visualisoinnin on todettu merkittävästi edesauttavan datan sisältämän informaation omaksumista ja siksi useita ontologioita visualisoivia työkaluja on kehitetty. Nämä työkalut eivät kuitenkaan huomioi ontologioiden sisältämän datan luonnetta, vaan hyödyntävät pelkästään graafipohjaisia visualisointimetodeja. Tässä tutkimuksessa rakennettava visualisointityökalu ottaa huomioon ne erityisvaatimukset, joita verkottuneita kotiympäristöjä kuvaavat ontologiat asettavat. Se tarjoaa käyttäjille kattavat vuorovaikutusmahdollisuudet, joiden avulla he voivat kommunikoida visualisoidun datan kanssa. Lisäksi työkalu tekee OWL-ontologioista mielenkiintoisempia, konkreettisempia sekä ennen kaikkea ymmärrettävämpiä.

Tämän tutkimuksen tärkein saavutus on ontologiavisualisointityökalun prototyyppi VantagePoint. Suoritetun validoinnin tulokset osoittavat, että VantagePointin avulla on mahdollista luoda ja visualisoida olemassa olevia verkottuneita kotiympäristöjä kuvaavia semanttisia malleja. Työkalun prototyypin jatkotutkimus- ja kehitystyö pyrkii korjaamaan validoinnin aikana ilmitulleita puutteita sekä laajentamaan työkalun käyttömahdollisuuksia entisestään.

ISBN
978-951-38-7033-1(URL: http://www.vtt.fi/publications/index.jsp)

Avainnimeke ja ISSN
VTT Publications
1235-0621 (nid.)
1455-0849 (URL: http://www.vtt.fi/publications/index.jsp)

Projektinumero
489

Julkaisuaika
Heinäkuu 2007

Kieli
Englanti, suom. abstr.

Sivuja
112 s. + liitt. 19 s.

Projektin nimi
Amigo

Toimeksiantaja(t)

Avainsanat
Ontology visualization, interactivity, domains, networked home environments, software tools, visualization tools, Web Ontology Language, semantic mark-up language, validation, VantagePoint

Julkaisija
VTT
PL 1000, 02044 VTT
Puh. 020 722 4404
Faksi 020 722 4374

VTT PUBLICATIONS

635 Intelligent Products and Systems. Technology theme – Final report. Ventä, Olli (ed.). 2007. 304 p.

636 Evesti, Antti. Quality-oriented software architecture development. 2007. 79 p.

637 Paananen, Arja. On the interactions and interfacial behaviour of biopolymers. An AFM study. 2007. 107 p. + app. 66 p.

638 Alakomi, Hanna-Leena. Weakening of the Gram-negative bacterial outer membrane. A tool for increasing microbiological safety. 2007. 95 p. + app. 37 p.

639 Kotiluoto, Petri. Adaptive tree multigrids and simplified spherical harmonics approximation in deterministic neutral and charged particle transport. 2007. 106 p. + app. 46 p.

640 Leppänen, Jaakko. Development of a New Monte Carlo Reactor Physics Code. 2007. 228 p. + app. 8 p.

641 Toivari, Mervi. Engineering the pentose phosphate pathway of *Saccharomyces cerevisiae* for production of ethanol and xylitol. 2007. 74 p. + app. 69 p.

642 Lantto, Raija. Protein cross-linking with oxidative enzymes and transglutaminase. Effects in meat protein systems. 2007. 114 p. + app. 49 p.

643 Trends and Indicators for Monitoring the EU Thematic Strategy on Sustainable Development of Urban Environment. Final report summary and recommendations. Häkkinen, Tarja (ed.). 2007. 240 p. + app. 50 p.

644 Saijets, Jan. MOSFET RF Characterization Using Bulk and SOI CMOS Technologies. 2007. 171 p. + app. 4 p.

645 Laitila, Arja. Microbes in the tailoring of barley malt properties. 2007. 107 p. + app. 79 p.

646 Mäkinen, Iiro. To patent or not to patent? An innovation-level investigation of the propensity to patent. 2007. 95 p. + app. 13 p.

647 Mutanen, Teemu. Consumer Data and Privacy in Ubiquitous Computing. 2007. 82 p. + app. 3 p.

648 Vesikari, Erkki. Service life management system of concrete structures in nuclear power plants. 2007. 73 p.

649 Niskanen, Ilkka. An interactive ontology visualization approach for the domain of networked home environments. 2007. 112 p. + app. 19 p.

Julkaisu on saatavana	Publikationen distribueras av	This publication is available from
VTT PL 1000 02044 VTT Puh. 020 722 4404 Faksi 020 722 4374	VTT PB 1000 02044 VTT Tel. 020 722 4404 Fax 020 722 4374	VTT P.O. Box 1000 FI-02044 VTT, Finland Phone internat. + 358 20 722 4404 Fax + 358 20 722 4374

ISBN 978-951-38-7033-1 (URL: http://www.vtt.fi/publications/index.jsp)
ISSN 1455-0849 (URL: http://www.vtt.fi/publications/index.jsp)

VTT PUBLICATIONS 649 An interactive ontology visualization approach for the domain of networked... Ilkka Niskanen