Towards Semantic Interoperability in Virtual Organisations

A. Gehre
Institute for Construction Informatics, Technical University of Dresden, Germany

P. Katranuschkov
Institute for Construction Informatics, Technical University of Dresden, Germany

V. Stankovski
Faculty of Civil and Geodetic Engineering, University of Ljubljana, Slovenia

R.J. Scherer
Institute for Construction Informatics, Technical University of Dresden, Germany

ABSTRACT: Virtual organisations (VOs) are amongst the most advanced forms of doing business to date. Their emergence and growing capacities are closely related to the networking and collaboration capabilities provided by a supporting ICT infrastructure. However, whilst considerable progress in the development of such infrastructures has been achieved over the years, interoperability still remains a major challenge.
In this paper we analyse the nature of the semantic interoperability problem, review the state of the art, and derive a set of requirements. On that basis, we propose a novel semantic interoperability framework for VOs, utilising current semantic Web technologies, and suggest possible design and implementation methods for its achievement. Reported is in-house work of the authors as well as on-going research in the frames of the EU project InteliGrid (IST-2004-004664).

1 THE VO INTEROPERABILITY PROBLEM

Virtual organizations have emerged in the last years as an answer to the changing socio-economic challenges brought about by the globalization process. Today they are quickly becoming a preferred organizational form for one-of-a-kind businesses delivering one-of-a-kind products, such as AEC/FM.

According to (Camarinha-Matos & Afsarmanesh 1999), a virtual organization (VO) is "*a temporary alliance of enterprises that come together to share skills or core competences and resources in order to better respond to business opportunism, and whose co-operation is supported by computer networks*". The Globemen project defines a VO as a *"customer solutions delivery system created by a temporary and reconfigurable ICT enabled aggregation of core competencies"* (Karvonen et al. 2003).

These and other definitions, as e.g. from (Camarinha-Matos et al. 2005), emphasize the importance of ICT for the effective functioning of a VO. Whilst there are many other aspects that distinguish VOs from traditional organisations, such as differences in teamwork, management style, resource utilisation etc., the efficiency of the installed ICT infrastructure is a widely recognised prerequisite for the operability of a VO. In that respect there are two major issues to be considered: (1) appropriate communication networks, and (2) interoperability.

Indeed, interoperability is amongst the most critical issues for the successful development and further growth of the VO paradigm. Connecting computers by fast communication channels is only a preliminary first step. Interoperability targeting efficient information sharing requires to go a step further. According to the *ICH Glossary* interoperability is defined as:

> *"the ability of information systems to operate in conjunction with each other encompassing hardware, communication protocols, applications, and data compatibility layers" (ICH 2004).*

Of these, data compatibility is of greatest importance for the interoperability in VO collaborative networks. It is a prerequisite for establishing a common shared language for interactions between the heterogeneous partners and services in the VO.

The *Dublin Core Metadata Glossary* (Dublin Core 2004) elaborates in more detail the data compatibility layer by defining three sub-layers, namely *syntactical, structural* and *semantic* interoperability:

> "Syntactic *interoperability is achieved by marking up our data in a similar fashion so we can share the data and so that our machines can understand and take the data apart in sensible ways; for example, XML, EAD and MARC.*

Structural *interoperability is achieved through data models for specifying semantic schemas in a way that they can be shared; for example, RDF.*

Semantic *interoperability is achieved through agreements about* content description standards*; for example, Dublin Core, Anglo-American Cataloguing Rules." (Dublin Core 2004)*

However, such content description standards can ensure semantic interoperability only if a domain ontology that suffices a set of basic features is established, in accordance with the following definition based on (Gruber 1993):

"An ontology is a formal, explicit specification of a shared conceptualization."

In this extended form of Gruber's initial definition *formal* means that the ontology should be machine readable (and interpretable), e*xplicit* means that it contains clear, unambiguous, assertive definitions of concept types and constraints modelling the targeted domain of discourse, and *shared* means that it is used to define a common standard in that domain. The term *conceptualization* refers to the objects, concepts and other entities that are presumed to exist in the domain of interest and the relationships that hold them (Genesereth & Nilsson 1987).

While the concept of interoperability is not new for VOs, how it needs to be applied, interpreted and implemented at practical level is subject to continuous controversial discussions, especially because achievement of efficient interoperability is generally seen as a major factor for a number of vital VO features (cf. Barbini 2001):

1 *VOs are subject to market-driven cooperation*. Not the available interoperability channels, but business opportunities determine the dynamic collaborative structure of VOs, i.e. heterogeneity of the consortium is inherent and underpins the requirement of using flexible structures and methods of interoperability, capable of integrating several parallel and oppositional business domains and branches.

2 *VOs are subject to complementarity*. Each VO partner excels in particular subprocesses and/or has critical knowledge about the market. Interoperability has to ensure that technical/technological fertilisation and concertation takes place, i.e. the knowledge and services of each involved partner should be available for effective exploitation.

3 The *dynamic participation of VO partners* requires that organisations can connect into the VO network and disconnect from it in a dynamic way. Therefore, just-in-time information about available services, actors, resources etc. is necessary and that information must be updated regularly.

4 *Process and resource sharing* is a major challenge for VO interoperability. By blurring single enterprise's boundaries, partners can be enabled to work together, integrating business processes and sharing data, information and knowledge resources.

5 There is *no single organisational structure* for all virtual enterprises; rather, the organisational structure and information flows depend on the business to be exploited and on the partner characteristics. This *organisational polymorphism* is another major challenge for the provision of interoperable methods that are generally applicable.

This set of issues advises that methods, services and structural definitions composing the interoperability framework of a VO must be designed as flexible, robust and fault tolerant entities, with special emphasis on maintainability. Current state-of-the-art solutions only tackle the problem from the technology side. Support methods and tools are available that allow to unify the process of accessing interfaces, facilitate finding of services and describe input/output parameters in an automated, machine interpretable way. However, such technical interoperability support ensures only access compatibility and at the most provides for syntactic interoperability. For more flexible management of information and "*meaning*", adequate semantic interoperability has to be established that enables communication on the basis of an elaborated common semantic vocabulary of concepts, i.e. on *ontological level*.

By this:

- Interfaces are freed from the burden of defining specialised functions with fixed input/output parameters that are only applicable for a specific task, i.e. connecting two VO partners or services would be no longer a matter of establishing hardwired communication channels, but becomes a challenge of sharing the meaning of the content;
- The interpretation of communicated content is separated from the communicative act itself, ensuring a maximum of flexibility and robustness for extracting interpretable and necessary information;
- Flexible business process integration becomes possible by establishing an expressive and sound semantic business process model that can then be supported by a coherent business process integration methodology and services.

Figure 1 below illustrates how a common ontology can facilitate integration between semantically heterogeneous tools. Indeed, on the high level, the relationship between terms, models and reality can be represented by the so called *Ogden triangle* (Ogden & Richards 1994). In IT the terms used are symbols that *designate* concepts, e.g. the term "building" designates the concept "building" which *stands for* a real world artefact. The problem is that different tools would typically use different models *related to* that real world artefact, depending on their specific objectives and domains of interest (structural analysis, HVAC, cost estimation etc.), i.e. for each tool its own Ogden triangle can be drawn. However, by *committing* to a shared set of concepts (the ontology of the VO), they would be capable to 'understand' each other and to communicate about a mutual domain of discourse. It is not necessary that each tool interprets the ontology concepts in the same way, it is only needed that its observable actions are consistent with the definitions in the ontology.

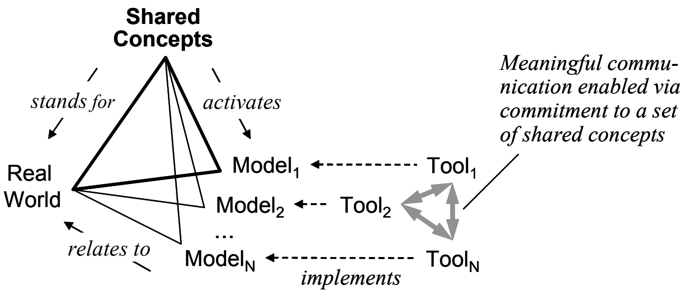

Figure 1. Ogden's triangle revisited. Interoperability between heterogeneous tools is enabled by a common ontology layer.

The price that has to be paid for the benefits of a common ontology layer is the increased development effort regarding complex algorithms for interpreting and formulating the content of interactions. Sustainable development of metadata ontologies and achieving agreement to these is the other challenge to face.

This brief outline of pros and cons shows why a properly elaborated framework for semantic VO interoperability is so important to meet the key requirements arising from the five major VO features mentioned above. Moreover, it provides valuable hints for establishing an appropriate development methodology.

2 STATE OF THE ART IN WEB-BASED SEMANTIC INTEROPERABILITY

The benefits of web-based semantic interoperability and the purpose of ontologies for its achievement are explained in a number of recent publications (cf. e.g. Berners-Lee et al. 2001, Miller 2003, Herman 2004). The hierarchical position of ontology specifications and their inter-realtionship to other specifications related to the World Wide Web is best illustrated by the so called Semantic Web Stack.

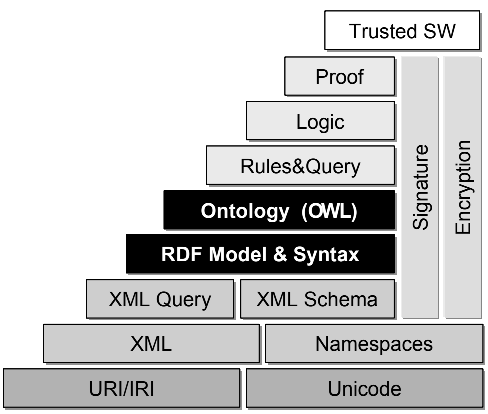

Figure 2. The Semantic Web Stack (after Herman 2004). Ontological layers are highlighted.

Languages

An important prerequisite for achieving semantic interoperability is the ability to make use of rich computer languages, such as the Web Ontology Language OWL (W3C 2004d) allowing to describe the various entities in the computing environment. OWL has been developed recently on top of the existing XML and RDF standards (cf. W3C 2004a, c) which did not appear adequate for achieving efficient semantic interoperability. Thus, although XML DTDs and XML Schemas seem sufficient for exchanging data between parties who have previously agreed to some set of shared definitions, their lack of constructs to describe the deeper meaning of these definitions prevents machines from reliably performing this task. For example, when a new XML vocabulary is introduced, the same term may be used with (sometimes subtle) different meaning in different contexts, and different terms may be used for items that have the same meaning. RDF and RDF Schema (RDFS) address this problem by allowing simple semantics to be associated with identifiers. With RDFS (W3C 2004c), one can define classes that may have multiple subclasses and superclasses, and can define properties, which may have subproperties, domains, and ranges. In this sense, RDFS can be seen as a simple ontology language. However, in order to achieve interoperation between numerous, autonomously developed and managed schemas, richer semantics are needed. For example, RDFS cannot specify that the classes *Person* and *House* are disjoint, or that a construction company has exactly four sub-contractors. In summary, OWL adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes. OWL provides three increasingly expressive sublanguages (Lite, DL, Full) designed for use by specific communities of implementers and users.

Web Services

Recent developments in IT have introduced the concept of *Web services*, which are self-contained, self-describing, modular applications that can be published, located, and invoked across the web. Web services perform functions that can be anything from simple requests to complicated business processes. Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service (cf. Vasudevan 2001). There seems to be a consensus that the future of e-business collaboration will be through Web services; so, obviously this is an aspect of interoperability that has to be considered when analysing the needs of future IT infrastructures utilising semantic interoperability.

Although generic in nature, OWL has been designed with the clear intention to support Web service discovery – a functionality by which Web services can be described, advertised, and discovered by others. Indeed, with OWL it is especially convenient to describe the characteristics and the functionalities of Web services. An important initiative of W3C in this respect is the Web Ontology Language for Web Services OWL-S, which evolved from its precursor DAML-S (cf. DAML 2004, W3C 2004b). OWL-S is an OWL-based Web service ontology, which gives Web service providers a core set of markup language constructs for describing the properties and capabilities of their Web services in unambiguous, computer-interpretable form. The OWL-S markup of web services aims to facilitate the automation of web service tasks including automated service discovery, execution, interoperation, composition and execution monitoring.

Support Tools and Frameworks

Work on tools and systems using and processing ontologies to achieve semantic interoperability brought about a number of promising developments.

For modelling ontologies with visualization support mainly the *Protégé* ontology editor and knowledge acquisition system is used today (Protégé 2005). It allows users to construct domain ontologies, customize data entry forms, and enter data. It is also a platform which can easily be extended to include graphical components such as graphs and tables, media such as sound, images, video, and various storage formats such as OWL, RDF, XML, and HTML. The Protégé API makes it possible for other applications to use, access, and display knowledge bases created with Protégé.

A full fledged ontology management infrastructure is provided by KAON which is an open-source system targeting business applications (cf. Gabel et al. 2004). It includes a comprehensive tool suite allowing easy ontology creation and management, and provides a framework for building ontology based applications. An important focus of KAON is the scalable and efficient reasoning with ontologies. The KAON system is the public continuation of the semantic web research and development effort that has lead to the commercial tools of the *Ontoprise* company (cf. Ontoprise 2005).

Finally, a number of ontology and metadata frameworks and servers exist, supporting RDF and OWL ontology languages. Very promising are the *Jena* framework (Jena 2005) and the *Kowari Metastore* (Kowari 2004). Jena is specifically suited to develop Java based Semantic Web applications. Among other features, it provides a programmatic environment for RDF, RDFS and OWL, including a rule-based inference engine. Kowari Metastore is an Open Source database for the storage and retrieval of metadata that is held in the form of short *subject-predicate-object* statements.

However, in practice and practice-oriented research there are yet little known efforts fostering ontologies and semantic interoperability. The focus is still mainly on tools and frameworks for general technological ontology support, rather than on practical business-relevant issues. One of the few exceptions in the construction domain is the e-COGNOS project (Lima et al. 2003). It specified and developed an open model-based infrastructure and a set of tools that promote consistent knowledge management within collaborative construction environments. Another valuable contribution has been made by the ISTforCE project which developed tools and applications that use an engineering ontology specification to facilitate end user interfaces and product data model translations (Katranuschkov et al. 2003). Continuing and extending that work, Gehre et al. (2004) suggested an agent-enabled peer-to-peer infrastructure for cross-company teamwork in building construction that uses metadata ontologies for managing distributed resources and teams. However, in spite of such efforts, all achievements are rather fragmentary and of limited practical value. Coherent comprehensive frameworks for VO interoperability are still an open issue.

3 REQUIREMENTS TO A SEMANTIC INTEROPERABILITY FRAMEWORK FOR VO

Support of VO interoperability by a semantic framework encompassing semantic services and ontologies offers good prospects, but it is also subject to a series of non-functional requirements that have to be carefully considered when realising it in practice.

Reactivity / Response times

Interoperability and interactions in general that involve ontology-based semantics tend to be highly resource and time consuming. Therefore, a major non-functional requirement is to ensure reasonable system response times. Whilst longer response times (10 seconds or longer) are acceptable for sophisticated ontology-based directory services, background services (e.g. responsible for translating ontology-based interactions into direct actions) ought to be much faster (less than 1 second) to achieve end-user acceptance.

Usability

Usability is a major requirement for end-user interface facilities that are upgraded by ontological features. It pertains to the level of training required to achieve a goal with the application, to which extend its use is intuitive, and the balance between efforts and benefits of usage.

Maintainability

For larger and longer lasting VO projects, the initially developed project specific ontology definitions will typically not be final in every respect. Therefore, the framework for ontology-based VO interoperability must enable flexible tackling of changes and extensions by means of adequate ontology management tools.

Compatibility with industry standards

As interoperability is one of the main success factors for VOs, support by ontological semantics has to follow available standards to the greatest possible extend. End users need reliability in API definitions, relying on standards fosters end-user acceptance.

Installability / Executability

VOs typically use heterogeneous IT infrastructures. Therefore, installability and executability are major concerns with regard to middleware technology. All tools and services that are needed to provide semantic interoperability with local applications have to be installable and executable at each VO partner.

Scalability

Ontology frameworks tend to have problems with larger ontology definitions and higher inherent complexity. Therefore, robustness of an ontology based VO interoperability framework is very important with regard to larger amounts of data. Consequently, the requirement for scalability introduces the need for less complex ontology definitions, which is contradictory to the needs for comprehensiveness and completeness.

Upgradeability

This requirement is specifically related to the need to upgrade/update ontology definitions at runtime. Automated schema generators translating ontology definitions to programming language classes can greatly help in meeting this requirement.

4 SERVICES FOR ACHIEVING SEMANTIC INTEROPERABILITY

Efficient semantic interoperability in a VO environment targets several operational objectives:

- By using semantic interactions between heterogeneous software of VO actors the established message based communication must create a desirable *loosely coupled system*;
- Semantic interoperability must provide advanced metadata support services enabling *management of decentralised heterogeneous information*;
- Coherent support for business process integration, a key enabler for modern IT supported VO businesses, should be provided by *dedicated ontology services*; such services can help in ad-hoc modelling, handling and processing of business processes thereby *fostering B2B integration*.

Especially for achieving the third of the above goals, the five principle life cycle phases of a business process must be taken into account (cf. Foss 1998):

1. Identification of a process prototype or, if necessary, ad-hoc definition of a new one;
2. Identification and procurement of information and services needed to perform the process, including e-payment;
3. Integration of all process input into a suitable processing workflow to perform the business process;
4. Performing the business process;
5. Disassembling of process components when the process is finally completed.

Except for the step of performing the process itself (step 4), each of these lifecycle phases can significantly benefit from the support of advanced ontology services. Whilst many useful ontology services can be envisaged directly from that life cycle, we identify four particular types as essential for achieving business process integration support. Three of these are dedicated to general VO business tasks, whereas the fourth stands for a (potentially unlimited) set of domain specific ontology services.

4.1 *General VO Ontology Services*

Harmonised management and integration of information and services is needed to establish support for business process integration. This includes the management of information directly related to the structure of and information logistics within the VO. Three essential General VO Ontology Services can be identified for that target.

VO Logistics Service

This ontology service has to provide information about general VO entities, such as actors, roles, persons, organizations, organizational structures and principal legal constraints of the VO. Even though VOs can be subject to organisational changes during their lifetime, information provided by this service is more static. It is defined at the initialization stage of the VO and only updated if something changes in the VO structure. This support service provides necessary input for the other services listed below.

Resource Sharing Service

Distributed resources in a VO network, i.e. services and information contained in databases and files, are typically heterogeneous with regard to type, structure, syntax, access directives, specifics of the owning local network and available metadata support. In order to exploit all available resources in a VO network coherently, a common resource sharing service that unifies the resource access and resource metadata services is needed. Essential prerequisite for the achievement of such a service is the commitment to (1) a formal explicit ontology defining resource metadata concepts, and (2) a superordinate ontology describing basic structural VO concepts. Using these ontologies the Resource Sharing Service can provide the necessary functionality to achieve resource integration in business processes. Thus, it directly supports the business process life cycle phase 2.

Resource Integration Service

Performing business processes that are based on distributed resources requires an integration approach. This can be fulfilled by a formal Resource Integration Service that provides information about what kinds of resources are put together and how in order to compose the business process. Using the Resource Sharing Service outlined above, the Resource Integration Service describes the information flows that establish the business process, i.e. what information resources are input for which service resource, and how these services are composed in order to establish the business process. Thus, it supports the business process life cycle phases 1, 2, and in particular 3.

4.2 *Domain Specific Ontology Services*

Along with the general VO services it is important to provide a spectrum of dedicated Domain Specific Ontology Services. Tightly associated with the specific business domain(s) of the VO, such services target VO support through domain specific functionality. This may include model extraction, translation and mapping, as well as specialised domain specific methods.

An especially important semantic service that is domain specific by nature but at the same time generic within its business domain, is the service for *information retrieval*. Its purpose is to extract information from structured documents, databases and product models through proprietary interfaces and then provide the extracted information through a shareable "condensed" ontological model. This service plays a core role for the achievement of efficient semantic interoperability. It may be augmented by services for automated model mappings to facilitate VO interoperability on model level. As a whole, these services provide domain specific support to the business process life cycle phases 2 and 3.

As an example in the construction domain, such services can be used to extract information from server-based IFC Product Models (IAI 2005), with subsequent mapping to a condensed ontological representation. This representation can then serve as a metadata repository, as well as a data pool for applications and users that are only interested in specific model aspects. The objective is to capture and describe an activity in a process on general "business object" level, using terms like "storey", "room", "wall" and messages like "calculate thermal resistance" or "show all rooms with office usage", rather than technical low-level model objects like *IfcSpace*, *IfcWall*, *IfcLocalPlacement*, *IfcCartesianPoint* etc. The concepts of this higher level ontological representation should act as logical containers for the data and the respective operations needed to carry out the required activity. Responsibility how exactly this activity is performed is at the respective domain tools and/or the user. More information to that is provided in the InteliGrid report D13.1 (Turk et al. 2004).

5 CONSENSUAL ONTOLOGIES

A consensual ontology is an ontology that is developed for the purpose of information and knowledge sharing in a certain domain and to which all players (and tools) in the VO acting in that domain have agreed to commit. Consensual ontologies are a prerequisite for semantic interoperability as they establish the common shared language that is the foundation of all ontology services interactions. Accordingly, commitment to existing ontology standards is an important issue. This applies to commitment to ontology language standards (e.g. OWL), as well as to existing consensual ontologies, e.g. the web services ontology defined as OWL-S.

A major design consideration in ontology development is that a structured system of interconnected specialised ontologies utilising Semantic Web technology is much more effective with regard to defining, maintaining, interweaving and processing them, than deploying a huge and complex single ontology. The semantic web ontology language OWL provides advanced mechanisms for referencing and integrating ontology entities of separate definitions. It thereby offers the needed flexibility for constructing powerful systems of hierarchical interrelated ontologies.

The two types of ontology services defined in the preceding chapter prompt for a hierarchical ontology structure comprising two systems of ontology definitions. The first describes generally applicable VO business constructs, and the second focuses on domain specific business processes. However, seamless support for business process integration can only be ensured if both these systems are coherently defined and elaborated to a granularity that allows to specify business processes in uniform, ad-hoc manner. For real practice scenarios this is a highly challenging design requirement.

VO Business and Environment Ontology

This generic part of the ontology definitions for semantic interoperability in VOs consists of an extensible core set of concepts targeting specific VO interoperability tasks. It establishes the underlying common language for the VO Logistics, Resource Sharing and Resource Integration Services. It captures information about general VO entities, ranging from actors and other VO structural concepts to administrative information and service resources within the VO. Service aggregation mechanisms, e.g. based on the established OWL-S standard, provide concepts for ontology supported automated integration of complex business processes.

Specialised Ontologies for the VO domain

Specialised ontologies for dedicated business processes of the VO are the basis for all Domain Specific Ontology Services. They integrate and build on the generic part of the ontology definitions. As currently relevant targets are seen in first place model mapping and extraction, as key enablers for achieving B2B integration.

Regarding the building construction domain, an OWL-IFC ontology can provide for model interoperability based on a subset of the IFC standard (IAI 2005), extended by ontology specific concepts and rules. Such OWL-IFC information can be created directly by qualified ontology-enabled applications, or extracted from IFC product model servers by a dedicated Construction Domain Ontology Service. It can greatly facilitate upgrading the IFC Project Model to a true Semantic Web ontology resource of real practical relevance.

6 SUGGESTED REFERENCE ARCHITECTURE FOR ONTOLOGY SERVICES

A generalised *Ontology Service System* comprises a set of self-sustained semantic services that can be structured into a layered architecture as suggested in Figure 3 below. Information flow with business services/applications and middleware services that provide dedicated resource access is only roughly represented on this figure, in order to focus on the structure of the ontology services itself.

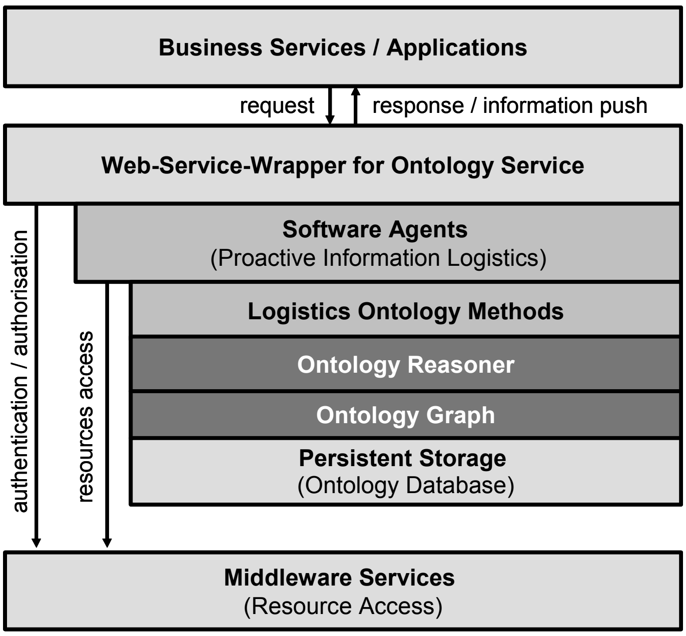

Figure 3. Suggested Architecture for a Web-Enabled Ontology Services System.

Following the modern philosophy of a Service Oriented Architecture (SoA), common access to the Ontology Service System is provided through a *Web Service Interface*, so that the ontology service is pervasive and platform neutral. In accordance to that, a *Web-Service-Wrapper* has to map HTTP-based SOAP interactions to the internal service infrastructure. Such interactions include: (1) client authorisation requests to external central authentication authorities, (2) resource access to middleware services, (3) response to requests from business services and applications, and (4) possible proactive information push to web entities that have registered to the recurrently provided service.

As essential central component of the Ontology Service System we suggest to use *software agent technology* for coordination and cooperation processes. This has several advantages (cf. Reitbauer 2005):

- use of components that are more closer to reality and provide improved level of abstraction;
- flexibility in modelling coordination behaviour, i.e. modelling ontology-based management processes as encapsulated agent behaviours, thereby ensuring extensibility of the service (especially with regard to domain specific services);
- enabling more efficient runtime reactivity with regard to dynamic events with the help of rules for ad-hoc composition of agent behaviours;
- utilisation of cyclic behaviours to automatically maintain the information pool of the Ontology Service System;
- capability to proactively initiate an *information push* to registered entities in the VO environment; such an information push can be initiated e.g. by cyclic agent behaviours (time intervals), by triggers fired by rules specified in the ontology, or as a result of some planning activities.

Underlying the software agents layer, an *ontology graph and related reasoning facilities* provide object-oriented access to the ontology. This allows for structured ontology management on model and entity level. Access on entity level is provided by the ontology graph via specialised access methods. Based on the ontology graph, a *reasoner engine* enables the derivation of additional assertions entailed from the model, together with any optional ontology information and the axioms and rules associated with the reasoner. The primary use of this mechanism is to support the inference process of deriving additional facts from instance data and class descriptions. A reference implementation for such an ontology framework is the *Jena* framework for building Semantic Web applications (Jena 2005).

Finally, the Ontology Service System must also incorporate persistent storage for ontology data and models by using a back-end database engine integrated through the ontology layer. Jena provides for that purpose an abstract Java interface for ontology database model management.

7 CONCLUSIONS AND FURTHER WORK

The presented ontology-based approach for tackling VO interoperability generalises early high-level conceptual development work done by the authors in the frames of the EU project InteliGrid. This work builds upon experience gathered from related in-house implementations and studies, as well as on findings from the earlier EU projects ISTforCE and e-COGNOS.

At this stage, several opportunities of the suggested approach can already be drawn up:

1. It enables management of various types of distributed information resources in uniform manner;
2. It promotes integration on the basis of a common language that describes VO concepts and inherent dependencies in a flexible way;
3. It can help to extend product data technology use beyond its current scope of application (mainly CAD);
4. Last but not least, much of the outlined concepts and services are generic and therefore applicable in various domains.

The approach is extensible regarding further intelligent ontology services, due to the coherent integration of software agents and ontological reasoning. Potential candidates for extensions are triggers based on VO events, automated business object processing, and advanced automated notification services. More detailed capabilities and options are expected to emerge in the progress of the work.

ACKNOWLEDGEMENTS

The presented research is partially carried out in the frame of the InteliGrid project (IST-2004-004664) funded by the EC. The contribution of the funding agency is herewith gratefully acknowledged.

REFERENCES

Barbini, F. 2001. *Fairwis: a Virtual Organization Enabler*. In Proc. OES-SEO2001 "Int. Workshop on Open Enterprise Solutions: Systems, Experiences, and Organizations", Rome, 14-15 Sept. http://cersi.luiss.it/oesseo2001

Berners-Lee, T., Hendler, J. & Lassila, O. 2001. *The Semantic Web*. Scientific American 284 (May/2001), 34-43.

Camarinha-Matos, L. & Afsarmanesh, H. 1999. *The Virtual Enterprise Concept*. In: Infrastructures for Virtual Enterprises. Kluwer Academia Publishers, ISBN 0-7923-8639-6.

Camarinha-Matos, L., Afsarmanesh, H. & Ollas, M. /eds/ 2005. *Virtual Organizations: Systems and Practices*. Springer, ISBN 0-387-23755-0 / (eBook) 0-387-23755-7, 340 p.

DAML 2004. *DAML Services*. http://www.daml.org/services/owl-s/

Dublin Core 2004. *Dublin Core Metadata Glossary*. http://library.csun.edu/mwoodley/dublincoreglossary.html

Foss, J. D. 1998. *Brokering the Info-Underworld*. In: Jennings, N.R. & Wooldridge, M.J. /eds./ "Agent Technology. Foundations, Applications, and Markets", Springer.

Gabel, T., Sure, Y. & Voelker, J. 2004. *KAON – An Overview*. Informal SEKT Deliverable, 61 p. http://kaon.semanticweb.org/main_kaonOverview.pdf

Gehre, A., Katranuschkov, P. & Scherer, R. J. 2004. *Agent-enabled Peer-To-Peer Infrastructure for Cross-Company Teamwork*. In: Dikbas, A. & Scherer, R. J. /eds./ Proc. "ECPPM 2004 – eWork and eBusiness in Architecture, Engineering and Construction", Istanbul, 8-10 Sept., Balkema, ISBN 04-1535-938-4, pp. 445-452.

Genesereth, M. R., Nilsson, N.J. 1987. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann.

Gómez-Pérez, A. /ed/ 2002. *A Survey on Ontology Tools*. Deliverable 1.3, EU Project OntoWeb, IST-2000-29243, 96 p.

Gruber, T. 1993. *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition 5, 199-220.

Herman, I. 2004. *Introduction to the Semantic Web*. W3C Presentation 27 Oct 2004 http://www.w3.org/Consortium/Offices/Presentations/SemanticWeb/Overview.html

IAI 2005. *IFC/ifcXML Specifications*. © International Alliance For Interoperability. http://www.iai-international.org/Model/IFC(ifcXML)Specs.html

ICH Architecture Resource Center 2004. *ICH Glossary*. http://www.ichnet.org/glossary.htm

Jena 2005. *Jena – A Semantic Web Framework for Java*. http://jena.sourceforge.net/

Karvonen, I., van den Berg, R., Bernus, P., Fukuda, Y., Hannus, M., Hartel, I. & Vesterager, J. 2003. *Global Engineering and Manufacturing in Enterprise Networks (GLOBEMEN)*. VTT Symp. 2003, p. 224-395. http://globemen.vtt.fi

Katranuschkov, P., Gehre, A. & Scherer, R. J. 2003. *An Ontology Framework to Access IFC Model Data*. ITcon Vol. 8, p. 413-437, ISSN 1400-6529. http://www.itcon.org/2003/29

Kowari 2004. *Kowari Overview*. © 2001-2004 Tucana Technologies, Inc. http://www.kowari.org/

Lima, C., El Diraby, T., Fies, B., Zarli, A. & Ferneley, E. 2003. *The e-COGNOS project: Current status and future directions of an ontology-enabled IT solution infrastructure supporting Knowledge Management in Construction*. ASCE / Construction Research Congress 2003, Honolulu, USA.

Miller, E. 2003. *Weaving Meaning: Semantic Web Applications*. Presentation at INTAP, November 11, 2003, Tokyo, Japan. http://www.w3.org/2003/Talks/1117-semweb-intap/

Ogden, C. K. & Richards, I. A. 1994. *The Meaning of Meaning*. In: Gordon W. T. /ed./ *C. K. Ogden and Linguistics*, Routledge / Thoemmes Press, London, ISBN 0415103533, (first published: 1923).

Ontoprise 2005. *Ontoprise Products & Applications*. © 2005 ontoprise GmbH. http://www.ontoprise.de/content/e3/index_eng.html

Protégé 2005. *Protégé Overview*. http://protege.stanford.edu/overview/

Reitbauer, A. 2005. *The Value of Agents in Business Integration*. In: AgentLink News. Issue 17, April 2005, 12-13.

Turk, Z. /ed./ 2004. *InteliGrid Deliverable D13.1 – Semantic Grid Architecture*. The InteliGrid Consortium, c/o Univ. of Ljubljana, www.inteliGrid.com.

Vasudevan, V. 2001. *A Web Services Primer*. O'Reilly Media. http://webservices.xml.com/pub/a/ws/2001/04/04/webservices

W3C 2004a. *Extensible Markup Language (XML)*. http://www.w3.org/XML/

W3C 2004b. *OWL Web Ontology Language for Services (OWL-S)*. http://www.w3.org/Submission/2004/07/

W3C 2004c. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation 10 Feb 2004, http://www.w3.org/TR/rdf-schema/

W3C 2004d. *Web Ontology Language (OWL)*. http://www.w3.org/2004/OWL/ .