Distributed Collaboration in the Context of the Semantic Web

Jakob Beetz, Jos van Leeuwen, and Bauke de Vries
Eindhoven University of Technology, The Netherlands

Keywords: Collaborative design, Multi-agent systems, Semantic Web

Abstract: In this paper we are proposing a Multi Agent System (MAS) framework for the facilitation of distributed collaboration in the AEC/FM domain. We are showing how the stack of technologies developed in the Semantic Web community can be put to use for the specific requirements of the building industry. Based on our earlier findings and developments in the area of logic based knowledge representations for the Design and Construction industry, we are outlining how these can form the semantic foundations of internal agent representations and their interconnection using speech acts.

1. INTRODUCTION

Several ongoing and future joint research efforts such as InteliGrid (Turk et al, 2004) in the AEC/FM domain have identified the 'intelligent' interconnection of information resources, data processing and retrieval services as well as distributed design project collaboration covering all lifecycle stages as key areas of future business models. In order to establish and profit from such a scenario a lot of research and development as been done in other industry domains, generally referred to as the 'Semantic Web Initiative'. Its main challenges lie in structuring information in such a way, that not only its human consumers are able to make 'sense' of it, but that the nature of its content will also be discoverable, readable, and processable by machines. The foundation of the approach that is taken by the Semantic Web community is by constructing knowledge on the basis of Description Logic (DL). Here, a set of formal axiomatic statements is used to create a model of

Jos P. van Leeuwen and Harry J.P. Timmermans (eds.) Progress in Design & Decision Support Systems in Architecture and Urban Planning, 313-323.
8th International DDSS Conference, 4-7 July 2006, Eindhoven University of Technology.

a domain of interpretation, referred to as ontology. In knowledge representation systems, a separation is made between the terminology itself (the TBox) and the instances of concepts (asserted statements, the ABox). In the AEC/FM domain, the organisation of the body of knowledge has a long tradition in researching known as Building Information Model (BIM). The most widely spread encoding of BIMs today, the Industry Foundation Classes, however have several drawbacks: Although the information is encoded in a modular way, separating the different domains into subparts, the technical means to do this (the EXPRESS way of modelling knowledge) is not based on rigid logics as required by knowledge engineering tools and has not been designed to distribute the model over a network structure. This very property on the other hand prevents the utilization of methods and technologies that have been developed in the Knowledge Engineering and Artificial Intelligence community. One of the essential building blocks to harness the new paradigms that are being developed by a large research community is to create a notation of the domain knowledge of the AEC/FM community in an appropriate format. Since experiences from the past show that the creation of a model that is agreed upon by a large number of players in the field usually takes a long time, we suggest that such a model should be founded upon structures that we already have agreed upon.

In the past, several projects have aimed at the construction of ontologies for the AEC/FM domain (Lima et al, 2003) or other, less expressive kinds of structures such as taxonomies LexiCon (Woestenenk, 2002) and BARBi (Bell et al, 2004). In our own research, we have successfully adapted the Industry Foundation Classes standard as a starting point for an ontology formalized in the W3C standard Ontology Web Language. After successfully creating this TBox-part, we have recently added the ABox part, enabling actual instances of concepts by converting them form the standard IFC-export many of the recent CA(A)D packages generate (Beetz et al 2006).

In this paper we are going to report on the methods, the implementation, the implications and practical use cases how a logically-based Building Information Model can be integrated with knowledge engineering tools. We are going to outline a framework architecture for a Multi Agent System (MAS) that acts as connecting mechanism between various resources e.g. human domain experts, product databases, rule bases or wrappers around numerical simulations. We are going to show how agents can make use of mapping and reasoning services that help to facilitate and negotiate the terms of information exchange between agents using different information models. Furthermore, we will report on the kinds of communication acts that can be expected to happen between these agents, and how a protocol should be designed to facilitate those communication acts.

1.1 Logic-based models for information and knowledge representation

Knowledge representation systems to capture structured information about a domain of expertise have been around since the early days of information technology. They have a long tradition in Epistemology, a discipline in philosophy, and have been applied with varying success in many different domains. The two most important families of knowledge representation systems that are relevant in the current semantic web developments are the frame-based systems family (Minsky 1975) including semantic networks (Woods 1975) and the Description Logic (DL) (Baader et al. 2002). More recent developments, that apply these methods such as the OIL and DAML have introduced a blended strategy of these methods, lending the frame-slot-filler concepts from one and the rigid axiomatic logic including entailment and theorem proving from the other. This leads to the possibility to encode ontologies, "formal specification of a shared conceptualization" (Gruber 1993)

OIL and DAML have been merged into the joint OWL standardization effort, which uses the concepts introduced in RDF(S) and extends them with entailment operators, multiple range and domain restrictions and cardinality constraints. In order to make use of the reasoning engines, which enable consistency checking and inference of implicit knowledge a hierarchical set of different flavours of OWL has been proposed, ranging from generally provable, but expressively poor OWL Lite to undecidable OWL Full.

To enable reasoning and logic programming systems to make more deeply 'hidden' implicit knowledge explicit, various efforts to standardize encodings of rule languages are currently struggling for acceptance. The most important ones among them are SWRL and RuleML that allow the encodings of rules and their variable bindings in way that integrates into the rest of the semantic web stack. The most important aspect of these efforts is their crucial position in the whole concept of an enhanced interoperability: Only if a successful separation of knowledge and application code is achieved, will the Semantic Web succeed in this regard (which, again, is only one of the possible, and maybe not its most important one).

1.2 Agent system frameworks and protocols

A lot of different definitions for the notion of 'agent' have been coined in literature. Even worse, the existing definitions have been blurred by 'marketing – speech' during the recent hype. Perhaps the most agreed upon definition is to be found in (Huhns and Singh, 1998) who define agents as "active, persistent (software) components that perceive, reason, act, and communicate".

While perception, reasoning, and acting strongly depend upon the specific domain in which an agent is designed for, no multi agent system can exist if they don't have a common way of communicating with each other. The communication acts themselves can be separated into three different levels:

1. The syntactical means to exchange messages: the envelopes of a letter, the addresses of sender and receiver and the post stamps to pay for the delivery service;
2. The syntactic encoding and decoding of the messages: how to write and read the language of the messages;
3. The semantics of the message: how to convey and interpret the actual content of the messages, ensuring mutual understanding.

Since the first layers are the common denominators of many different purposes, a number of standards and implementations have been developed in both research and industrial environments to enable developers from different domains to tailor systems for their own purposes. For these layers, standards have been proposed, amongst which the Agent Communication Language (ACL) standard of the Foundation for Intelligent Physical Agents (FIPA) (now a member of the IEEE) is the most accepted one. In this standard, a number of atomic speech acts have been defined that regulate how agents should communicate amongst each other, how these communications acts should be interpreted and how to react to them.

The missing level of these principle communication acts however, are the domain-specific higher-level semantics. In order to successfully design, implement and use a generic framework, a range of semantic representations of the various bodies of knowledge of the domains involved, differing with regard to the context must be available. Here, the context is not only limited to the general issues such as roles, permissions, locations and trustworthiness of the participating agents, but also on a content level that has to accommodate a wide variety of building-industry related knowledge, such as the different stages of a project within its complete lifecycle, its various granularities of details, its discipline-dependant views and aspects of a building project, its various local, legal and process related aspects among other things. Since one of the most important lessons that have been learned from the various approaches of modelling all these aspects into a single, omni-potential building information model is the general impossibility of the very idea, the distributed nature of the semantic web stack might turn out as a viable improvement in future systems. The compelling idea here is, that isolated autonomous units that have their own specialised internal representations of certain expert areas are enabled to make use of common denominators by using (cascading) semantic mapping services that act as translators and negotiators between these separated islands of knowledge and expertise.

2. EXISISTING WORKS IN THE FIELD

While the overall approach of agent-based computing, and the notion of service oriented distribution of software architectures is still in its infancy, a number of prototype systems and frameworks have been developed in both research and industrial environments.

MAS architecture have been successfully applied in a wide range of industries such as telecommunication.

The application of industrial-scale MAS for the support of concurrent engineering like PACT (Chutosky et al, 1997), SHADE (McGuire et al 1993), the NASA aeronautical collaboration environment (Monell and Piland, 2000), DIDE (Shen and Barthes, 1997) and others have a strong focus on CE tasks evolving around large-scale, complex product modelling in the aerospace industry. In production environments like these, but also in other engineering domains such as automotive and shipbuilding industries, a number of advantages over the AEC/FM problem set from the concurrent engineering perspective can be identified:

- **Centralized structure**: Due to the scale of the corporations required to design an manufacture products like airplanes, ships and cars, the power to enforce the use of advanced technologies differs quite substantially from what is usually found in the AEC/FM industry

- **Product design cycle duration and instantiation**: While the design and development of new airplanes, ships and cars usually take several years and the number of actual instantiations goes into the thousands, the “one-of-a-kindness’ found in the AEC/FM industry has often been identified as one of the key obstacles to adapt new technologies.

- **Heterogeneity and scale of participants:** Another advantage of the mentioned industries is the scale of the businesses involved and the heterogeneity of software tools used in day-to-day praxis.

- **Necessity for formalization.** In machine building industries, the final products are almost always made by machines themselves. Also, a good deal of the assembly and manufacturing process is being designed for automation from the ground up. This leads e.g. to a widely adopted habit of complete 3D models including annotation, material properties etc.

A good overview of applied MAS in engineering domains as well as other fields, such as telecommunication, electricity planning, and the 'classic' playground scenarios such as meeting and trip scheduling, trading and stock brokering can be found in (Parunak, 1998) and (Shen, 2000).

Recent developments in both the MAS and the Semantic Web and web services communities show, that the very concepts of agents and services are started to get blended into another, as web services enhance their current non-semantic protocols with semantic annotations (OWL-S) and some developments are aiming at bridging agents in the classic frameworks with regular WSDL-based web services (Petrie et al, 2003), (Greenwood and Calisti, 2004).

3. OVERALL FRAMEWORK ARCHITECTURE

Our proposed framework consists of series of services, template-based agents, bridging and connection services based on the Java Agent DEvelopment Framework (JADE), a FIPA-compliant Open Source effort of a multitude of academic and industry institutions led by a not-for-profit board organisation. The underlying system takes care of lower-level technical communication tasks, and provides a common environment for interactions of distributed agents across different physical machines and operating systems. Several additions and plugins allow the use and integration of other technologies such as a number of language codecs, among them XML-notated RDF, which in turn is the basis of OWL.

Our main contributions and additions are the domain-specific ontology services, agent templates and exemplary wrappers that provide semantic interfaces for generic applications and traditional web services. Our logic-based OWL notation of the Industry Foundation Classes (ifcOWL, 2006) is the core content language of inter-agent communication in this system. By making use of the most widely accepted format of interoperability in the industry domain, we are able to study the implications on the use of such systems using real-world data instances. To achieve this, we have implemented a set of tools that enables the conversion of building information data modelled in generic CA(A)D packages into logic-based representations that enable semi-automatic processing within the agents. Here, the derived representations not only act as content language used in communication acts, but also as foundation of internal agent representations. The advantage of this approach is the facilitation of methods and implemented tools to query and reason in generic ways over instances of data that leads to a significant decrease in demand in manual interoperability work.

4. LOGIC-BASED KNOWLEDGE REPRESENTATIONS FOR THE AEC/FM DOMAIN

Although the building information model that is described by the IFCs is a good starting point and already captures a lot of the principle information that is being communicated in AEC/FM projects, the quality and amount of knowledge that is necessary to describe in formal ways in order to make it machine-processable in (semi-)automatic ways goes beyond that. Despite the presence of a principal hierarchy of concepts and their manifestations (such as “a door is a building product, which is a specialization of a generic product, which – among other descriptions – might have geometrical representations”), the real work in capturing useful information on certain aspects of a future or existing building is the connection of these hierarchical elements in form of higher-level dependencies and rules. These rules and advanced semantic descriptions can capture knowledge on different levels of abstraction and in different contexts, e.g.

- **Domain-depended**: The important attributes of structural elements such as walls, beams and columns differ from the points of view of a structural engineer to those of, e.g. the HVAC expert within a project.
- **Local and regional context**: Starting from the basic SI-units that are being used in different regional settings to complex building regulation information such as the fire-protection rating of building elements, a lot of care and a significant amount of knowledge engineering has to be undertaken to formalize ‘simple’ characteristics as “a fire protection level X of a product Y is describes the property of Y to withstand a standardized temperature Z for the duration of m minutes before structural collapse”.
- **Project context**: Despite the differences to larger-scale projects in other domains as described earlier in this paper, and the significant amount of work that goes into modelling dependencies per-project, the addition of machine-readable project-constraints and dependencies in larger building projects might significantly reduce the number planning failures who to a large portion can be traced back to miss-communications within project teams. Adding automated monitoring systems in form of agents that re-evaluate a given set of constraints each time modifications have been made requires the easy integration of such rules and constraints on a project instance level.
- **Organizational context**: The concept of dynamic, project-bound business-entities that form “Virtual Organizations” for a short period of time create the need of dynamic and flexible models of setting up and

maintaining such temporary structures. Although generic models to cover these aspects are under development in other e-business areas, the need to adapt these models to the specifics of the building industry is quite apparent and is one of the key focuses of the InteliGrid project.

The encoding and mapping of these kinds of higher-level formal knowledge descriptions as describe in the examples above and their exposure as (commercial) services on large networks structures (both for external audiences as general but still confidential offers or within project specific intranets) might turn out as one of the cornerstones of future business models in the building industry that has to cope with an increasing amount of complexity in globalized and fast-moving markets.

5. TYPES OF DOMAIN SPECIFIC COMMUNICATION ACTS

In order to establish a domain specific, extendable and flexible system that can accommodate as many potential scenarios as possible, we have constructed a number of use cases to gather the potential interactions in agent systems. The use cases are taken from a scenario in which design team members in different phases of a building design perform common tasks such as checking the compliance to regulations, look up products and their specifications and estimate the impact of certain decisions on the overall performance of the building. In traditional collaborative setups this requires a lot of manual work preparing building model information aspects for external application, launching and operating the external processes, regather and reimport the new information. In our proposed system we aim at solving some of these tiresome, reoccurring interoperability tasks by orchestrated agent interactions.

From these use cases we can generalize a number of reoccurring families of tasks. They can be divided into the following categories:

- **Formalization and formulation of queries, constraints, and desired results**. Every interaction of human domain experts with the system consists of a number of formalization and formulation tasks that spread over a wide range of syntactic and semantic expressiveness. In most ‘simple’ cases this is a natural language question from one project participant to a single project team member or groups of individuals transmitted in a traditional form of email, postings to a group message board or annotation to a model or plan using traditional design collaboration methods and tools. The MAS plays the role as matchmaker and negotiator in this scenario. In more advanced scenarios a user is offered a choice of standardized conformance checks or project-specific compliance tests that result in short checklists, detailed reports or highlighted and annotated parts of the model. For these preconfigured checks domain application developers have to formalize the most frequent inquiries, and encode their necessary inputs and outputs for both the user and the system backend. In the most advanced scenarios the user formulates the constraints or queries on her own, using appropriate interfaces such as (simplified and domain-specific) query and rule languages. Again, the results in these cases vary from simple boolean conformance answers to complex incorporations of new information into the existing building information model.

- **Semantic mapping, match making, and constraints propagation**. In order to gather the additional information starting from existing information and queries provided directly by the user or predefined by the interface agent front end, the system has to translate and map the existing information (the BIM instance) and the queries into the proper equivalences in other domains, languages, regional settings or levels of granularity. In simple cases these mappings are the conversion between different unit systems, in advanced settings appropriate product specifications or building regulations in a targeted project area are being looked up. A frequent operation here is the conversion from one domain-specific representation of a model into another. Match making involves the lookup of semantically described services in directory facilitators, their respective content languages, preconditions, and result formats.
- **Task decomposition and plan generation.** In this family of tasks, complex actions are divided into smaller chunks that in turn are solved in either sequential or arbitrary order by specialized agents. The generation of such plans can come from previous successful solutions that are being fed back into the internal representation of the agent or might be suggested by a service or by another agent. For complex operations that involve the consultation of multiple agents, a facilitator agent acting on behalf of the user has to schedule the necessary steps and keep track of the dependencies and results.
- **Inference, constraints checking and solving.** Once the appropriate services and agents have been found and the necessary types of inputs have been generated by mappings, the actual atomic sub processes are being invoked. Again, the variety of actual interactions computations carried out here range over a wide band of possibilities. From simple product database queries that are resolved within seconds, to inferences of implicit model information to complex simulation runs that need days to produce a solution.
- **Compilation, preparation, presentation, and notification of results.** The results calculated by the individual services or agent have to be

mediated back to the user. In simple scenarios, the user is presented reports that indicate regulation compliance, building performance estimates, or product details using classic media such as texts, diagrams, and images. In other cases, this compilation and presentation of results require a (temporary) change or addition of the BIM instance itself.

Although they are interdependent, the type families of operations described in this section cannot be seen in a sequential order. Each of these operations comprises complex and difficult computational challenges in themselves. The effort in bringing connecting and integrating them into a single platform, referred to as orchestration, forms the main focus of our current research efforts.

6. DISCUSSION AND FUTURE WORK

In this paper we have proposed and outlined a multi agent based framework for the use in distributed AEC/FM collaboration scenarios. We have shown how the methods and technologies developed in the context of the Semantic Web initiative can be adapted and facilitated for the respective needs of the building industry. We have shown what the principal communication acts in such systems are, and how they can be implemented and coordinated in a generic FIPA-compliant framework.

After the completion of the basic infrastructure, we will implement a number of exemplary modules and agents that demonstrate the use and implications of the proposed framework system in engineering practise. Although we strongly believe in the added value of such distributed approaches, a lot of research and experiments still have to proof the added benefit. However effective these scenarios are 'in the laboratory' many of the real obstacles to overcome can only be studied in heterogeneous environments using real-world scenarios. Only then is it possible to identify the drawbacks and implications of such systems. One of the main thresholds for these experiments is the willingness of stakeholders in the industry to apply and adapt these technologies even in their early stages.

REFERENCES

Baader, F., D. Calvanese, and D.L. McGuinness, 2003, *The Description Logic Handbook: Theory, Implementation, Applications* Cambridge University Press, Cambridge, UK.

Beetz, J., Leeuwen, J.P van, Vries, B.de, 2006, "Towards a topological reasoning service for IFC-based Building Information Models in a Semantic Web Context", to appear in *Proceedings of the ICCCBE 2006*, Montreal, Canada.

Bell, H, Bjørkhaug, L, Rønning, J. ,2004, "ICT-Platform for Object Oriented Knowledge in the Building and Construction Industry", *B4E Conference Maastricht*, http://www.barbi.no/doc/pdf/2004_10_14_Paper_B4E.pdf

Cutkosky, M. R., Englemore, R. S., Fikes, R. E., Genesereth, M. R., Gruber, T. R., & Mark, W. S., 1997, "PACT: an experiment in integrating concurrent engineering systems". In M. N. Huhns & M. P. Singh (Eds.), *Readings in Agents,* San Francisco, CA, USA: Morgan Kaufmann, p. 46–55.

Greenwood, D., and Calisti, M., 2004, "Engineering web service - agent integration". In *Proceedings of the IEEE international conference on systems, man & cybernetic.* The Hague, Netherlands.

Gruber, T., 1993, "A translation approach to portable ontology specifications". Knowledge Acquistion **5**(2): 199-220.

Huhns, M. and Singh, M. (eds.), 1998, *Readings in Agents*, Morgan Kaufmann.

ifcOWL, 2006, http://semantic.ddss.nl/ontologies/2006/03/ifcOWL2x2

Lima, C. P., Fiès, B., Lefrancois, G., Diraby, T. E., 2003, "The challenge of using a domain Ontology in KM solutions: the e-COGNOS experience," In: proceedings of *10TH ISPE 2003*, Funchal, Portugal, p. 771-778.

McGuire, J. G. Kuokka, D. R.et al., 1993, "SHADE: Technology for knowledge-based collaborative engineering," in *Concurrent Engineering: Research and Applications*, **1**(3), 1993.

Minsky, Marvin, 1975, "A Framework for Representing Knowledge", in Patrick Henry Winston (ed.), *The Psychology of Computer Vision*, McGraw-Hill, New York, 1975.

Monell D.W. and Piland W.M., 2000, "Aerospace Systems Design in NASA's Collaborative Engineering Environment." *Acta Astronautica*, **47**(2–9): 255–264.

Parunak, H.V., 1998, "Industrial and practical applications of DAI", in *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Cambridge Massachusetts, p. 377-421.

Petrie, C. J., Genesereth, M. R., Bjornsson, H., Chirkova, R., Ekstrom, M., & Gomi, H., 2003, "Adding ai to web services" in Proceedings of *Agent mediated knowledge management 2003,* Stanford, ca, USA.

Shen, W., & Barthes, J. P., 1995, "Dide: A multi-agent environment for engineering design," In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, San Francisco, CA, USA.

Shen W., Norrie, D.H. and Barthes J.P., 2000, "Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing", Taylor & Francis, London, UK, 2000.

Turk, Ž., Dolenc, M., Nabrzyski, J. , Katranuschkov, P., Balaton, E. , Balder, R., Hannus M., 2004, "Towards Engineering on the Grid" in Dikbas A and Scherer R (eds.), *eWork and eBusiness in Architecture, Engineering and Construction* Taylor & Francis Group, London, p. 179-186.

Woestenenk, K., 2002, "The LexiCon: structuring semantics" in *Proceedings of CIB W78 conference on Distributing Knowledge in Building*, Vol 2, Aarhus, Denmark, p. 241-247.

Woods, William A., 1975, "What's in a Link: Foundations for Semantic Networks," in: D.G.Bobrow & A.M.Collins (eds.), *Representation and Understanding: Studies in Cognitive Science*, Academic Press, New York, p. 35-82.