Theme:

Title: **Human-Centred Knowledge-Based Model Access Service for Engineers**

Autor(s): Alexander Gehre, Peter Katranuschkov

Institution(s): Institute of Applied Informatics in Civil Engineering, TU Dresden

E-mail(s): Alexander.Gehre@cib.bau.tu-dresden.de

Peter.Katranuschkov@cib.bau.tu-dresden.de

Abstract: *This paper presents the Model Access Service (MAS) developed in the EU ISTforCE project (IST-1999-11508). It uses the capabilities of product data technology to provide value added services in a human-centred, web-based collaboration environment. Along with standard services for product model access on model and object level, it encompasses also two advanced knowledge-based features: a Reasoning Agent and an Explanation Component.*

MAS is developed as a self-contained system that can be used in a variety of ways. In the ISTforCE framework, it is integrated with an external Product Data Server (PDS) and a Core Information Server (CIS) which provide the necessary infrastructure enabling the full functionality of MAS.

Standard functionality for model level access is provided by using MAS as a central tool that enables generalised data exchange capabilities to all relevant product models in one or more construction projects, even if they are stored on different product data servers. This is done on the basis of user and account information retrieved from the CIS. To enable the use of the product models directly, MAS provides an API for generalised RPC-based model access on object level.

One of the two advanced features of MAS is the integrated Reasoning Agent which is responsible for replying to sophisticated queries concerning a specific structural design model, extending the IFC2x core model. It allows clients to use AI planning methods remotely, to generate solution sequences that combine the computed single Solver items. The second advanced feature incorporated in the MAS is the Engineering Ontology layer. It enables the translation of IFC data to the vocabulary and semantics familiar to end users. This unique feature of the MAS, distinguishing it from most proposed product data services to date, is augmented by an Explanation Component which opens many of its functions to standard Web Browsers.

Keywords: *Product data technology, Human-centered collaboration, Engineering Ontology, Middleware, EJB, ISTforCE.*

Introduction

Since the early 90s several research and development projects have proven that product data technology (PDT) can successfully and beneficially replace the traditional document-centred approach to project realisation. Today, with the development of the IFC project model by the IAI, the product modelling paradigm is being rapidly introduced in commercial software. However, the actual state of the art of PDT application in the AEC domain is limited basically to CAD data exchange and, in a few cases, some project-centred data management facilities. What is missing are *human-centred product model services* supporting the engineer with additional knowledge about the models to:

- provide customisable user-friendly capabilities for knowledge procurement and modification of the data, and
- enable concurrent access to the information on the several projects he/she may be working on at the same time.

These are the primary objectives for the design and the implementation of the integrated Model Access Service (MAS) presented in this paper.

MAS was designed as the central product model access point of the concurrent engineering environment developed in the EU ISTforCE project (IST-1999-11508). The innovative aspects in ISTforCE that are strongly supported by MAS are [1, 2]:

- the provision of a human-centred instead of a project-centred environment to meet in a better way the needs of the end users who typically have to work on multiple projects at the same time;
- the achievement of an open collaboration environment where end users and providers of engineering information, services and tools can meet;
- the enabled object level data exchange and sharing,
- the provision of an infrastructure for on-line e-Business, including legal and financial aspects of the collaboration process.

The core of the ISTforCE environment is comprised of the developed Concurrent Engineering Services Platform (CESP) where engineers, architects, managers, chief information / internet officers and providers of knowledge, information and software can come together [2]. The platform is designed to support a large (potentially unlimited) number of engineering services that can be plugged into it, as well as the needs of different users that have to deal with product model data of different projects. Therefore, in the ISTforCE framework MAS has to act not only as a model server that provides whole product models, e.g. as STEP physical file (SPF) export/import, but performs also knowledge-based services with real added value for the engineers and the application tools they use. These advanced services range from automated processing of complex queries concerning specific model details, up to a user-friendly Explanation Component providing an easy to use and understand view on the product model data to engineers.

The product models maintained by MAS can remain outsourced on several project-specific product or document data servers emphasising the ISTforCE approach that brings currently practised project-centred working to a new operability level. Projects and product models are assigned to specific CESP users as needed, by utilising the information that is retrieved from the ISTforCE Core Information Server (CIS). The latter provides all necessary data for that purpose, such as login and account data, as well as user related high-level project / services specifications.

Operational Objectives of the MAS

In detail, MAS aims at meeting the following operational objectives:

- Provide content information by enabling both personalised SPF-based access to whole or partial product models, as well as object / attribute level access using appropriate RPC mechanisms
- Support different model access paradigms
  (As a lot of different clients should be enabled to use the services of MAS, a range of different access paradigms had to be implemented. Thus, along with SPF-based model access and RPC interfaces, such as Java RMI and CORBA for application access on object level, a HTTP based web interface had to be provided as user-friendly interface as well)
- Develop a Reasoning Agent supporting the execution of complex automated structural design tasks to provide a practical example of the open architecture of the MAS and the possibilities to extend its functionality and the interoperability with client applications
  (The Reasoning Agent has to act in responsibility of client programs on the server side, enabling the outsourcing and the bundling of knowledge-based components)
- Develop an Explanation Component to facilitate human-centred product data retrieval for engineers not familiar with the technical IFC structure and specifications.

These objectives had to be achieved in an integrated and coherent approach, that additionally ensures the maintenance and extensibility of the system.

System Architecture

The MAS architecture, illustrated on Fig. 1 below, is comprised of three major modules: an Enterprise Java Beans (EJB) Container, an Enterprise Information System module (EIS) and a Web Container. Each of these modules is internally structured into several components (beans) dedicated to the realisation of specific tasks.

The central module of the system is the EJB Container. It provides all application logic components, as well as all plug-ins necessary for the communication with other external infrastructure elements. One of these external infrastructure systems is part of MAS itself. This is the EIS module with its Persistent Model Cache, which supports all MAS internal data and model storage functionality. The third essential component of the system architecture, i.e. the Web Container, provides the MAS www-interface, including the Explanation Component with the Engineering Ontology framework.

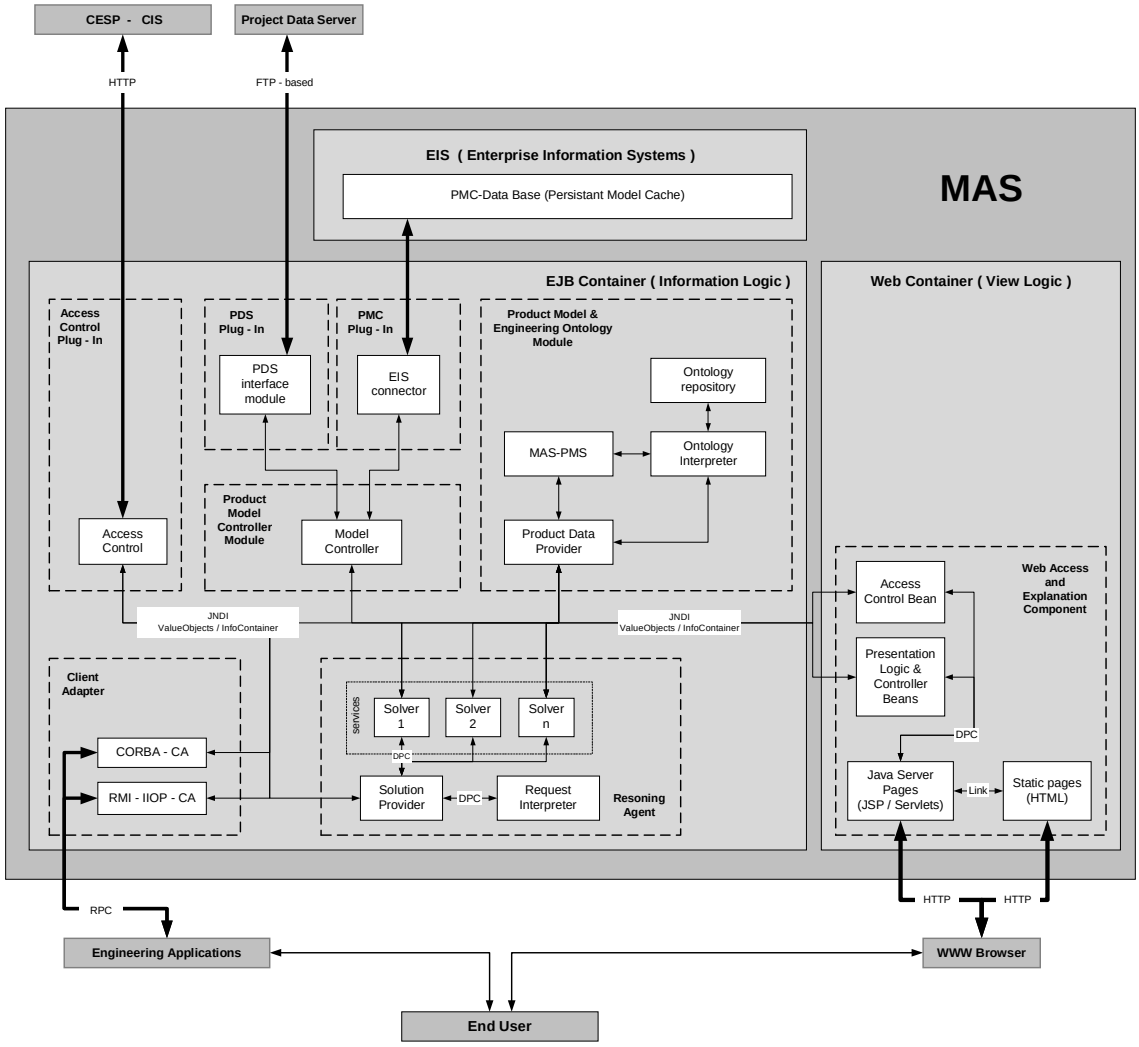

Fig. 1: Software architecture of the MAS

The end user has two possibilities to access the MAS services:

- *By using basic services for product data access* on model or object level that will normally be implemented by the engineering applications of the user, utilising the RPC API of one of the provided Client Adapters, as described further below. Such engineering applications are also the envisaged clients that may use the Reasoning Agent of MAS[1].
- *By using the human-centric www-interface* which allows to access many of the MAS services by a standard Web Browser.

[1] In ISTforCE, one such application has been prototyped as a practical example. This is the Knowledge-based Design Assistance Service for preliminary and conceptual structural design (DAS), developed by TU Dresden and FIDES, Munich.

Fig. 1 also shows the supporting external infrastructure, provided by other CESP components. The Core Information Server (CIS) provides to MAS all user related information and the PDS provides access to the product models the user has to work with.

Client Adapters to Support Different Access Mechanisms

As mentioned, one of the major objectives of MAS has been to enable the use of its services by different types of client applications, written in different programming languages and implementing different data access paradigms. To put this into practice, MAS implements different *Client Adapters*, each responsible for a separate API but based on a common conceptual framework. The Client Adapter approach was introduced to ensure strictly formalised separation of the application logic from the interface definition, while additionally providing a generic concept for easy maintenance and extensibility.

Currently, MAS encompasses an adapter based on Java RMI-IIOP, and a second one using the CORBA framework. They are implemented using a plug-in mechanism and are technically completely separated from each other. In addition, uploading / downloading of STEP physical files is enabled in much the same way as for individual objects by means of BLOBs.

A special Client Adapter provides MAS access to standard Web Browsers. It is implemented within the Web Container module of the MAS, and does not follow the normal plug-in mechanism of the other adapters. In fact, the Web Container utilises the Servlet and JSP (Java Server Pages) techniques, providing also plain HTML support. This browser-based interface is the main human-centred gateway to MAS, and is therefore also responsible to provide the Explanation Component described in detail further below. It uses a *web session approach* facilitated by the information retrieved from the CIS.

By using the developed plug-in technique, new adapters can easily be plugged in as it becomes necessary.

Basic Services for Product Data Access

Personalised SPF-based product model access

Product model access through SPF-based data exchange is a widely used practical approach and therefore one of the initially required basic functions to be supported by MAS. In the MAS architecture, clients using that approach can connect to the system using the communication and access methods that are most appropriate to them. Along with the possibility for model access on the basis of Java RMI and CORBA, the user can upload and download his/her models using a standard Web Browser.

In contrast to currently available document management servers that may also store and provide product data files, MAS is capable to integrate the models of different project-specific product data servers in a transparent manner. Supported by the information provided by the ISTforCE CIS, it manages various account / access control data, such as the projects in which the engineer is involved, his roles in these projects etc., and "knows" how to access the corresponding product data servers on technical level. The user gets a generalised access to all the product data that is relevant for him by a single tool and does not have to be aware of the technical logistics or the network topology of the distributed environment.

Additionally, MAS integrates a Persistent Model Cache to efficiently support long transactions. Hence, each user can operate on his/her model data without being forced to continuously check-out / check-in this data on a product data server any time he wants to use a model. This feature also ensures, that only consistent and completed model development steps will be stored at the PDS - an essential aspect w.r.t. version management issues on the PDS.

Product data access on object / attribute level

A central component of MAS is the integrated Product Model Service (PMS) acting on object / attribute level. It is implemented generically, enabling to import any product model and its corresponding EXPRESS schema. The PMS then dynamically instantiates the objects of the imported model and provides generic access methods for simple filtering / inspection / querying / retrieval / modification of these objects or their individual attributes.

The functionality of the PMS encompasses an adapted subset of the SDAI specification [3], implemented within the RMI / CORBA APIs. It enables applications to easily access and modify product data using appropriate RPC mechanisms. Session management a la SDAI is not implemented since MAS provides its own more advanced session management model based on EJB [4].

Additionally, for IFC data, extended functionality specifically adapted to the current IFC 2x project data model [5] is provided. This includes services enabling the retrieval of all objects in an *IfcBuilding* or *IfcBuildingStorey* container, tracking of related objects specified by subclasses of *IfcRelationship*, operations on *Property Sets*, retrieval of "composite" objects, such as all objects that collectively represent a building element (wall, column, beam …), etc.

This capability is essential both for the advanced services of MAS, like the Reasoning Agent and the Explanation Component, and for the development and the integration of client applications. The latter only need to provide their application-specific logic, whereas the RPC mechanisms of the MAS APIs can be utilised for all actions related to product model data.

Advanced Services

Two types of advanced services are included in the MAS system. These are: (1) the knowledge-based Reasoning Agent, and (2) the Explanation Component, based on the integrated Engineering Ontology framework. They are currently prototyped only for specific engineering domain tasks but can easily be extended by using the same software pattern. In fact, the conceptual architecture of MAS enables a potentially unlimited number of "helper" agents and ontology based processors to be added to the system.

Reasoning Agent

As already indicated, the Reasoning Agent is responsible to accept sophisticated queries concerning some specific product model and to execute appropriate automated tasks.

This service is provided by a special API mechanism based on a generic *Information Container* concept which introduces a set of fundamental data types that can be coherently used for various complex queries and assertions [6]. Its goal is to enable the packaging of any supported data definitions by a simple representation formalism, so that the specification of a generalised client / server model of communication, and the uniform definition of any more specific project data operations can be achieved. In this way, the interface to the Reasoning Agent is explicitly modelled, while ensuring its flexibility and extensibility.

To provide appropriate processing of such generic requests, it is necessary to make use of a preprocessing component. This component, called *Request Interpreter*, examines the query, decides if the request can be interpreted or not, and if necessary and possible, edits the request structure in order to facilitate the subsequent work of the *Solution Provider* component.

The *Solution Provider* itself uses AI planning methods that try to find a solution sequence based on the service of one or more of the available *Solver* modules. Therefore, if necessary, the *Solution Provider* decomposes the solution in several simpler subtasks that can then be executed successively.

The final part of the Reasoning Agent module are the *Solver* components. Solvers are "closed service" entities dedicated to the solution of specific problems. Broadly speaking, a Solver can be compared with a complex database request. It provides a single solution, while internally performing appropriately sequenced tasks. Additionally, a Solver may include a pre-condition and/or a post-condition description, refining and facilitating the work of the Solution Provider.

The power of the Reasoning Agent can be increased by implementing a larger number of Solvers and by enhancing their capabilities to solve complex tasks. Currently, MAS contains only a few Solvers, prototyped as proof-of-concept examples of the developed approach. Included are sophisticated product model querying / filtering methods that work across the model's data structures, and a more comprehensive Solver component that can help derive the structural analysis model of a frame structure on the basis of initial IFC building element data describing geometric, material and connectivity properties of these building elements.

Explanation Component Based on an Integrated Engineering Ontology Framework

The Explanation Component presents the second advanced feature of MAS. It provides a translation of the strictly formalised IFC data structures, which are not readily understandable to the end user, to the engineering vocabulary he/she is used to work with. This task is accomplished with the help of an *Engineering Ontology framework* integrated in the data structures of the MAS [7, 8, 9].

The Explanation Component module is activated by a client-side *Engineering Ontology Navigator* which provides the front-end user interface for browsing / inspecting / manipulating the product data by means of model requests utilising familiar engineering semantics. In this way, it becomes possible to understand

and handle the data not only indirectly, through CAD or other specialised engineering applications, but also directly, through a standard Web Browser. Thus, the engineer can retrieve and modify product model information fast and without deeper knowledge of the underlying IFC structure, whilst at the same time he can be aware of the engineering meaning and structuring of the data.

As a simple example, consider the ontology concept “frame” which is a firm part of the structural engineering vocabulary but is not contained within the IFC product model kernel. By utilising the developed Engineering Ontology framework, the engineer can use this concept to navigate through model parts defined as “frames”, that are “known” only as beams and columns in the IFC model. Another example is the “join” operation that brings together dispersed information within the model to one “object”, e.g. the ontology representation of a structural element contained in several IFC objects providing its material details, connectivity, geometry, location and so on.

The Engineering Ontology itself is completely based on the XML standard which enables its straightforward usage on the Web. It is comprised of three inter-related specification layers: (1) a Core Engineering Ontology Specification Schema (EOSchema), formally defined as an XML Schema instance [10], which provides the meta structures enabling the definition of all user-level concepts, (2) an extensible set of domain-specific ontology schemas, also based on the XML Schema specification, which imports and extends the core schema with appropriate user-level concepts, and (3) pure XML-based ontology definitions, providing the details of the defined concepts on the first two layers [9]. This "layered" approach greatly facilitates the development of adequate ontologies by domain experts because the basic XML syntax is much easier to learn and understand than the specific, semantically richer, but also more complex XML Schema constructs. Additionally, it is also easier to implement and has broader applicability.

The ontology specifications are stored in an Ontology Repository which provides functions to retrieve, update, add and validate them (see Fig. 2). It is implemented as a database and is accessible by an API that encapsulates the database functionality in more sophisticated RPC calls.

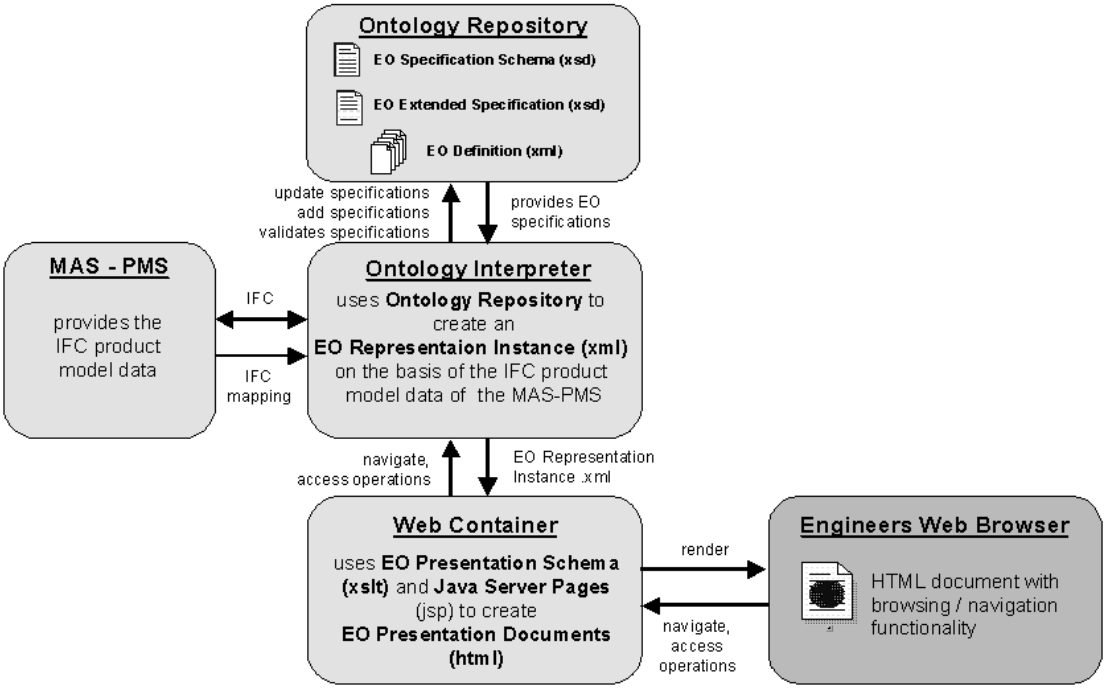

Fig. 2: The Explanation Component Information Flow

The central processing unit of the Engineering Ontology framework is the *Ontology Interpreter*. It is initialised with an exclusive reference to an instance of the MAS internal product model service, so that it can act on the product model directly.

The Ontology Interpreter uses the specifications provided by the Ontology Repository for real-time *mapping* between the IFC 2x model data and the implemented Ontology [6, 9]. However, as this is a very complicated task by itself, and because not all information is present in both models, a complete mapping cannot always be achieved. Therefore, the mapping process is facilitated by additional information provided in the detailed EO Extended Specification Schemas and in the EO Definition files. As a result of the process, the Ontology Interpreter generates *Engineering Ontology Representation Instances* as “pure" XML files. Such Representation Instances can be seen as a special ontology-based “view” on the underlying IFC model.

The Representation Instances are delivered to the Web Container which sends them directly as XML files to the client, if the client is an advanced ontology enabled application, or translates the XML files to a more readable presentation form. The translation is done by using XSLT and JSP technology to create one or more HTML files that can readily be presented by a standard Web Browser. Moreover, the HTML files do not only provide the “raw” ontology view of the model but are also enhanced by navigation and access operations, defined as hyperlinks in the HTML document that can again access the Web Container. Through this recursive navigation process, schematically illustrated on Fig. 2 above, a comprehensive browsing of the IFC product model becomes possible.

As a whole, the Engineering Ontology framework follows the MVC pattern (model-view-controller). The MAS internal PMS and the Ontology Repository constitute the model layer. The Ontology Interpreter acts as the Controller for the user actions and also as a *mapper* between the Engineering Ontology and the IFC 2x model. The view layer is represented by the Web Container which generates the final presentation documents.

Implementation Aspects

MAS is implemented in a highly modularised manner, on the basis of the Enterprise Java Beans technology. It makes use of the standardised EJB interface and the robust and consistent environment provided by the *Sun Java 2 Enterprise Server*. This ensures the realisation of an extensible, easy to maintain environment [4, 11].

One of the great advantages of the EJB technology is the comprehensive session support mechanism, facilitating the human-centred approach of the MAS. It is provided by the EJB Container for the application logic components, as well as by the Java Servlets in the Web Container. The personalised sessions are realised by using the information from the ISTforCE Core Information Server which provides the account and user information.

The architecture of the MAS, shown on Fig. 1, reflects the implemented modular approach. Within the server, the separate MAS modules communicate also on the basis of Java RMI-IIOP, supported by the JNDI framework. The information exchange between them is achieved by using the Information Container concept. Furthermore, it is possible to add, exchange or maintain components at runtime. For instance, if the integration of a new product data server for an additional project becomes necessary, it can easily be plugged in with minimal programming effort.

An Oracle database is used as internal information and data storage system. It is not an integrated part of the Enterprise Server but can also be regarded as a MAS module. In fact, it is plugged in by a specialised connector facilitating the access to the models that are stored in the PMS database.

The information exchange to the external MAS infrastructure is realised using state of the art standards. The interface to the CIS uses XML messages in HTTP-based calls. Currently, MAS is supported by only one Product Data Server whose API features advanced FTP-based functionality for SPF data exchange.

Conclusions

The Model Access Service presented in this paper introduces an innovative human-centred approach for working with IFC product model data. Its strengths are provided by the developed open and readily customisable architecture, the easy to use versatile methods to access and modify the product data and the robust and stable environment, hiding all technical logistics aspects from the end user. Especially beneficial to the end user is the capability to work on several product models in one or more construction projects at the same time. This feature is enabled by the efficient support for long transactions and the implemented robust access and session control mechanisms.

As a whole, MAS proved to be a powerful and convenient tool during the first tests of the complete ISTforCE platform. It worked very well with external infrastructure components, and client applications had no problems to connect to it. Engineers that explored the Explanation Component found the interface very convenient and easy to comprehend.

However, along with the achieved benefits, some aspects requiring further development work need to be mentioned as well.

1. The introduced Reasoning Agent framework requires a great number of Solvers for different tasks which are not implemented until now. This is a very extensive and time consuming work which needs to be done in future to provide real added value to construction practice and potentially bring MAS to market ripeness.
2. The developed Engineering Ontology framework has proven to be very practical for engineers, normally not so familiar with the IFC model specifications. However, the mapping process between the IFC model and the Ontology is very complicated, and often too slow. Additional work is needed to refine the mapping methods so that they can be fully usable in daily practice.
3. Whilst the use of the Enterprise Java Beans framework has been an excellent technical decision for the implementation of the MAS, EJB itself is under continuous development and may be replaced by a new, eventually partially incompatible version, which would require additional re-engineering work that cannot be foreseen in advance. The used Servlet, JSP and XML processing techniques are also changing rapidly and are therefore often not compatible downwards.

Acknowledgements

The presented research is carried out in the context of the IST programme funded partially by the EU. The contribution of the funding agency is herewith gratefully acknowledged.

References

[1] Turk, Z., Scherer, R. J., *Towards the next generation of civil engineering collaboration platforms*, Proceedings of the eBusiness and eWork 2001 Conference, Venice, Italy, (2001).

[2] Katranuschkov, P., Scherer, R. J., Turk, Z., *Intelligent Services and Tools for Concurrent Engineering - An Approach Towards the Next Generation of Collaboration Platforms*, ITcon Vol. 6, special issue: "Information and Communication technology advances in the European construction industry", www.itcon.org, (2001).

[3] ISO 10303-22 IS, *Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 22: Implementation Methods: Standard Data Access Interface*, ISO TC184/SC4, Geneva, (1998).

[4] Kassem, N. and the Enterprise Team, *Designing Enterprise Applications with the Java™ 2 Platform, Enterprise Edition*, Sun Microsystems, Inc., Palo Alto, USA, (2000).

[5] IAI, *Industry Foundation Classes: IFC 2x*, technical specification,
available at: http://cic.vtt.fi/niai/technical/IFC_2x/index.htm, © IAI, (2000).

[6] Katranuschkov., P., *A Mapping Language for Concurrent Engineering Processes*, Diss. Report, Schriftenreihe des Lehrstuhls für Computeranwendung im Bauwesen, Heft 1 (Hrsg. R. J. Scherer), Fakultät Bauingenieurwesen, TU Dresden, (2001).

[7] Gehre, A., Katranuschkov, P., *Engineering Ontology*, Deliverable D5-1, EU Project IST-1999-11508 ISTforCE, (2000).

[8] Woestenenk, K., *A Common Construction Vocabulary*, in: Amor R. (ed.) "Product and Process Modelling in the Building Industry", Proc. ECPPM'98, Building Research Establishment, Watford, 19-21 Oct., Clowes Group, Beccles, Suffolk, UK, (1998).

[9] Katranuschkov, P., Gehre, A., Eisfeld, M., *Engineering Ontology – Formal Representation of the Data Structures*, Deliverable D5-2, EU Project IST-1999-11508 ISTforCE, (2001).

[10] W3C, *XML Schema (Part0: Primer, Part 1: Structures, Part 2: Datatypes)*, W3C Recommendation, available at: http://www.w3.org, (2001).

[11] Bodoff, S., Green, D., et al., *The J2EE™ Tutorial*, Sun Microsystems Inc., Palo Alto, USA, (2002).