8th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS
S u c e a v a, R o m a n i a, M a y 25 – 27, 2 0 0 6

IMPROVEMENT OF THE DESIGN PROCESS BY USING SEMANTIC WEB TECHNOLOGIES

Sabin-Corneliu BURAGA[1], Marius CIOCA[2], Lucian-Ionel CIOCA[2]
"Al. I. Cuza" University of Iaşi, Romania – [1]busaco@infoiasi.ro
"Lucian Blaga" University of Sibiu, Romania – [2]{marius.cioca, lucian.cioca}@ulbsibiu.ro

Abstract. *The collaborative work in distributed environments has a significant role in the process of virtual product design within diverse CAD/CAM systems by successfully utilizing different (semantic) Web technologies, thus decreasing the time required for bringing new products on the market and improving the knowledge transfer between business partners. The paper proposes a concrete solution for building a virtual environment that can ensure proper communication facilities in order to design and implement a high complex project. The environment uses different storage methods for involved design process information and is based on the actual semantic Web technologies in order to offer a knowledge-oriented production system.*

Keywords: *Distributed environment, modeling, semantic Web technologies, virtual enterprise.*

Introduction

In the context of product design, one of the most important issues is the existence of a powerful IT (Information Technology) infrastructure in order to permit communication, cooperative work and exchange of information between the specialists can contribute, in a multifaceted manner, at the design of a given set of products.

The article's purpose is the presentation of a virtual computerized environment that would guarantee the improvement of the cooperation and of the operative communication between the members of a team formed, for a limited time period, for realizing of a high complexity project.

More concretely, the paper subject targets the virtual product design by a team from an university (in our case, "Lucian Blaga" University of Sibiu, Romania) and its industrial partners (e.g., suppliers) in order to give a proper solution for shortening of the time needed for realizing a certain product. On the other hand, the proposed environment could allow the knowledge exchange between teams composed by members from different domains of activity. This environment will provide a better relationship between the academic education institutions and the business actors existing in the same area, by exchanging fundamental and technological knowledge experience. Also, the promotion of ISO/STEP (*STandardized Exchange of Products*) standards [13, 14, 17] is required and encouraged, as well as their inclusion in new instruments as support for concurrent engineering.

The structure of the paper is the following: section called "*System Architecture: General Overview*" will provide details regarding the general system architecture of the environment aligned to STEP standards and section "*Using Modelling Languages*" will present the use of UML language for the process modeling phase. In the next section, our paper will give details about the manner of data storage within the proposed system. Additionally, a semantic Web-based solution is presented in section "*Capturing Knowledge by using Semantic Web Technologies*". The material ends with conclusions and further directions of our work.

System Architecture: General Overview

Important Aspects The architecture of the collaborative work system takes into account:

- The type of process interaction regarding time: synchronous or asynchronous;
- The geographical dispersion of the virtual team's members: local or distant;

- The main domain of interest of the persons involved in the design process;
- Another important aspect is given by the security problems that could arise.

Existing Infrastructure

To develop a desired architecture of our proposed environment, we need to use a proper IT infrastructure. Basically, a new (virtual) network is created over the existing interconnected computer networks [5, 6].

In this network of heterogeneous computing systems, one is a server designated by consensus and put at the disposal by the academic part (Figure 1).

For the design process, the application packages provided by the university are the following: ProENGINEER, Solid Edge, I-DEAS, Unigraphics, CATIA, AutoCAD, by means of the Centre for Research and Implementation of Numerical Methods (CCIMN), through a financing from the European Union of 40,000 EUR. CCIMN is the only center in the Sibiu area acknowledged as Training Centre in the application packages Edge, Unigraphics and I-DEAS.

The software components of the infrastructure allow the saving of drawings according to STEP AP203/214 standard (see [13] and [14]) in a joint database from which the team members can access these drawings and process them further by converting STEP format into the native environment that uses. Also, "guest" members (such as students) have access to the realized models and can visualize these drawings in VRL format.

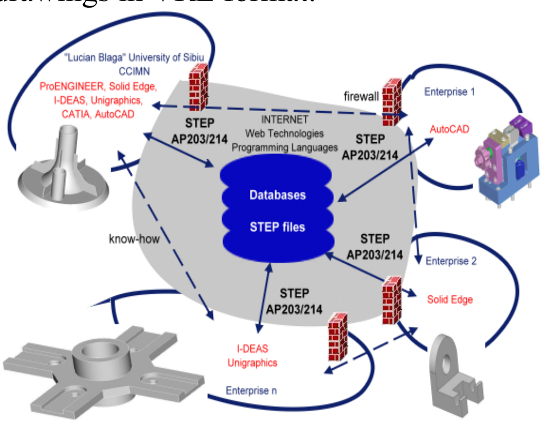

Figure 1. General IT Infrastructure.

Using Modelling Languages

The process modeling is based on UML languages, because of the following reasons (more details are provided by [7], [11] and [12]):

- UML offers a simple and expressive visual modeling language;
- UML gives extensible and specialized mechanisms, that allow the extension of basic concepts;
- UML is independent from hardware and/or software platforms, programming languages and development processes or methods;
- UML suggests a formal base for the understanding of modeling languages;
- UML is well implemented by the actual object-oriented instruments of the market;
- UML gives support for expressing high level development concepts, comprising the state of the art methods and techniques;
- The designers can use a rich palette of UML applications, frameworks, and components.

A detailed multi-attribute analysis for the choice of the best suited modeling language from the multitude of solutions offered by the market (such as CIMOSA, IDEF0, IDEF1X, IDEF3, GRAI/GIM, IEM, Petri Networks, OOA, OMT, EXPRESS, ARIS, and NIAM) is given in [7].

Storage Solutions

With regard to the structure of a STEP file type, a database is designed. This database includes primitives and geometrical models, which in the end will be used in the modeling application. Of course, the specialist will use a series of predefined or previously created objects.

General Characteristics The STEP data structure is very simple. In essence, a number of constructors are used (these constructors will be presented below).

The constructors are usually well-known geometrical shapes. The coordinates or the geometrical constructor's position is also well known. The position in the geometrical plane of these shapes can be absolute or relative to a position of another entity which was previously determined. This connection (through reference) generates a structure, a network of points that can act in the following as a system, if translations and/or rotations are desired (see [17] and [18]).

The main STEP entities (primitives) are the following: *Cartesian Point*, *Line*, *Circle*, *Ellipse*, *PolyLine*, *Composite Curve*, *Trimmed Curve*, *Plane*, *Cylindrical Surface*, *Conical Surface*, *Spherical Surface*, *Surface of Linear Extrusion*, *Rectangular Trimmed Surface*, *Curve Bounded Surface*, *Offset Surface*, and *Shell Based Surface Model*.

In order to render on a screen or plotter such geometrical information, different members of the design team can use standardized Web languages like VRML (*Virtual Reality Modeling Language*) and its successor X3D (*Extensible Three Dimensions*) [19] or SVG (*Scalable Vector Graphics*) [16]. Three-dimensional STEP primitives can be expressed by VRML/X3D constructs and two-dimensional primitives can be denoted by several SVG elements.

Data Structure Using the above ideas, an object database can be created, that can be included within a system [4]. We propose the following database schema to be used. The structure of the database's tables is presented below.

geometrical_shapes – stores existing geometrical shapes
Id, int : unique identifier in the system,
Name, char() : name of the shape
Indate, date : insertion date
Owner, int : user who created/inserted it
Status, int : finished, unfinished, under construction

shape_specifications – offers information regarding the shape coordinates and other useful data
Id, int : unique number for identifying the line
Id_shape : identifier of the shape to which it belongs
Order, int : order of the line inside the shape
Data, text : set of coordinates (here, according to needs, a connection table can be implemented – see OBS, below)
Type : type of the STEP entity

OBS: shape_specifications_data
Id_spec, int : id specifications
Order, int : order
Param1, text : parameter 1
…
paramN, text :parameter N

Taking into account the above-described model, it is easy to make operations such as conversions to other formats, because this table structure satisfies the requirements of a coherent run-through (the *order* attribute).

This model is proposed for the storage of primitives. For storing composed models (bodies composed of objects already existing in the database), we can elaborate a system of storage similar to the one developed above. This is an interesting problem from the point of view of model unification. There should be no major differences with regard to a primitive or a composed model. In order to satisfy this problem, an additional data structure is suggested:

ConnectionModelPrimitive
Id_shape : shape identifier (primitive from the composition of the model)
Id_model : model identifier

Applying such a simple structure, several unity-related problems can be solved, at least from the point of view of the processing software. In this way, a model is seen in the same way as a primitive.

Converting Data into XML Documents To assure the flexibility of the design environment, we also propose an XML-based representation of the above presented data structures.

The *Extensible Markup Language* (XML) [1] language is a recommendation of the World Wide Web Consortium for a meta-language to define mark-ups (annotations) for content publishing particularly on the World Wide Web space.

The main objective of the XML meta-language is to provide some benefits not available in HTML (*HyperText Markup Language*), such as arbitrary extensions of a document's elements

(tags) and their attributes, support for documents with complex structure, and validation of document structure with respect to an optional document-structure grammar: DTD (*Document Type Definition*) [1] or XML Schema [20].
As a standard recommended by the Web Consortium, XML is considered as the data format for information interchanging between a variety of Internet and Web applications. The XML attractiveness is primarily due to its flexibility in the representation of many data types (see below). The uses of mark-ups give to the XML language the opportunity of self-description, and its extensible nature makes possible the definition of new document types, with a particular destination (e.g., user profiles, business rules, multimedia, data-flow etc.).

In the case of XML, exchanged data structure is preserved. One of the key advantages is that the data can be organized as in an object-oriented database. As XML is format-independent, there is possible to generate multiple – XHTML, SMIL, WML or XUL – outputs easily by transforming XML documents via XSL (*Extensible Stylesheet Language*) constructs [10, 20]. XML has grown into a large family of standards integrating key technologies from three previously independent domains: documents, databases, and the Internet.

It is easily to map the existing data stored into the database suggested in previous section into certain XML documents. This will assure the flexibility of the proposed environment, because the persons involved into the design process can view, transform and/or exchange desired data without problems of compatibility between existing software tools and applications. UML diagrams, STEP-based drawings, and other visual materials (e.g., figures, schemas, sketches, etc.) can be converted – via XSL stylesheets – into JPEG or PNG images or into PDF documents. Also, UML data can be stored by XMI (*XML Metadata Interchange*) [10] documents.
Furthermore, all information can be easily published on a Web portal or wiki site to be publicly used by third-party organizations or companies. Of course, different information could be accessed only within the university's intranet via regular authentication methods.

Example In the following example, we propose a straightforward structure of an XML document that can store the information of the *geometrical_shapes* table that we presented in previous section.

```
<shapes xmlns=”urn:ulbsibiu.ro:STEP:shapes”>
   <shape id=”...”>
        <name>...</name>
        <insert_date>...</insert_date>
        <owner>...</owner>
        <status>...</status>
   </shape>
</shapes>
```

Capturing Knowledge by using Semantic Web Technologies

Semantic Web Technologies The actual WWW space is primarily compounded by pages (documents that contain mark-ups) with information in the form of natural language text and multimedia intended for humans to read and to understand. Computers are principally used to render this hypermedia information, not to reason about it. Information retrieval has become omnipresent and information needs no longer to be intended for human readers only, but also for machine processing, enabling intelligent information services, personalized Web sites, and semantically empowered search engines – this is the seminal idea of Semantic Web [2, 8-9].
Semantic Web technologies are based on the XML and are structured on three main layers (see Figure 2):

- *metadata layer* offers an extensible framework in order to express simple semantic assertions (e.g., vocabularies or taxonomies); this conceptual model can be use to attach metadata (data about data) to each resource of the environment;
- *schema layer* can help to specify simple ontologies in order to define a hierarchical description of the concepts and properties for a given resource;

- *logical layer* introduces ontological languages that are capable to model complex ontologies;

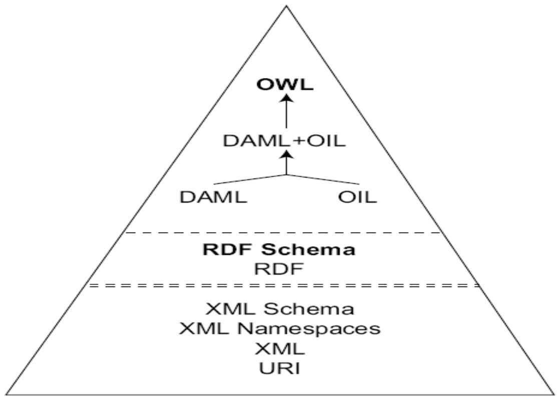

Figure 2. Actual Semantic Web languages [10]

In order to shift towards the Semantic Web, there were developed a series of XML-based languages specialized in the modeling of knowledge – for example, RDF (*Resource Description Framework*) and OWL (*Web Ontology Language*) [8-10, 20].

These languages could be the proper solution to store and process different information within the proposed environment.

Using Metadata Each component of the system can be described with the help of metadata. The metadata level is the first level of a semantic Web-based application. This metadata can be attached to each software component of the environment in order to store several important characteristics (e.g., information regarding uptime, ownership, execution platform, etc.).

Also, for each user we can retain the information about his/her status. For example, we can store the user role – administrator, database manager, security monitor, regular user (specialist or visitor) etc. Also, the system can retain personal data (e.g., age, e-mail address, location, occupation, etc.), and user-interface preferences (layout, chromatic and interaction preferences).

To associate and store metadata, we use RDF – an XML-based model for processing metadata. RDF standard provides interoperability between applications that exchange machine-understandable information on the World-Wide Web. RDF is intended to be used to capture and express the conceptual structure of information offered by the Web.

RDF metadata can also describe client's session information, such as host and user profiles, user profiles of participants of previous and current sessions, session profiles (e.g., timing, available resources, activity history, etc.), private user applications and resources (tools, plug-ins, content editors, etc.). More details in [2] and [4].

Expressing Relations between System's Components

The relationships established between the components of the system can be also expressed by RDF statements. For example, we can trace a relation of dependence between different modules of the communication environment (e.g., user authentication module, workflow monitor, designer module, etc.) and the involved resources (drawings, storage systems, format converters and others).

The relations are useful to maintain the connection between the involved design and additional applications and their users. Using FOAF (*Friend Of A Friend*) [15] statements, we can create the graph of inter-connections between the persons that participate into a task or use a certain component (such as the modeler software tool or the authentication module).

Also, for exchanging data between the components of the system, an XML-based format can be used [3].

Conclusion and Further Work

In the present domain located at the crossroads of production systems and communications, any enterprise is bound to be unsuccessful, on a world-wide market where the quality-cost-time factors become ever more demanding. The paper presented a solution based on semantic Web technologies for building a knowledge-oriented distributed production system.

After general considerations regarding the IT infrastructure and general architecture of the proposed system, the material suggested the use of UML language and STEP standard. From the authors' point of view, an interesting approach is given by the use of XML technologies to assure the platform independence and a proper exchange of data between the involved applications and users. Also, the paper gave a solution of capturing the knowledge within the environment by using semantic Web languages, especially RDF framework.

Using the proposed approach, we intend to develop – by using exclusively open-source technologies, such as Linux operating system, Apache Web server, MySQL/PostgreSQL database servers, PHP Web application server, Xindice or eXist native XML database servers, etc. – a Web portal that offer a flexible support for collaborative design between academic and industrial actors.

A further direction of research is the study of how semantic Web technologies can be effectively used in order to facilitate an intelligent communication channel between persons involved into different production activities.

Another approach is to make simulations of the enterprise's activities via a multi-agent system, such as Omega [3].

References

[1] Bray, T. *et al.* (eds.) (2004) *Extensible Markup Language (XML) 1.0*, W3C Recommendation: http://www.w3.org/TR/REC-xml.

[2] Buraga, S. C. (2004) *Semantic Web* (in Romanian), Matrix Rom.

[3] Buraga, S. C., Alboaie, S., Alboaie, L. (2005) *An XML-RDF-based Proposal to Exchange Information within a Multi-Agent System*. Proceedings of the NATO Advanced Research Workshop on Concurrent Information Processing and Computing, IOS Press.

[4] Buraga, S.C., Cioca, M. (2004) *Using XML Technologies for Information Integration within an e-Enterprise*. 7th International Conference on Development and Application Systems, Suceava.

[5] Cioca, M. (2003) *Application of Information Technologies and Communications in Mechanical Engineering: using Web Technologies*, Internet and e-CASE Instruments, 3rd International Conference "Research and development in mechanical industry", RaDMI, Herceg Novi, Montenegro Adriatic.

[6] Cioca, M., Buraga, S.C. (2003) *New Tools for Human Resource Management in e-Business: Combining UML Language*, Reference Architectures and Web Programming". Proceedings of IEEE International Conference on Industrial Informatics, IEEE Press.

[7] Cioca, M., Cioca, L. (2005) *Multi-criterion Analysis of Reference Architectures and Modeling Languages used in Production Systems Modeling*, Proceedings of 3rd IEEE International Conference on Industrial Informatics, IEEE Press.

[8] Daconta, M. *et al.* (2003) *The Semantic Web*, Wiley & Sons.

[9] Davies, J. *et al.* (eds.) (2003) *Towards the Semantic Web*, Wiley & Sons.

[10] Geroimenko, V. (2004) *Dictionary of XML Technologies and the Semantic Web*, Springer.

[11] Rutakyamirwa, N.N. (2002) *Modeling of Business Process in Distributed Environment: a Case Study of Issuance of Certificate of Occupancy in Tanzania*, Technical Report.

[12] Vernadat, F. G. (1997) *Enterprise modeling languages.* Enterprise Engineering Integration: Building International Consensus, Springer. [13] *

[13] * * *, ISO 10303-11:1994 (1994) *Industrial automation systems and integration* – Product data representation and exchange. Part 1: Overview and fundamental principles.

[14] * * *, ISO 10303-13:1995 (1995) *Industrial automation systems and integration – Product data representation and exchange.* Part 13: Architecture & Methodology Reference Manual.

[15] * * *, *Friend Of A Friend – FOAF* (2006) http://rdfweb.org/.

[16] * * *, *Scalable Vector Graphics – SVG* (2006) http://www.w3.org/TR/SVG/.

[17] * * *, *STEP* (2006) http://wheger.tripod.com/step/.

[18] * * *, *Studio Tools* (2006) http://www.alias.com/eng/support/studiotools/documentation/DataTransfer/appendix13.html.

[19] * * *, *Web 3D Consortium* (2006) http://www.w3d.org/.

[20] * * *, *World-Wide Consortium's Technical Reports* (2006): http://www.w3.org/TR/.