Collaborative work in distributed environments using web technologies and programming languages for the improvement of design processes in virtual production systems

CIOCA MARIUSII, CIOCA LUCIAN-IONELI, BURAGA SABIN-CORNELIU2
1Faculty of Engineering, 2Faculty of Computer Science
1 „Lucian Blaga” University of Sibiu, 2“A.I. Cuza” University of Iasi
Sibiu, B-dul Victoriei, no 10.
ROMANIA

Abstract: The collaborative work in distributed environments has a decisive role in the process of virtual product design in different CAD/CAM systems by successfully utilising web technologies and the programming environments, thus shortening the time needed for bringing new products on the market and the improving the transfer of knowledge between business partners.

Key-Words: Distributed environments, CAD/CAM, programming languages, web technologies

1 Introduction
The article's purpose is the presentation of a virtual informatic environment that would ensure the improvement of the cooperation and of the operative communication between the members of a team formed, for a limited time period, for the realising of a high complexity project. More concretely, it targets the virtual product design by a team from the University and partners from the industry, on the one hand for the shortening of the time needed for realising the product and on the other hand for the knowledge exchange between pluridisciplinar teams (for a better connection between the higher education and the business environment from the area of Sibiu). Also, the promotion of ISO/STEP, standards is sought and encouraged, as well as their inclusion in new instruments as support for concurrent engineering.

The application packages offered by the university are: 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 centre in the Sibiu area acknowledged as Training Centre in the application packages Edge, Unigraphics and I-DEAS.

2 System architecture
The architecture of the system for collaborative work takes into account:
• the type of process interaction function of the time: synchronous or asynchronous;
• the geographical dispersion of the virtual team's members: local or distant.

2.1 IT infrastructure
Basically [3], a new (virtual) network is created over the interconnected networks. In this network formed of several computing systems, one is a server designated by consensus and put at the disposal by the university (fig. 1).

The infrastructure allows the saving of drawings (via the STEP AP203/214 standard) [5], [6] in a joint database from which the team members can access these drawings (reverse conversion; from the STEP format in the native environment used) and process them further. The “guest” that has access to the realised models can visualise these drawings in VRL format.

3 Modelling languages
For the process modelling, it has been opted for the UML language for following reasons [2], [7], [8]:

• the offering of a simple and expressive visual modelling language;
• extensible and specialised mechanisms, that allow the extension of basic concepts;
• independence from programming languages and development processes/methods;
• the offering of a formal base for the understanding of the modelling languages;
• encouraging of the growth of the market for OO instruments;
• the supporting of high level development concepts, such as cooperations, frameworks, configurations and components, the comprising of the best methods and techniques.
A detailing of the multiattribute analysis for the choice of the best suited modelling language from the multitude of solutions offered by the market, such as: CIMOSA, IDEF0, IDEF1X, IDEF3, GRAI/GIM, IEM, Retele PETRI, OOA, OMT, EXPRESS, ARIS, NIAM is given in [2], the final result being presented in fig. 2.

4 Technical design

With regard to the structure of a STEP (STandardised Exchange of Products) type file, a database with primitives and geometrical models is built, that in the end will be used in the generation-modelling application, based on a sum, a series of predefined or previously created objects.
4.1 General characteristics

The STEP structure is very facile. In essence, a number of constructors is used (presented below).

These constructors are usually well-known geometrical shapes. The coordinates or the geometrical constructor's position is also well known, existing the possibility to indicate it as absolute value or as a reference to an entity (usually a point) which was previously determined. This connecting (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 [9], [10].

- STEP Entity
- Cartesian Point
- Line
- Circle
- Ellipse
- Parabola
- Hyperbola
- PolyLine
- Composite Curve
- Trimmed Curve
- B-spline Curve
- Plane
- Cylindrical Surface
- Conical Surface
- Spherical Surface
- Toroidal Surface
- Surface of Linear Extrusion
- Surface of Revolution
- B-spline Surface
- Rectangular Trimmed Surface
- Curve Bounded Surface
- Offset Surface
- Manifold Solid Brep
- Shell Based Surface Model

4.2 Implementation

Using the above ideas, an object database can be created, that can be composed into a system [1].

The module, the employed database structure, can be similar to the one described in the following.

Tables:

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

- **shapeSpecifications**
  - tables, coordinates
  - Id, int : unique number for identifying the line
  - Id_shape : id 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)
  - Type : Type of the Step entity

OBS: **shapeSpecifications_data**

- Id_spec, int : id specifications
- Order, int : order
- Param1, text : parameter
- .
- .
- 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 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, following structure is suggested:

Table:

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

- **shapeSpecifications**
  - tables, coordinates
  - Id, int : unique number for identifying the line
  - Id_shape : id 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)
Type: Type of the Step entity

OBS: shape_specifications_data
Id_spec, int: id of specifications
Order, int: order
Param1, text: parameter
...
paramn, text: parameter n

ConnectionModelPrimitive
Id_forma, text: shape id (primitive from the composition of the model)
Id_model, text: model id

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.

4.3 Needed tools
Conversion from STEP or another format in one that can be inserted in our database.
Conversion from the information stored in the database in a known format (wrl etc.). However, with a scripting language an instance of the object may be created that is represented by a JPEG or PNG image.

5 Conclusion
Without a pluridisciplinary approach in a modern domain located at the crossroads of production systems and communications, any enterprise is bound to fail, on a globalised market where the quality-cost-time factors become ever more demanding. Also, open-source web technologies need to be considered that dramatically reduce costs and that have assured their supremacy through flexibility, maturity and platform independence. Therefore, they are recommended in the virtual product design process using different CAD/CAM systems in distributed environments, technologies such as the Apache web server, the application server PHP, Java and the database server MySQL run on a UNIX platform [1], [4].

References: