Int J Adv Manuf Technol (2009) 42:822–830
DOI 10.1007/s00170-008-1622-2

ORIGINAL ARTICLE

A framework for collaborative product review

Gorazd Hren · Anton Jezernik

Received: 7 November 2007 / Accepted: 16 June 2008 / Published online: 16 July 2008

Abstract In a globalised industry, it has been noted that an increase of interest in collaborative virtual environments as an alternative or extension to collaborating using CAD systems. The high product variety, the "mass customisation" of products, demands a collaboration chain. Collaborative chain includes the customers, managers, designers, manufacturers and suppliers for specific partial solutions and information exchange on remote locations. Our approach is combining the modelling system with virtual prototype via Web for evaluation of products, simultaneously involving their behaviour simulations. In order to avoid repetition of time-consuming virtual model preparation, product variety is defined in single XML file covering the various configurations. The uniform intranet/internet application combining Java, XML and VRML has been developed on Thin Client–Fat Server architecture.

Keywords CAD · Virtual prototype · Web services

1 Introduction and review

The management in industry demands the decrease of product development and manufacturing time cycle. The consequences of high product variety, known as "mass customisation" of products, demands collaboration of engineers from design to manufacturing with customer, to share the ideas and solutions to evaluate products development. Typically, the development is an iterative process, changing, due to development of computer network, from local to global cooperation in order to integrate resources of knowledge and solve the conflicts caused by design specifications as early as possible in the development process. Virtual collaboration intends to solve the problem of geographical limitations for formal or informal collaboration needed to explore the ideas and solutions. The developing teams and customer could use the networking technologies and exchange their information and experience.

Computer-Aided Design (CAD) is used to create digital models and manage data for downstream processes as, bill of materials (BOM), Computer-Aided Product Preparation (CAPP) and Computer-Aided Manufacturing (CAM) for production process optimisation. Virtual Prototype (VP) refers to a computer model of a product presented in virtual environment with, ideally, all information and properties included, for the analysis and evaluation. The design solutions, operability, assembly constraints and production parameters can be controlled virtually before the physical product exists. The major obstacle in sharing design and manufacturing knowledge in collaborative production is the heterogeneity of engineering knowledge representation in Virtual Environment (VE).

In order to integrate the VE and CAD into single application Bimber et al. [4] presented a method for precise solid modelling using 3D input device to improve the interactivity of solid modelling and in Gao et al. [5], an approach is described in a semi-immersive environment to create and manipulate the solid model in CAD quality with 3D manipulators and voice commands. The approach considers only geometrical primitives and offers no possibilities for more complex geometry modelling and assembly definitions. The VE systems use the different dataflow for manipulating virtual models in real time and

G. Hren (✉) · A. Jezernik
Faculty of Mechanical Engineering, University of Maribor,
Smetanova 17,
2000 Maribor, Slovenia
e-mail: hren@uni-mb.si

converting data into the VE-suitable format is maintained differently, depending on the application as the conversion is not standardised. The idea of using translating models from CAD into polygonised mesh is not new since the CAD models require large files and only a polygonal representation allows fast rendering in real time. The creation of models for the VE still requires a high degree of manual work. The data needed differs a lot from the CAD models; for better performance, they have to be converted, adapted and supplemented with additional information, as realistic material appearance, animations and interactive properties. Today practice involves hard work of combining these features by users each time a CAD model is prepared or modified for VE. The commercially available systems for virtual collaboration are well described in Kan et al. [6], where systems as: dVISE, PIVOTAL and Deneb are analysed in the view of functionality, customizability and hardware necessity. Virtual collaboration systems are dedicated to high-tech and expensive products accompanying with substantial investments in hardware and software. Visualisation of VPs is mainly performed by developing applications based on CAD models and Applications Programming Interface (API), using standard geometry formats as IGES, STEP, STL etc., and only STEP offers a standardised way to propagate other product data, besides geometry. The drawback of such systems is that they provide a static interface. Shyamsundar and Gadh [8] proposed collaborative virtual prototyping of assemblies via web in client–server architecture. In the system, the model is visualised in polygonised representation so the modifications cannot be automatically transmitted to the users, they receive a warning. Choi et al. [13] suggested collaboration framework using the reference model to construct a product design. The reference model is introduced to avoid two major obstacles in collaboration during product design: highly iterative nature of the design process (negotiations, feedback) and limitations in modelling methodologies that are generic for specific processes and consequently lucks of common terminology for non design participants in the design chain as managers or customers.

Recent research in application development focuses on developing distributed applications. Several researchers have proposed the use of a central geometric modelling server for distributed applications. Han and Requicha [1] introduced an approach that provides transparent access to diverse solid modellers for applications in a distributed environment. Their system includes a feature-based design system, a central geometric modelling server and an automatic feature recognizer. The central geometric modelling server stores the B-rep model of a designed part. When a design change occurs, the design system processes the change to the feature recognition system.

A trend in Web applications shows the need for ability to preview the product and investigate or test its functional behaviour. In engineering domain, very often, the visualisation support is essential for evaluation. 3D virtual presentation of a product is a major step towards convincingness of the preview leading to several technical and integration issues in the production process.

VRML (virtual reality markup language) is the standard included in all geometry modellers for visualisation of VEs via Web using mesh presentation. VRML was originally not designed for engineering applications, although provides a cost-effective tool for sharing virtual models in design and manufacturing processes. It is supporting collaborative work and concurrent engineering between users on remote locations. The main problem is information loss in converting process from CAD models to VRML. VRML with external authoring interfaces (EAI) enabling the control of the contents of a browser window. Java networking capabilities and EAI can be used to construct the network application. The applications have to be very balanced to exploit the advantages from both, since the Java is procedural and VRML is event driven. Roy and Kodkani [3] and Xie et al. [7] proposed a framework repository of existing parts to be used in design process using VRML for parts visualisation. Chen et al. [9] introduced an application called Collaborative Assembly Representation for the assembly representation using STEP for linking the master assembly model and slave assembly model. It relies on VRML to view the models, with Java-based interface for an interactive behaviour of the application. Qin [12] describes a 3D simulation modelling system to support distributed machine design and to simulate machine operation via Web. The visualization and animation is performed by VRML browser and extends the accessibility of VRML assembly model. In [16], Cecil and Kanchanapiboon give the review of the virtual prototyping research in design and manufacturing process. They find VR technology very promising in reduced overall product development time and cost with improved quality. In conclusions, they emphasised the need to explore research issues in distributed virtual prototyping across heterogeneous platforms and from CAD to downstream applications. Li et al. [11] developed an Internet-based system to support collaborative and concurrent engineering design by integrating three functions: design, visualization and manufacturing analyses. VRML is used for reviewing the model as well as maintaining the important parameters and attributes of a feature. In [17], Gonzalez et al. proposed the solution for multi-body system dynamics combining STEP and XML to solve the interoperability problems between CAD/CAE/CAM systems. They found XML very promising and STEP as less usable and hard to implement. Mervyn et al. [10] employed Java RMI (Remote Method Invocation) and XML technologies to realise an interactive fixture design system. However, their works have limitations to evaluate assembly models with respect to manufacturability and assembly procedure. Hoffman and Joan-Arinyo [2] describe another approach based on client/server architecture for a product master model that unites CAD systems with downstream application processes for different views that are part of the design process. They presented a practical approach to synchronize geometric data contributed by a CAD system and data from other application programs through the creation of associations. Kim et al. [15] introduced a Web-based approach to feature-based part and assembly modelling in distributed environment using STEP PDM for mapping metadata organised in XML schema. The environment provides methods for exporting product data from the CAD system for viewing and exchanging XML messages for searching and registering.

In CAD systems, models are defined with features, relations between parts are motion driven (primarily using DOF) and can be defined easily. In the VRML models, in contrast with CAD assemblies, the constraints are generally performed by geometric calculation and sensors and correspondingly, motions defined are time- or event-driven.

The decomposition method is described by Jiang-sheng et al. [14] which divides the assembly definition into three parts: geometry, topology and assembly information, using a mate table, that are bonded in database. The issue is that all bodies are represented with surfaces and axes, although only a few are needed to define the mating conditions that become a problem with large assemblies or complex geometry. The translation method also lacks functional features and animations. In Wang and Tian [18], the VRML-based virtual assembly is proposed with synchronous moving algorithm to retrieve assembly information from basic assembly model to target assembly model. In practice it is very rare that two or more participants do the assembly specifications via Web. It is more efficient to assemble the parts locally with appropriate software and send the results. The similar method is used in CAD system to work over the net using macro programs to collaborate sending a few lines of macro program.

Today, the industry demands fast and effective development and production of products, so companies use the highest possible modular structure to enable the “mass customisation” exchange of parts and for automation of the product planning and manufacturing processes. Many times, it is not clear if another part or sub-assembly with similar functional behaviour could directly replace the existing one, or could be implemented with minimal changes in overall design. Due to the modularity structure of assemblies, very often, different configurations are tested; one element (part or sub-assembly) is swapped with another at the same position in assembly offering the same or slightly different functional behaviour. For such part exchange in the assembly, different VPs have to be configured and prepared. Highly effective management of data is needed when using VE and CAD systems because of a huge amount of data that is still missing in the industry infrastructure. The present work is an approach to combine or integrate CAD and VE systems. The target is a centralised management of all data needed for VP. Virtual environments are not very suitable for SME (small and medium enterprise) companies due to hard customization of functionality, advanced technology and human resources.

The present work describes the ongoing research on the architecture of the application addressing the sharing of VPs over the Web with standard programming tools. The proposed approach enables cooperation of broad set of participants in production process, irrespective of tools used for modelling the assembly. The approach gives the opportunity for interactive modifications in configuration from prescribed set of components, including their functionality description, investigate the results of modifications and finally reconstruct the assembly with limited features in the CAD system. The VRML is used for viewing the VPs and XML as configuration data carrier. In the system, it is possible to rebuild the different configurations via macro programming to retrieve the full assembly in the CAD systems.

2 Framework architecture

The Web-based framework provides a convenient platform for users to view and evaluate the virtual prototype. A distributed system generates models from XML-style representation to allow a Web browser to perform viewing and manipulation, leading in two main features:

1. By taking advantage of the effective utilizations of the Web and Java technologies, this system is independent of the operating system, scalable and service-oriented. The services located in the Internet can provide an effective manner for designers and customers to conduct the development process.
2. An XML-style representation has been used to carry out some features for visualisation and manipulations in the Web-based system. This format incorporates the characteristics of VRML and features to support Web applications. The XML-based information representation enables the system to be effectively adaptable to meet the new development of the Internet technology.

The current system and services are based on the Java Servlet mechanism. With the development of some new Internet integration technologies such as the Web service, it is necessary to explore new alternatives to integrate the current functions under the new system infrastructure.

The goal of the presented research work is to develop a framework: to handle viewing of assemblies, to manipulate different appearances of the parts, to enable the exchange of parts simultaneously and finally to propagate the modifications back to the CAD system. The important issue is to assure bidirectional link between CAD and VR system.

A framework is schematically presented in Fig. 1. Only standard free-ware tools are used: VRML as standard data transfer, Web browser as virtual model viewer and XML as the configuration data carrier.

The possibilities of VRML are far from immersive technologies, but a key advantage in desktop systems is that standard computer techniques could be used. VRML is neutral file format and models from most modelling applications can be directly translated to VRML. Animation can be employed to show objects behaviour over the time or with the user interactions.

In order to assure the bidirectional link between CAD and VR system, the configuration file is introduced to satisfy the dual nature: generation of the VR model for evaluation and to generate a macro program in CAD system to rebuild the assembly from existing parts in CAD database. The logical choice is the XML format. It is standard, with clear hierarchical structure and well adapted for programming on Web. The complete description of the assembly configuration with all possible variations is defined in a single XML file. With Java applet, it is possible to change the configuration and properties of the virtual environment and perform evaluations simultaneously. Since the communication is performed through an application on the server, the client is independent of the system architecture and database scheme. All the data are organised in files: geometry description, sensors, routing definitions, scripting nodes, interpolation data and material appearances with colours, textures, sounds, etc. The objects in the structure are represented with tags and the objects features with attribute pairs. The attribute name refers to the purpose and the attribute value refers to the name of the file, which contains an appropriate description. The main role in the application relies on the configuration file that is used as integrator for VE and CAD systems following XML syntax.

2.1 Configuration file definition

The properly nested objects define hierarchical structure and may contain references. With DOM parser it is possible to read the XML configuration file, check formatting and semantic validity, built and present a complex hierarchical object structure with few lines of code. The basic object is defined, that could be a part or a grouping element where the multiple choice of parts at the same level in hierarchy could be, but not necessarily, defined.

The basic object is defined with following object classification:

object $\boldsymbol{B}(\boldsymbol{b}_{id}, \boldsymbol{A}, \boldsymbol{C}(c_{id}, c_{def}))$

$\boldsymbol{b}_{id}$ unique object identifier,
$\boldsymbol{A}$ set of attributes pairs represented by names and values,
$\boldsymbol{C}$ set of object constrains.

Following the XML syntax the object has a general form:

- <**obj**$_{id}$ attributes>
- <c$_{id}$>c$_{def}$ </c$_{id}$>
- </**obj**$_{id}$>

The following are some of defined attributes:

choice attribute to select a sub-object from the list of sub-objects,
NAME attribute denotes the unique name of the object,
GEO attribute denotes geometry definition of the object,

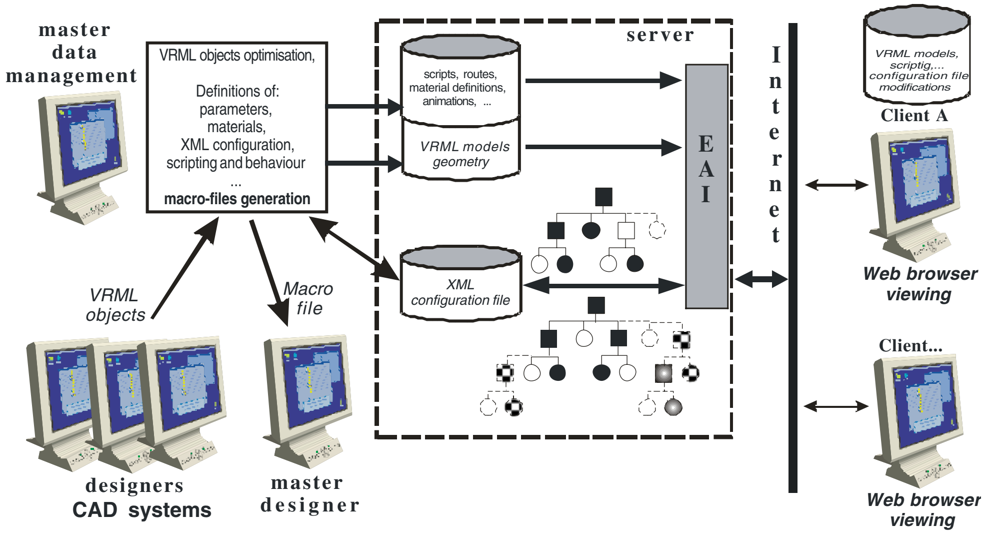


Fig. 1 Schematically presented framework

SENS attribute denotes sensor nodes,
ROUTE attribute denotes routes definition,
INLINE attribute denotes the URL and name of the remote object,
SCR attribute denotes file with corresponding script node,
USE attribute denotes using existing VRML object geometry, if the object geometry is already defined (multiple instances of VRML object geometry),
EX attribute denotes constrains for object behaviour in CAD manner.

Following the syntax of XML the *C* set includes four elements that define the object:

$\boldsymbol{C(c_{id}, c_{def})}$

c_{id} object constrain identifier (*Poz*, *App*, *Reference* and *CatConstraints*),
c_{def} object constrain definition with relationships.
Poz position of the object in VRML syntax with four entities and values as attributes pairs. The entities are in generation of VP converted from CAD description to Euler angles as used in VRML.

– < **Poz**>
– < **translation** *x="0.0" y="0.0" z="0.0"*/>
– <**rotation** *x="0.0" y="0.0" z="0.0" ang="0.0"*/>
– <**scale** *x="1.0" y="1.0" z="1.0"*/>
– <**center** *x="0.0" y="0.0" z="0.0"*/>
– **</Poz>**

App appearance of the object in VRML syntax that could include more appearance definitions and *choice* attribute to activate one of them.

– **<App choice="n">**
– **<mat$_{01}$ file="app-mat$_{01}$._"/>**
– **...**
– **<mat$_{nn}$ file="app-mat$_{nn}$._"/>**
– **</App>**

References references for assembly reconstruction and simulation generation in VRML.

– **<References>**
– **<reference r$_{id}$="ref$_{i}$ "value="val$_{i}$"/>**
– **...**
– **</References>**

CatConstraints constraint definitions for assembly reconstruction in CAD system and generation of simulations in VRML.

– **<CatConstraints>**
– **<Cons_type Cons_Att$_{id}$=value$_{i}$...Cons_Att$_{nd}$= value$_{n}$/>**
– **...**
– **</CatConstraints>**

The ***References*** and ***CatConstraints*** are described in CAD syntax and used for automatic conversion during VRML model generation to define sensor and interpolator nodes in VRML to simulate the functionality of objects. That solution enables to define the constraints in VRML syntax with no additional processing to find mating conditions, axes of rotation in VRML description as in [18]. In our approach, assembly definitions are directly processed from the CAD system definitions and converted to Euler angles of rotation as defined in the VRML language.

Further, the generation of interpolators for simulations (routing mechanism) is easier and accurate. The references and constraints are used to rebuild the CAD model with all assembly constraints in CAD system.

Often, more components share the same definitions and in order to enable single change of attributes values that could be defined as global parameters in the beginning of the configuration file.

<**Global_parameters** *d="1" open="left"*/>

Since the CAD model is often defined with parameters defining the geometry of parts or relations in assemblies, that parameters are also included into global parameters. Changing global parameters directly affect the generation of CAD assemblies and parts.

Some rules are bonded to the configuration structure:

– The object that represents a part has to have all attributes to define its existence and behaviour (at least name and geometry), all other attributes are optional.
– The parts have to be properly nested into the assembly clearly defining hierarchical structure. The depth of the structure is not limited.
– The value of the ***choice*** attribute could be an integer limited to the number of objects in the list or a string indicating that the selection is defined by global parameter. The string value gives the name of the global parameter and the value of the global parameter is comprehended.

The robustness of the VRML is welcome, since during the programs testing the errors in results are easy to find and, in cases of badly defined routing and scripting, the events are not performed.

2.2 Preparation process and data management

VRML models generated directly from CAD systems, in general, are composed from many nested, complex and repeated structures. To accelerate the rendering performance of a system, a global optimization of a polygonal data is necessary. The macro program in CATIA was coded to rearrange and optimise the VRML model:

– the geometry of parts is separated; the separated surfaces of a part is merged into single geometry file including compression of geometry data,
– additional information as appearance, view, background and navigation definitions are arranged in separate files,
– the hierarchical structure in XML is generated,

It has to be pointed out, that the configuration file is generated only for the first main assembly. For alternate sub-assemblies or parts, all actions are performed to accept configuration file. The configuration file is supplemented with additional and alternate objects. The information as appearance, sensors, interpolators, scripts and routing parameters for object functionality and dynamic behaviour definitions are added and could be interactively controlled by the user. These preparations were firstly done manually with a lot of tedious and time-consuming work, but when all objects and scripts are prepared, the task is not so complicated if the rules and definitions are obeyed. Recently, we developed the editor that allows changing the configuration more effectively and interactively. Finally, the extended database consists of geometry and material appearances of parts, additional definitions and the configuration file that links the structure of the assembly with attributes. The output of the editor could be seen in Fig. 2. The editor enables to change and add different parameters into the configuration file. The configuration objects is shown as a tree where, with different colours, the selected options are indicated. With selecting the object, the respective parameters are shown divided into three major parts:

1. general properties
   – name of the element,
   – appearance (selecting from defined appearances or define new one with all parameters e.g. colour, transparency, texture definition),
   – all sub-elements are listed (adding or removing elements),
   – adding or changing the global parameters.
2. geometrical position of the element (transformation, rotation, scaling and centre—needed for VRML sensors)
3. additional options:
   – the name of the file with geometry,
   – the element added using Inline definition (URL),
   – adding different views if needed (name of the file with the view definition),
   – additional parameters as sensors and routes also defined in files.

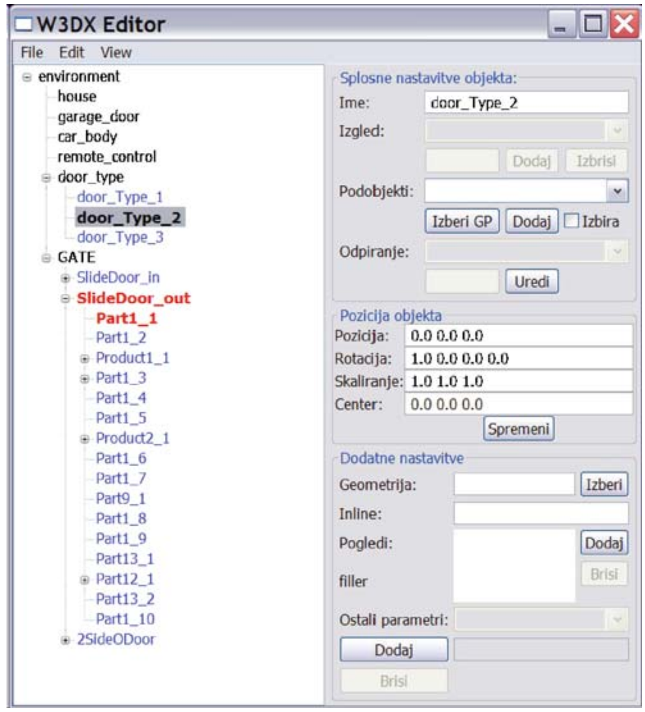


Fig. 2 The editor outlook for the case study described in next chapter

The outlook is possible to check anytime with view option in separate window with VRML browser so the preparation process is the subject of the instant evaluation. The editor has replaced extensive previous manual work needed for preparing the configuration file with several options.

3 Implementation

The following section describes the implementation of the framework on real example. The aim of the implementation on engineering domain is to test different solutions. In mechanical engineering, the detail geometry is an important matter. In Fig. 3 is the example of the remote-controlled gate solutions with the limited space problem. Three solutions were implemented; two with slide gates (inside and outside) and one with opening each half of the gate. All parts and assemblies were modelled with CATIA. For first solution, the macro program generated a configuration file, gate geometry and appearance files with assembly constraints. The alternative solutions were modelled and extracted separately and afterwards added to the structure using editor. The house, remote control device and car were included in the configuration file and positioned with the editor as *Inline* node. To simulate the behaviour, sensors and scripts nodes are linked to the remote control to open the gate. The very detailed solution is geometrically

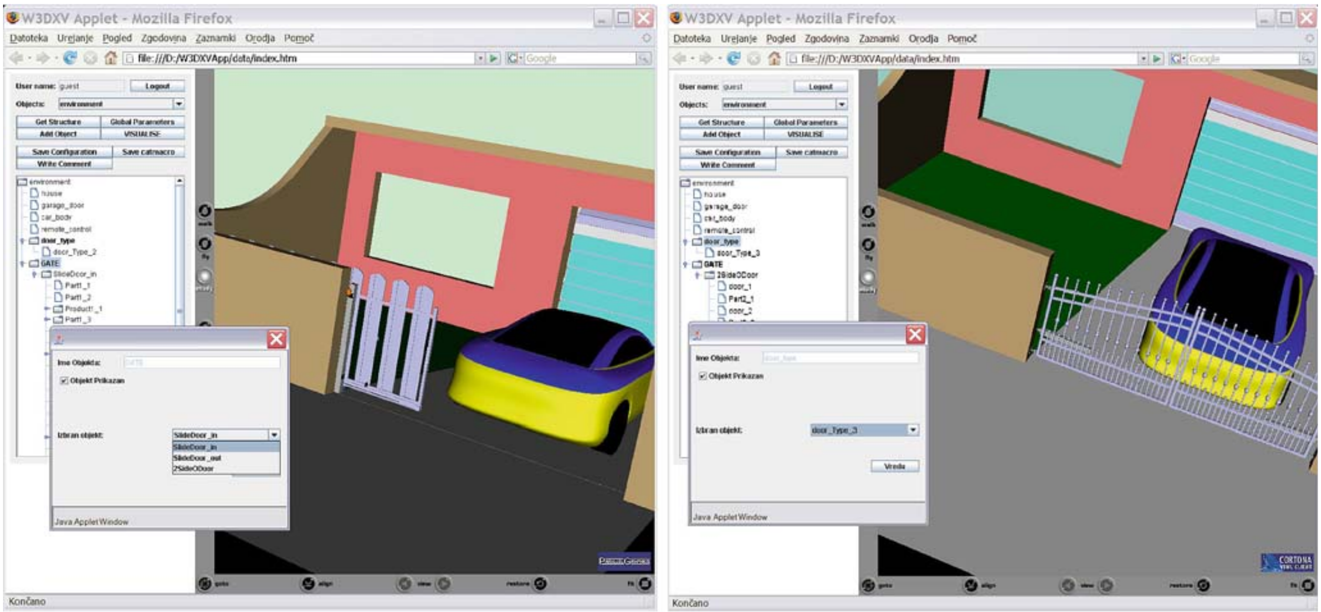


Fig. 3 The application outlook using different choices

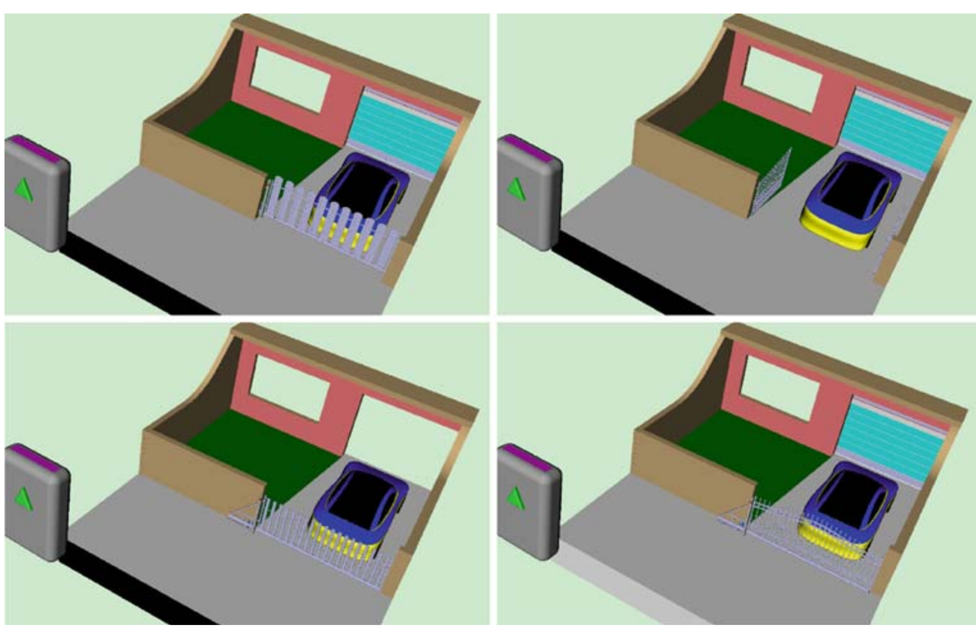
complicated and found almost inappropriate for application working over Web. The amount of data needed to transfer was very extensive. The speed and response of the application is sensitive to local computer performance.

In Fig. 3, the outlook of the framework is presented. In the left window, the user has to log-in and select the name of the configuration file to get the structure. Selecting the *VISUALISE* button, the server generates the VRML file from the configuration file and displays it in view window. The user could navigate the space and use interactivity possibilities of the browser to open and close the gate. The user could easily select one of the three given choices of opening the gate as main solutions and three types of the gate to investigate the behaviour of the selected one.

Few possibilities of choosing different configurations, with different gate types are shown in the Fig. 4. The different outlook of the gates is presented for the same solution (outside slide gate).

In Fig. 5, some details are presented showing the same type of the gates using for side opening and slide version. Fig. 5a and b is showing the opening mechanism in opened and closed position.

Fig. 4 Few possibilities in the example



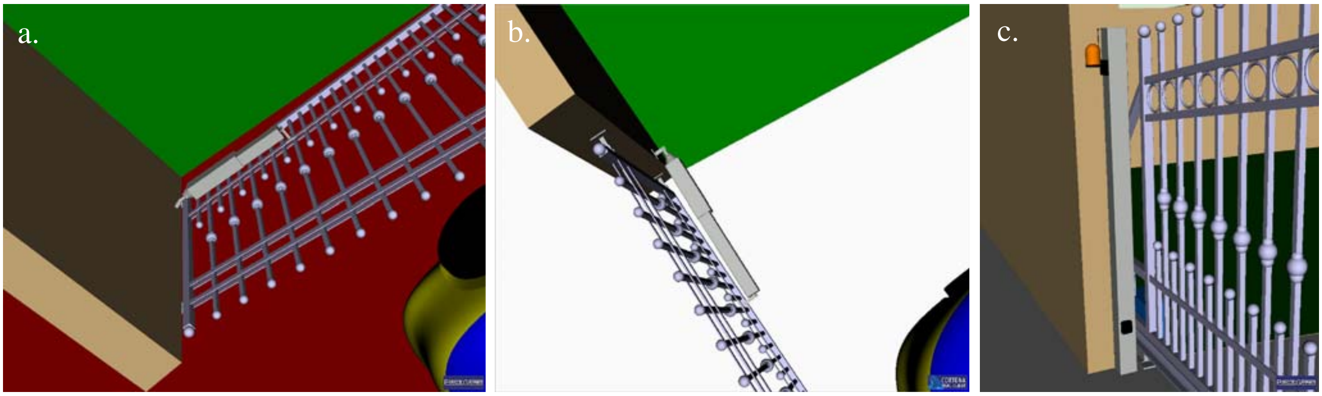


Fig. 5 Detail snapshots (**a** opened, **b** closed gates and **c** slide gate detail)

In the framework, the user has the possibility to add an object into the configuration file as an alternative to the existing objects or write the comment to the master user.

In Fig. 6, the fractions of the configuration file for the discussed example is presented. All main functions are shown: clear hierarchical structure, choice definition including the elements from remote location (inline).

Fig. 6 The fragments of configuration file

```
<object name="environment" BG="Background._">
  <poz>
  <scale x="0.0010" y="0.0010" z="0.0010"/>
  </poz>
  <object name="house" inline="http://?/vrml/house.wrl"/>
  <object name="garage_door" inline="http://?/vrml/garage_door.wrl">
    <poz>
    <translation x="-9325.685" y="212.264" z="3608.284"/>
    <rotation ang="3.142" x="0.0" y="1.0" z="0.0"/>
    </poz>
  </object>
  <object name="car_body" inline="http://?/vrml/car.wrl">
    <poz>…</poz>
  </object>
  <object name="remote_control" inline="http://?/vrml/remoteC.wrl">>
    <poz>…</poz>
  </object>
  <object choice="0" name="GATE">
    <object name="SlideDoor_in" ROUTE="SlideDoor._">
      <poz>
        <translation x="-3460.006" y="104.436" z="5236.704"/>
        <rotation ang="2.067" x="-0.596" y="-0.568" z="-0.568"/>
      </poz>
      <object name="door_type" choice="2">
… … …
    </object>
… … …
    <object name="SlideDoor_out" ROUTE="SlideDoor._">
      <poz>
        <translation x="-3022.33" y="104.436" z="5179.186"/>
        <rotation ang="2.067" x="-0.596" y="-0.568" z="-0.568"/>
      </poz>
      <object name="door_type" choice="1">
        <object name="door_Type_1" GEO="GEO_dT1._" ROUTE="R_door._"/>
        <object name="door_Type_2" GEO="GEO_dT2._" ROUTE="R_door._"/>
        <object name="door_Type_3">
        <object name="dT3_left" GEO="GEO_dT3._" ROUTE="R_dT3Left._"/>
        <object name="dT3_right" GEO="GEO_dT3._" ROUTE="R_dT3Right._">
          <poz>
            <translation x="-1890.006" y="0" z="0"/>
            <rotation ang="3.142" x="0.0" y="1.0" z="0.0"/>
          </poz>
        </object>
      </object>
… … …
      <object name="2SideODoor" ROUTE="2SideODoor._" >
      <object name="door_type" choice="2">
… … …
      </object>
… … …
      </object>
… … …
  </object>
</object>
```

The selected configuration is saved in user folder with time and user name as a selected version. The different options could be shown to the customer on remote location and accepted solution can be forwarded directly to CAD system.

The master designer has the authority to run the macro program in CATIA to rebuild assembly from the configuration file. The next step is to generate all documentation (drawings) and BOM for production purpose that is automatically obtained from CATIA. The example demonstrates the effectiveness and usability of the framework in collaborative work using Web.

4 Conclusions

Collaborative design chain includes the customer, designers, manufacturers and suppliers for specific partial solutions. The concurrent methodology can be implemented when the internet-based technologies link all participants with continuous data and information exchange on remote locations.

In contrast to an expensive physical prototype for the product design and performance verification the virtual prototype offers evaluation in the digital world. The digital mock-up is not just for a graphic visualization of the assembly, but also includes the behaviour of response to user's interaction.

In engineering, the visualisation is of crucial importance. Usage of standard Web technologies leads to easier, effective and more general applications. In presented approach, the VRML is used for sharing models over the Web introducing XML as structure data carrier and integrator. The framework enables generation of VP for viewing and evaluation of the functionality in the standard VRML browser. We have to consider the consistency of databases, which is ensured since the programs perform almost no arithmetical operations. To implement LOD would be logical choice to accelerate the application response. Introducing LOD is reasonable when the assembly components are designed in top-down methodology, when during the design process the parts are refining form concept (available space) to detail and milestone solutions could be used as levels of the geometry description. The framework is cost effective since the required infrastructure exists and the viewing software is available. Today's limitations are dictated by network capabilities (download times for large VRML files describing complex virtual models) and the speed of the user's local computer (responsible for real-time rendering and interactions).

This approach is presenting a solution of virtual prototype exchange over the Web using standard free-ware tools that makes it very suitable for SMEs with limited resources.

Acknowledgements The research work is partly supported by the Slovenian Ministry of Science and Technology and Adria Mobil Company (grant no. L2-6525-0975).

References

1. Han JH, Requicha AAG (1998) Modeler-independent feature recognition in a distributed environment. Computer-Aided Des 30:453–463 doi:10.1016/S0010-4485(97)00097-3
2. Hoffman CM, Joan-Arinyo R (1998) On user-defined features. Computer-Aided Des 30:321–332 doi:10.1016/S0010-4485(97)00048-1
3. Roy U, Kodkani SS (1999) Product modelling within the framework of the World Wide Web. IIE Trans 31:667–677
4. Bimber O, Encarnação LM, Stork A (2000) A multi-layered architecture for sketch-based interaction within virtual environments. Comput Graph 24:851–867 doi:10.1016/S0097-8493(00)00088-1
5. Gao S, Wan H, Peng Q (2000) An approach to solid modeling in a semi-immersive virtual environment. Comput Graph 24:191–202 doi:10.1016/S0097-8493(99)00154-5
6. Kan HY, Duffy VG, Su C (2001) An Internet virtual reality collaborative environment for effective product design. Comput Ind 45:197–213 doi:10.1016/S0166-3615(01)00093-8
7. Xie SQ, Tu PL, Aitchison D, Dunlop R, Zhou ZD (2001) A WWW-based integrated product development platform for sheet metal parts intelligent concurrent design and manufacturing. Int J Prod Res 39:3829–3852 doi:10.1080/00207540110072290
8. Shyamsundar N, Gadh R (2002) Collaborative virtual prototyping of product assemblies over the Internet. Computer-Aided Des 34:755–768 doi:10.1016/S0010-4485(01)00204-4
9. Chen L, Song Z, Feng L (2004) Internet-enabled real-time collaborative assembly modeling via an e-Assembly system: status and promise. Computer-Aided Des 36:835–847 doi:10.1016/j.cad.2003.09.010
10. Mervyn F, SenthilKumar A, Bok SH, Nee AYC (2004) Developing distributed applications for integrated product and process design. Computer-Aided Des 36:679–689
11. Li WD, Fuh JYH, Wong YS (2004) An Internet-enabled integrated system for co-design and concurrent engineering. Comput Ind 55:87–103 doi:10.1016/j.compind.2003.10.010
12. Qin SF, Harrison R, West AA, Wright DK (2004) Development of a novel 3D simulation modelling system for distributed manufacturing. Computers in Industry 54:69–81
13. Choi Y, Kim K, Kim C (2005) A design chain collaboration framework using reference models. Int J Adv Manuf Technol 26:183–190
14. Jiang-Sheng L, Ying-Xue Y, Pahlovy SA, Jian-Guang L (2006) A novel data decomposition and information translation method from CAD system to virtual assembly application. Int J Adv Manuf Technol 28:395–402
15. Kim H, Kim H-S, Lee J-H, Jung J-M, Lee JY, Do NC (2006) A framework for sharing product information across enterprises. Int J Adv Manuf Technol 27:610–618
16. Cecil J, Kanchanapiboon A (2007) Virtual engineering approaches in product and process design. Int J Adv Manuf Technol 31:846–856
17. Gonzalez M, Gonzalez F, Luaces A, Cuadrado J (2007) Interoperability and neutral data formats in multibody system simulation. Multibody System Dynamics 18:59–72
18. Wang QY, Tian L (2007) A systematic approach for 3D VRML model-based assembly in Web-based product design. Int J Adv Manuf Technol 33:819–836