Middleware-based Integration of Multiple CAD and PDM Systems into Virtual Reality Environment

SeokRyul Kim[1] and Daniel Weissmann[2]

[1]Institute for Graphic Interfaces, srkim@igi.re.kr
[2]Institute for Graphic Interfaces, dawei@igi.re.kr

ABSTRACT

The data exchange between CAD and VR systems is recognized as an essential element for the integration of the digital design process in the manufacturing industry. This paper introduces the MEMPHIS system, a data exchange middleware, that closes the manual gap of data processing between CAD and VR environments. This system was designed to integrate PDM, CAD and VR systems through common interfaces which are based on international standards. Taking advantage of these interfaces and standardized data structures, MEMPHIS will allow the efficient reuse of VR specific modifications and eventually reduce time and repeated efforts during the design process of a product.

Keywords: PDM, Virtual Reality, Middleware, Data Exchange.

1. INTRODUCTION

In the manufacturing industry, various solutions are used to enhance the collaboration between different departments and companies during the production process. Most common examples for these systems are Computer Aided Design systems (CAD), Virtual Reality systems (VR) and Product Data Management systems (PDM). These solutions are implemented in most of the manufacturing enterprises and are target of the technology described in this paper.

A CAD system enables the design of physical products in a virtual three-dimensional (3D) spatial environment. The result data of CAD systems are used e.g. for planning and performing the production and for analyzing properties or behaviors of the physical products.

PDM systems are complex environments used to manage all data related to the engineering process. This includes the geometrical representation as a 3D model and the meta-data. The meta-data consists of information for product structure, registration, work flow, materials, costs and the production process.

A VR application uses three-dimensional visualizations of the data and adds technologies for exploring and interacting with the virtual models. This data represents the physical product and allows applications such as motion characteristic verification, virtual prototyping and virtual design review without the need of a physical model. Depending on the purpose, rendering features are added to raise the visualization quality of the virtual scene. This can include light settings, materials with reflections and refractions or animated components. VR models are converted from models in a CAD file format using external file conversion tools.

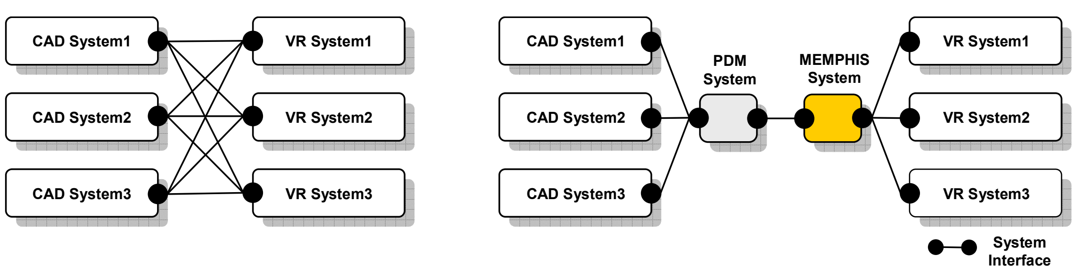
Usually PDM systems can not manage meta-data related to the VR files (e.g. light settings, materials, textures, behaviors). They lack functionalities to maintain links between the meta-data and the geometric models inside the VR files. Thus, the data exchange between PDM (or CAD) and VR systems concentrates on geometrical data. VR-specific information has to be added manually each time the CAD models are converted to VR data.

The described systems are used in the manufacturing industry to shorten the lead time for developing a product. The data exchange between them is essential as many departments and companies can be involved in the design and manufacturing process. Most popular fields of research are the integration between CAD and PDM systems and the data exchange between CAD and VR systems. The first has been carried out with active research and commercial developments by major PDM development corporations (e.g. UGS, PTC and Dassault System). The latter field has not been developed actively. Only a few researches have been performed until now.

Researching on the data exchange between CAD and VR systems, Corseuil et al. presented in [2] the ENVIRON (Environment for VIRtual Objects Navigation) system as a CAD to VR data exchange tool. In their paper, they pointed out the critical problem of CAD data's file size which is too large and complex to be visualized in real time VR environments. They presented the ENVIRON system as an effective CAD to VR conversion tool for direct data exchange. Von Lukas, M. and Vahl, M. [6] suggested an integrative approach to join the still separate application areas

of CAD and Virtual Reality. To implement common interfaces for the data exchange environment, they used OMG's CAD Services, an international standard. Rabätje, R. [8] presented the possibility to manipulate CAD data in a virtual environment and to integrate these data into a conventional CAD program for further elaboration. Gomes de Sá, A. and Zachmann, G. [4] presented several interaction paradigms and functionalities which a VR system must implement in order to be suitable for that application area. They aim on enabling inexperienced users to work with virtual prototypes in an immersive environment and help them experiment efficiently with CAD data.
Above researches concentrate on the exchange of detailed geometry data between CAD and VR systems. Hence the main problem is related to exchangeability of product meta-data. As the research topics are only handling geometry data exchange, other fundamental information such as material of products, structural information etc. from the original CAD data are not incorporated in the VR environment. The inclusion of those additional information, important for the visualization in VR, are not within the scope of the topics and would still have to be handled manually.
Further obstacles lie in the use of heterogeneous environments inside and between companies. This prevents easy exchange between multiple systems. The variety of software solutions, data structures and file formats complicate the access or even the synchronization of all data. The communication between all systems involved has to be managed via interfaces that connect only two systems directly at the time.

To solve the above problems, we will introduce MEMPHIS (Middleware for Exchanging Machinery and Product Data in Highly Immersive Systems) as a data exchange middleware that will allow centralized communication between multiple CAD and VR systems via a PDM system. It consists of a server implementation accessed by a client. Other systems (e.g. PDM and VR systems) will be attached by adapters. Using this system, the communication and the exchange of data will be unified. In form of the adapters, it will provide common interfaces that rely on international standards for PDM product structures and the transmission of data (PLM Services) and enable a centralized integration of multiple PDM systems (server) from any point (clients).
MEMPHIS consists not only of interfaces and communication services. A database also representing the product structure introduced by PLM Services is part of the system. This will allow the tracking of meta-data and its connection to geometrical data by storing and maintaining these information. Even after changes were made to the 3D model, the VR-specific meta-data will automatically be re-inserted to the VR file. The problem of repeated manual efforts for this integration is supposed to be solved.
Finally, MEMPHIS will be expandable with services related to the process of VR-file creation and enhancement. This includes e.g. the conversion from CAD formats to VR formats or methods for geometrical healing. Depending on the need of the company more services can be added to MEMPHIS. On the side of virtual reality systems, third party implementations can use an interface on the client side to access the services and data directly. So, based on MEMPHIS, more complex applications are encouraged to simplify the work flow between PDM/CAD systems and VR environments.



(a) (b)

Fig. 1. System interface method between CAD and VR systems:
(a) a common data exchange method and (b) the proposed data exchange method.

Fig. 1 shows a simple comparison between common data exchange methods and the proposed data exchange method within MEMPHIS. The first figure (Fig. 1 (a)) shows the direct data exchange between CAD and VR systems. This method leads to problems as it requires specialized interfaces for the data exchange between each system. The second figure (Fig. 1 (b)) shows the proposed data exchange method which requires only one PDM interface to access multiple CAD systems. The access for the CAD systems is enabled by commercial PDM CAD integrations.

In chapter 2 we will give a short introduction to the technologies related to our system before we describe MEMPHIS, its components and their functionalities more detailed in chapter 3.

2. APPLICATIONS OF RELATED INTERNATIONAL STANDARDS AND TECHNOLOGIES

As MEMPHIS aims on integrating a wide range of systems related to the process of designing and planning industrial products, high values were set on utilizing international standards for organizing and exchanging all data. Thus the interface for exposing PDM and CAD structures is based on PLM Services while the communication relies on the web service technology for easy extensibility.

2.1 Web-Service Technology

A Web-Service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine processable format (specifically WSDL (Web Services Description Language)). Other systems interact with the Web-Service in a manner prescribed by its description using SOAP (Simple Object Access Protocol) messages, typically conveyed using HTTP (Hyper Text Transfer Protocol) with XML (eXtensible Markup Language), serialization in conjunction with other Web-related standards ([9], [11]) (Fig. 2).
We used the Web-Service technology within the MEMPHIS system to implement the client/server communication in a globally accessible manner. By introducing the Web-Service communication, we could implement an efficient communication protocol based on RPC (Remote Procedure Call) techniques. As the technology is based on HTTP, network boundaries between organizations (firewalls), do not hinder the communication.

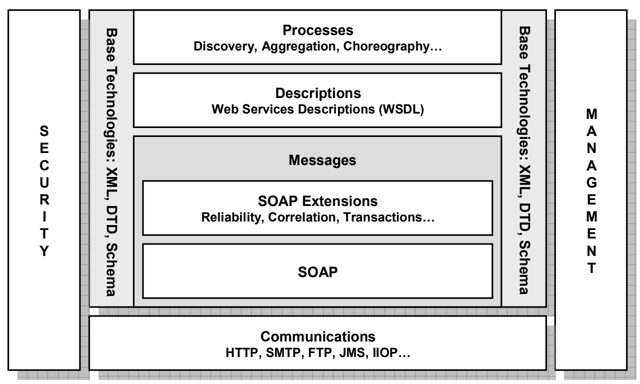


Fig. 2. Web-Service architecture stack (from W3C).

2.2 OMG PLM Services

The PLM Services is an Object Management Group (OMG) specification for the exchange of product lifecycle data with Web-Service technologies and it has been developed by the Extended Product Data Integration (XPDI) taskforce of the ProSTEP iViP association in April 2004. The information model of the PLM Services is completely compliant to the STEP (STandard for the Exchange of Product model data) PDM Schema. It is further extended by relevant subsets of STEP (ISO 10303-214:2000) and here especially the configuration management modeling parts according to CC8. The selected scope of the information model is chosen based on the requirement analysis in the PDTnet project. PLM Services is platform independent. It was designed to implement, operate and support online access and batch operation using several international standards such as STEP, XML, UML (Unified Modeling Language), SOAP and WSDL ([5], [7]).

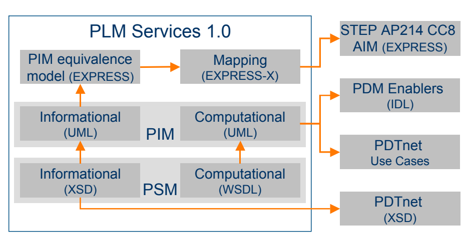


Fig. 3. Structure and sources of PLM Services 1.0 (from ZGDV).

In the MEMPHIS system, these standards fulfill central roles such as providing the structure of the common interface for data exchange and data management. To apply the standards, the database, the client and the server were designed based on the PLM Services. Fig. 3 below shows the structure and sources of PLM Services as presented in [5].

3. THE MEMPHIS SYSTEM

The MEMPHIS system takes care of the communication and data management within the design workflow of a company. Relying on web service technology and OMG PLM Services, Virtual Reality data (VR data) no longer has to be stored and processed only locally on one computer. The VR data, which is needed to present 3D models of future products in real-time and high quality, can be produced directly from CAD and PDM systems. VR systems can be initiated using the original CAD sources that are used to plan and build the product.
As a centralized middleware technology, MEMPHIS is specialized on the consistent management and the reusability of VR information related to the CAD data. Thus, it allows centralized conversions from CAD to VR formats, the management and manipulation of the semantic product structure and the extension of this structure with VR specific information. This information is reusable even after subsequent conversions, as the MEMPHIS system keeps track of changes and attachments in the data structure. Furthermore, the efforts for the users to create the virtual scenes are widely reduced, as most of the tasks can be automatically or semi-automatically accomplished by the MEMPHIS system.

The MEMPHIS system consists of several sub-components. Each component has a relationship with other systems or components e.g. PDM systems, VR systems, the data base (DB), the CAD2VR Converter or the Security Component. The main components used in the MEMPHIS system are listed in table 1 and will be described more detailed in the following sub-chapters.
Every relationship between components or to other systems are accomplished by adapters or internal connectors. Adapters are used to make a connection to external systems such as PDM systems or VR systems. They are utilizing two kinds of methods. The first method is to include a sub-application using batch calls to start the related functionalities. The second method is the usage of public libraries which can be integrated into the system on the source level. Internal connectors are used to make a connection with internal components such as the DB, the CAD2VR Converter or the Security Component. These internal connectors are embodied as source based components, as their functionalities are integrated as class libraries. This paper will concentrate on the adapters, as the processes, relevant for the core functions, are not related to the functionalities of the internal connectors. The adapters have a key role for the data exchange between PDM and VR systems.

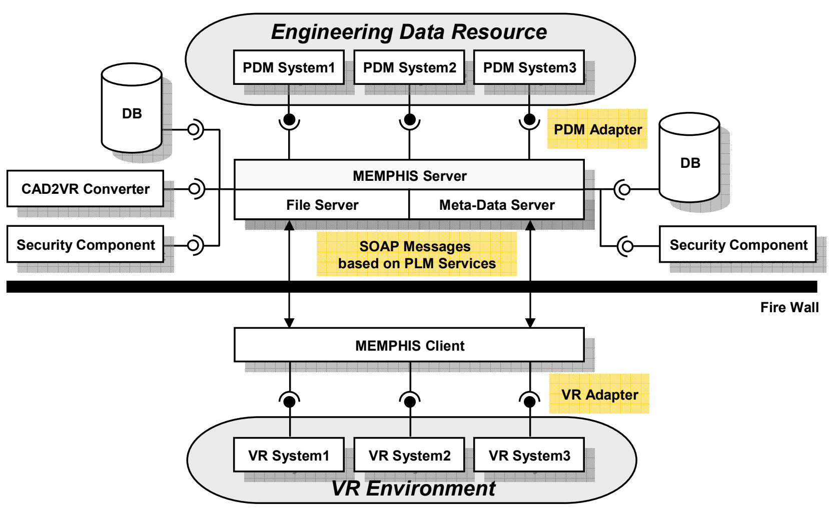


Fig. 4. Overall architecture of the MEMPHIS system.

Component	Task
Meta-Data Server	Management of stored product meta-data
File Server	Management of file data such as CAD and VR data
MEMPHIS Client	Serialize and de-serialize communication from and to the server
PDM Adapter	Data exchange component between the MEMPHIS system and PDM systems
VR Adapter	Data exchange component between the MEMPHIS system and VR systems
CAD2VR Converter	Data conversion component from CAD to VR data
Security Component	Security application of managed data in the MEMPHIS system

Tab. 1. Component list of the MEMPHIS system.

3.1 Sub-components

3.1.1 Meta-Data Server

The Meta-Data Server is managing the meta-data of a product, e.g. the Bill of Material (BOM), level of detail for the product model data (LOD) as well as other VR related property data (light, material, behaviours etc.). This server represents the major difference of the MEMPHIS data management compared to other engineering data management servers like PDM systems or drawing management systems. It enables the parallel management for CAD and VR data. Generally, most of the management systems take care of CAD data as the most important information. Although they are able to manage VR data, these systems can not handle various VR specific information such as LOD, textures, shaders, etc. For this reason, they can not be used sufficiently within VR environments. File formats usable for VR have to be converted manually from the CAD files. VR specific information has to be added after each conversion and can not be reused.

The Meta-Data Server can manage CAD and VR data more equally than other engineering management systems as they are integrated into one data structure. All data are stored and managed as Data Management Containers (DMC) based on the PLM Services specification (Fig. 5a). A DMC can be divided into six data management layers: Person, Organization, Project, Item, Document and Property. Internally all data layers are linked as a tree structure (Fig. 5b) and each data layer has a 1:N relationship for the related data layer. With these containers the complete structure of a product and its relations can be represented. This includes arbitrary meta-data, attached to the geometrical information as properties.

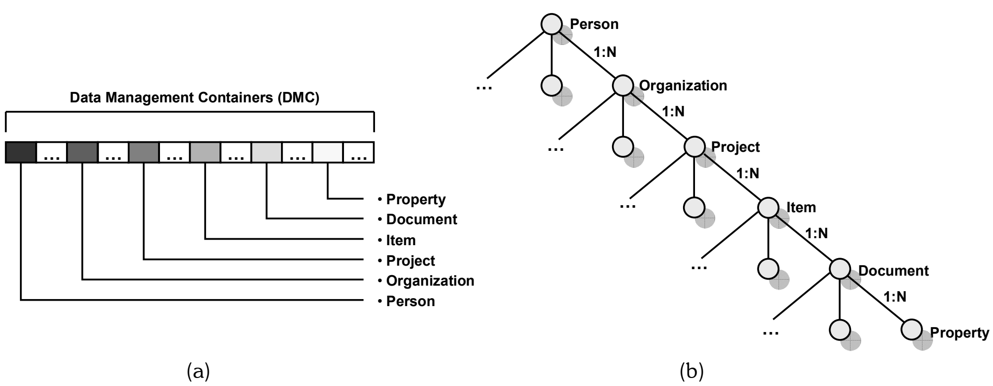


Fig. 5. Data management in the Meta-Data Server:
(a) A data management container and (b) data management layers.

Due to this shared data structure, requests for geometrical data can be connected with requests for the VR-specific meta-data. Both sources can be integrated into one output file. The underlying data base, which represents the structure, is also used to allow keeping track of changes. If a geometry item is modified or the product structure itself is changed, the concerning properties and information stay connected to its logical location. Unique IDs allow the recognition of manipulations. Thus, by reading out the information from the data base, requests for VR data will result in correctly formed files including all expected meta-data.

3.1.2 File Server

This File Server is implemented as a File Data Management (FDM) system. Within the FDM various physical file information, for example file size, file type, file creation date, etc., are stored. Additionally, the File Server can operate the integrated CAD2VR Converter (see 3.1.5). The CAD2VR Converter is used to convert CAD files to VR files such as VRML (Virtual Reality Modeling Language) and X3D (eXtensible 3D). Thus the MEMPHIS Client can perform remote file conversion by handling the File Server from the client side (Fig. 6). The benefit of this server lies in the centralized handling of the files (VR, CAD) even if no data structure is provided or needed. That allows the use of the MEMPHIS functionalities and its services for single files.

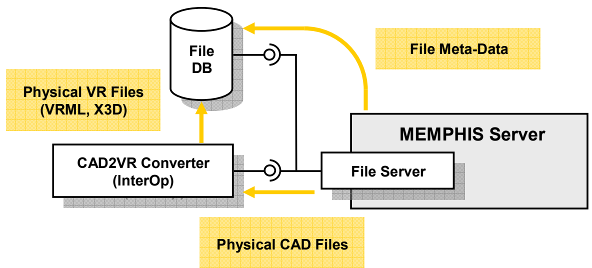


Fig. 6. File Server in the MEMPHIS system.

3.1.3 MEMPHIS Client & VR Adapter

The purpose of the MEMPHIS Client is to provide a well defined interface to the system's functionalities for the user. By restricting the access to the MEMPHIS data exclusively through the client implementation, a controlled data exchange can be guaranteed. This control includes authorization checks of the user and secure data wrapping of the exposed information. This also leads to a more convenient operation for the user who has not to take care of internal data type matching, connection protocols etc.

The MEMPHIS Client's main features are:

1.) Secure exposing of data by wrapping of the internal data structure,
2.) Simplified data structure management and manipulation using functions of the client, including
    a. Managed object structure,
    b. Functions for loading data from and writing data to server,
    c. Creating new and editing existing data,
    d. Manipulation of the data's tree structure;
3.) Hiding the complex data structure of the MEMPHIS database,
4.) Integrated authorization check including login, allowing requests to server only to authorized users.

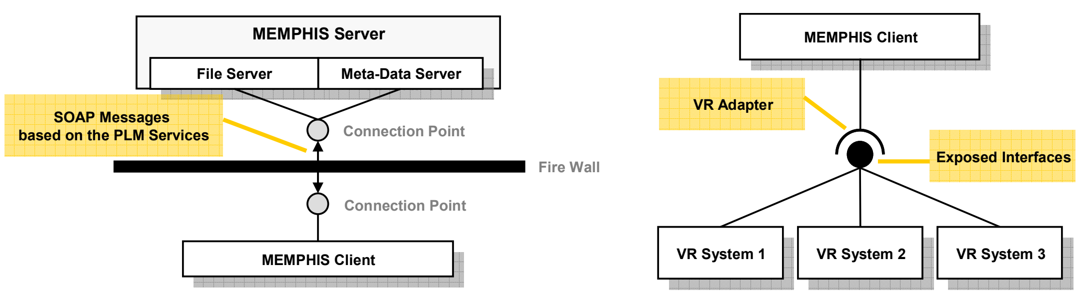


Fig. 7. MEMPHIS Client and VR Adapter:
(a) MEMPHIS Client in the MEMPHIS system and (b) Detailed VR Adapter.

The implementation features a stand-alone version of the client, providing a user-friendly graphical interface for managing and manipulating information from the server. Further, a library version (VR Adapter) of the client can be used to integrate the MEMPHIS Client interface inside of other applications (e.g. VR systems) giving direct access to its functionalities.

Using the VR Adapter instead of the stand-alone client offers extended flexibility especially for VR applications. The featured data structure allows the integration of arbitrary properties into the semantic structure of the virtual models. This extends the scope of CAD properties and enables the addition of attributes such as 3D shaders, events, animation sequences or behaviors. As these information are stored on the server side and tracked in case of changes in the structure or geometrical information, the VR Adapter leads to a high degree of reusability of additional VR data. This data is not supported in CAD or PDM systems and commonly has to be added manually after each changes made in the original data.

3.1.4 PDM Adapter

The CAD data made within many different commercial CAD systems is stored and managed inside PDM systems and must be transferred to the MEMPHIS system. To do that, there are strong needs to develop an integrator between diverse PDM systems and the MEMPHIS system. For this purpose the PDM Adapter was developed. Particularly, it is composed of the PDM Connector and the MEMPHIS Connector. These sub-components are developed conform to the PLM Services specification. The PDM Connector is directly connecting to the PDM system. Usually, most of the commercial PDM systems (e.g. TeamCenter, ENOVIA, WindChill, etc) are providing their own API (Application Programming Interface) to customize their provided services from the PDM server ([10]).
To connect to a commercial PDM system, this APIs are used. The objects and methods of each of them are mapped to the XML elements defined in the PLM Services specification to implement a common interface for the PDM data (Fig. 8). In this context, common interface means a sharable neutral file between PDM and the MEMPHIS system. After the creation of the common interface, the MEMPHIS Connector can be used for data parsing and data transmission to the MEMPHIS server via SOAP messages. For a newly integrated PDM system only the Connector to its API has to be implemented. The MEMPHIS Connector can be used for all connected systems.
To this point, the description of the data flow covers only the direction from PDM to the MEMPHIS system. The PDM Adapter can further be used to transfer data to the opposite direction. To transmit data from MEMPHIS to the PDM system, the MEMPHIS Connector that provides the common interface in form of a neutral file can also be utilized while the PDM Connector is performing the data parsing and data transmission to the PDM server via the PDM API.

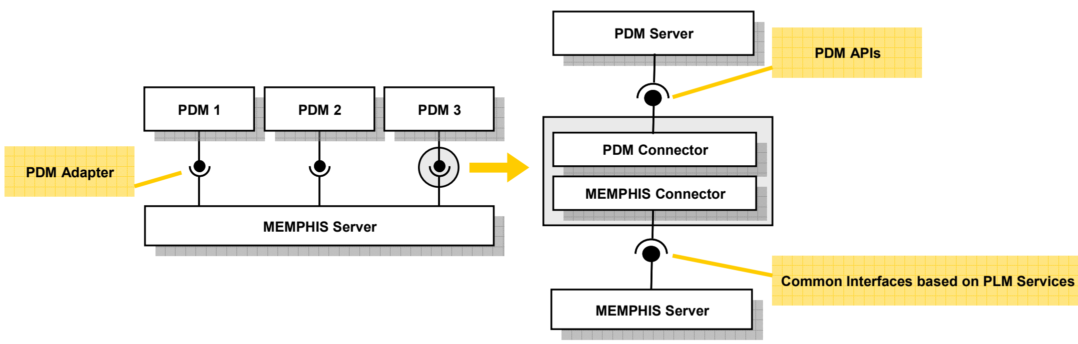


Fig. 8. PDM Adapter in the MEMPHIS system.

3.1.5 CAD2VR Converter

For the conversion of CAD formats to VR compatible files, the CAD2VR Component is integrated. On request by the MEMPHIS Server the CAD2VR Converter processes the CAD data conversion and performs automated modifications of geometry and topology. These modifications are necessary in order to get visually correct results. For geometry or topology changes, the current version of the component includes tessellation/triangulation functions for the CAD data, LOD generation and the calculation of vertex normals.
For the generation of VR data from CAD data, we used the InterOp of the ACIS CAD kernel from the Spatial Corporation [1]. The ACIS CAD kernel is an object-oriented three-dimensional geometric modeling engine and it provides an open architecture framework for wireframe-, surface- and solid-modeling using a common, unified data structure.
The InterOp provides several kinds of commercial CAD file formats for reading or writing geometry or topology data. As examples for possible CAD file formats, CATIA V4/V5, I-DEAS, Unigraphics, ProEngineer, SolidWorks and Inventor are supported. As possible neutral file formats, STEP (AP203, AP214) and IGES(Initial Graphics Exchange Specification) are supported as well. That allows a broad base for the generation of VR files (VRML, X3D) from

tessellated CAD geometry models. Yet, the generation of VR files (VRML, X3D) required the development of components for computing the tessellated boundary representations and for surface optimizations in order to solve several visualization problems connected with CAD models. For instance, the mesh quality and accuracy of the approximated surfaces must be adapted to the requirements of the target VR platform. Thus using the CAD2VR Converter allows the setting of parameters for quality or size to influence the generated result.

3.1.6 Security Component

The Security Component is applied to guarantee the safety of the managed data in the MEMPHIS system. The managed data, physical file data and plaintext stored in the database are covered. To raise the level of security, two possible methods are used. The first one is the encryption method by hash code generation which cannot be easily decrypted. The second one is the encryption and decryption method by a symmetric key algorithm (Triple DES). As the development concentrates on the middleware the cipher algorithms were not implemented but introduced by public cipher libraries.

3.2 Work Flow

Recently, most of the manufacturing companies are using PDM systems to shorten the lead-time for developing products and for the effective management of product data. PDM systems can manage CAD data of various types as mentioned in chapter 1. At that point, the user connects with a PDM system via MEMPHIS to obtain the CAD data instead of establishing a direct connection to the PDM system. Here we will give a description of work steps and the data flow in the MEMPHIS system. In Fig. 4 the location for each step is visualized. For the detailed functionalities of each component during the process please refer to 3.1.

① *CAD data acquisition from PDM system*: the physical CAD file and CAD-specific meta-data are acquired from the PDM system via the PDM Adapter utilizing the PDM Connector that is specialized to the specific PDM system.

② *Data storing with security policy*: the acquired data is stored to the File Server's DB (which keeps the CAD files) and the Meta-Data Server's DB (which keeps the semantic product structure) and integrates the meta-data (e.g. the BOM information) into this structure. To each stored data the MEMPHIS system's security policy provided by the Security Component is used to secure the data from unauthorized access.

③ *Data conversion from CAD to VR*: the stored physical CAD file can be converted to a physical VR file by the CAD2VR Converter. Optionally LOD settings can be made. The converted VR file is also stored in the File Server's DB. Recently, CATIA V4/V5, I-DEAS, Unigraphics, ProEngineer, SolidWorks, Inventor, STEP and IGES files are supported as physical CAD files. As VR output formats VRML and X3D can be acquired. If not handling single files but the product structure, each item of the model is converted and provided separately, which is necessary for the processing in step 3.

④ *Communication between the MEMPHIS Server and the MEMPHIS Client*: stored data on the server side which was acquired through the above steps (steps 1-3), can be manipulated (data search, addition, elimination and renewal) by end-user's requirement on the client side. To do this manipulation, the MEMPHIS Server (the File Server and the Meta-Data Server) and the MEMPHIS Client communicate in a distributed computing environment via SOAP messages.

⑤ *Data manipulation in VR systems*: in the MEMPHIS architecture, the VR system is introduced to create enriched VR data and to visualize this data. With the VR Adapter, the MEMPHIS Client has a connection with the system to get the VR file that was converted from CAD data in step 3. Further the related meta-data, which includes LOD information and BOM information for geometry model data can be loaded. Changes made on the client side can include attachments of new or changed properties to the product structure. By updating these changes to the server and into the meta-data DB, they are integrated into the product's structure. Later requests for this data will also allow the inclusion of this new information.

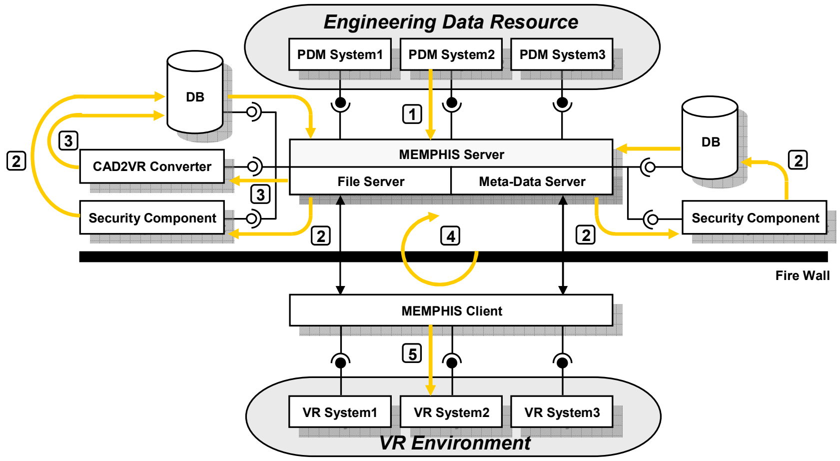


Fig. 9. Work flow in the MEMPHIS system.

4. SAMPLE APPLICATION

This chapter shows the practical use of MEMPHIS as an integrated component of the design process. As a sample for a network using the MEMPHIS system, we used Team Center, the PDM solution of UGS. Further an in-house VR authoring tool (TiciPrep) is integrated on the VR side of the system. TiciPrep concentrates on applying and managing realistic shaders to virtual models adding features like reflection, refraction and shadowed lighting to the scenes. As a VR viewer with a high quality rendering engine TiciView is used, which is also an in-house solution.

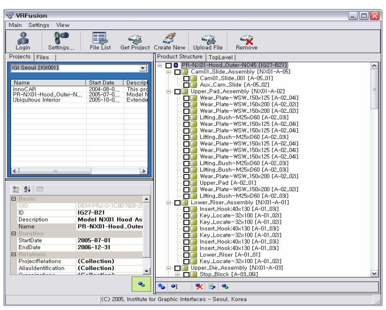


Fig. 10. Graphical user interface of the MEMPHIS client.


Fig. 11. Manipulation of VR data, provided from the MEMPHIS system, in a VR system (TiciView).

As shown in Fig. 12 the CAD information can be processed for VR presentation in a sequential manner. Commonly, after the conversion of CAD files to VR files, additional settings would have to be added by a programmer. Depending on the VR format, material settings are included to the geometry nodes. Scripts for behaviors are added as well as animation information etc. If changes have been made to the original CAD models and this data is again converted to the VR format, all inclusions have to be made again. Even if the blocks of code are stored separately to be reused, the correct integration costs high efforts repeatedly.

Using the middleware MEMPHIS, these efforts are not required anymore, as shown in the following example. The data in TeamCenter or on the MEMPHIS Server can be requested by the user of TiciPrep (either internally with the VR Adapter or with the stand-alone client [Fig. 10]). Through the MEMPHIS Client implementation, the required project and sub-items of interest can be selected. This includes already integrated VR properties. Settings for the conversion can be made. The request is processed by the MEMPHIS server. If the original data is in the PDM system and not stored in the MEMPHIS database, it will be accessed through the PDM Adapter. The specified conversions are performed by the CAD2VR component and the selected properties from the data base are added to the resulting VR file. This file is then sent back to TiciPrep.

(a) (b) (c)

Fig. 12. Data conversion process in the MEMPHIS system: (a) CAD data stored in a PDM system (TeamCenter), (b) Converted data (CAD to VRML) in the MEMPHIS system and (c) Enriched data in a VR system (TiciPrep).

In TiciPrep, VR specific properties such as high quality shaders can be added or changed. At the end of the process these changes can be written back to the MEMPHIS server for future access for e.g. TiciView (Fig. 11), other programs or other users from different locations by there client instances.

5. CONCLUSION & FUTURE WORKS

In this paper, we proposed the MEMPHIS system as a data exchange middleware between three-dimensional CAD systems and virtual reality applications via PDM systems. The system proves to be the missing “VR/CAD data hub” for the complete design phase including the conceptual design using VR and the physical design with CAD. The features for linking VR meta-data to information from multiple sources simplify the process of creating and testing virtual products. It introduces also non-technical users to the advantages of VR applications. The removal of redundant working steps is saving time and efforts in an industry with short time-to-market periods.

Recently, the system was only implemented for one PDM (TeamCenter) and VR system (TiciView) to show the proof of concept. Later it can be extended with less effort than implementing direct system-to-system interfaces. Adapters for arbitrary commercial or noncommercial PDM and VR systems can be introduced following the schema provided by the MEMPHIS system, making it a highly flexible tool within the production process.

6. ACKNOWLEDGEMENT

This work has been performed in the framework of the IMS Project, which is funded by IITA. The authors would like to acknowledge the contributions of their colleagues and project partners: IGI, ETRI and the members of INI Graphics Net.

7. REFERENCES

[1] ACIS R14 Online Help, Spatial Corporation, 2004.

[2] Corseuil, E. T. L., Raposo, A. B., da Silva, R. J. M., Pinto, M. H. G., Wagner, G. N. and Gattas, M., ENVIRON – Visualization of CAD Models In a Virtual Reality Environment, Eurographics Symposium on Virtual Environments, 2004, pp 79-82.

[3] Feltes, M., Habel, P., Lämmer, L. Nowacki, S. and Staub, G., Product Lifecycle Management Service Revised Submission, OMG document number mantis/2004-04-01, 4 May 2004, http://www.prostep.org/file/13580.plm-serv10.

[4] Gomes de Sá, A. and Zachmann, G., Virtual reality as a tool for verification of assembly and maintenance processes, *Computers & Graphics*, Vol. 23, No. 3, 1 June 1999, pp 389-403.

[5] Lukas, U. von and Nowacki, S., High Level Integration based on the PLM Services Standard, ProSTEP iViP Science Days 2005:Cross-Domain Engineering, 2005, pp 50-61.

[6] Lukas, U. von and Vahl, M., Integration of Virtual Reality and CAD based on OMG’s CAD Services Interface, European Multidisciplinary Society for Modeling and Simulation Technology (EUROSIS):European Concurrent Engineering Conference, 2003, pp 54-61.

[7] Object Management Group, Product Lifecycle Management Services Convenience Document, dtc/05-03-08, March 2005, http://www.omg.org/cgi-bin/doc?dtc/2005-03-08.

[8] Rabätje, R., Integration of Basic CAD Functions into a VR Environment, Computer Graphics International, 22-26 June 1998, pp 238-241.

[9] Seely, S., SOAP – Cross Platform Web Service Development Using XML, Prentice Hall, 2001.

[10] Teamcenter Engineering – Online Help Collection, UGS Corporation, 2003.

[11] W3C, Web Services Architecture W3C Working Group Note, 11 February 2004, http://www.w3.org/TR/ws-arch.