

A Virtual Workspace for Distributed Design and Engineering Tasks

P. Benölken, M. Wewior and U. Lang

Abstract— The development and production of consumer goods is increasingly affected by a world-wide distribution and specialization of development and production locations. Efficient IT tools are required for synchronizing decision makers, developer teams and production planners. In this paper we present the design and evaluation of our virtual workspace for supporting the collaboration of team members in geographically distributed locations. Our contribution focuses on the support for real-time collaboration in the design and engineering field but addresses also some aspects of offline collaboration.

Index Terms—Collaborative work, distributed design, real time collaboration.

I. INTRODUCTION

During the past decades the worldwide distribution and specialization of experts, vendors and customers has become an increasing factor for the development and production in different industrial sectors. The coordination between provider, developer and supplier thereby underlies high time and cost expense. Therefore effective tools for supporting collaborative tasks in geographical distributed teams have significantly gained importance over the last decades.

Prominent examples for such geographical distributions of development and production sites are found in the aerospace and automotive industry. Further applications are found in different areas of manufacturing as well as in the construction field. Frequently scenarios include the planning and design of parts and assemblies by multiple, independent teams working in parallel at separate locations with different tools. For integrating these substructures into common product design, errors, clashes and conflicts need to be resolved. Further scenarios focus on collaborative design reviews and decision supports or product presentations to the customer in early development stages.

In today's product life cycles typical design and engineering tasks include multiple development stages with different phases of individual and team work in local and distributed environments. In the initial *conception phase* the overall design goals are discussed and agreed with all stakeholders, a time plan with milestones is defined and tasks are assigned to the different partners and expert groups. This phase is usually followed by the *development-, implementation-* and

evaluation- phase with different planned and ad-hoc meetings for synchronizing the individual and team work, resolving conflicts and for planning future steps. This requires the support of efficient IT systems for managing the collaborative access and manipulation of different types of data and documents as well as for providing the required communication channels. In the EC funded project CoSpaces[31] we developed a workspace for supporting geographically distributed teams in collaboratively exchanging, analyzing, and discussing their developments.

The structure of our contribution is organized as follows: First we will briefly review the work and available solutions related to our topic. Afterwards we describe the concept and implementation of our distributed design workspace. Furthermore we present the results we obtained from evaluating our implementation with different users. Finally we close with our conclusion drawn from our work and give an outlook on future activities.

II. RELATED WORK

Techniques and solutions for computer supported collaborative work (CSCW) has been the subject to different research and development activities in the past decades. Several approaches and solution have been published and created during this time. According to Applegate's frequently cited place-time matrix [8] CSCW is classified into the following four categories:

- 1) different time, different place
- 2) same time, different place
- 3) different time, same place
- 4) same time, same place

Our contribution will mainly focus on the different place but same time section, referred as *synchronous* or *real-time collaboration* but also addresses some aspects of *asynchronous* or *offline collaboration* (different time and place).

A. Offline collaboration

Design and engineering in (distributed) teams requires the exchange and versioning of different documents and data. Version control systems and document management systems are frequently used for proper handling file based resources such as CAD models or simulation results. Besides this, tools for supporting the offline communication between project partners such as IBM Lotus Notes [14] or Microsoft Exchange [15] are required. An extensive overview and evaluation of the

different solutions in this field is given in the study of Skopik et al. [1].

Several open source and commercial solutions are available for managing different kind of contents in groups and organizations. One of the matured and hence widely used open source document management systems is Alfresco [16]. Key features are on a-synchronous collaboration support (calendar, e-mail, task, note and document management) for (web based) enterprise project, content and knowledge management. Besides such open source systems there are several commercial solutions available.

Microsoft Sharepoint [17] combines the Microsoft Windows Sharepoint Services (WSS) and Microsoft Office Sharepoint Server (MOSS) into one product. Besides the aforementioned services for supporting a-synchronous collaboration it provides tools for synchronous discussions via chat and audio/video conferencing. Teams can create project areas for exchanging and following the lifecycle of documents.

BSCW - *Basic Support for Cooperative Work* [18] is a web-based collaborative tool, developed in Python, that allows different members of a group to cooperate asynchronously over the Internet or from an intranet. Besides several groupware services for team communication, time schedules and tasks planning it offers collaborative document management facilities including an extensible role and access right management.

B. Real-time collaboration

In addition to offline collaboration, different developments and systems for real-time collaboration have been created for coordinating distributed teams and project partners. Platforms such as GoogleDocs¹ or Microsofts Groove² provide methods for co-editing documents, spread-sheets and presentations with integrated chat facilities. However, as some of these applications are hosted on third party servers (GoogleDocs) their acceptance by commercial organizations is still restricted.

Online conferencing tools such as instant messaging and audio-video conferencing provide important communication means for distributed teams. Besides different stand alone conferencing systems like Cisco's high-end TelePresence³ series, several software based solutions for audio-video conferencing with different features and capabilities became available. Typically the quality and scalability of software systems like Google Video Chat [19], Skype [20] or WebEx [21] lag behind such professional conferencing hardware. On the other hand the software solutions offer a higher flexibility in quickly accessing and operating audio-video conferences.

AccessGrid [22] is a collection of technologies for enabling large scale audio and video based online collaboration between groups in distributed locations. It interfaces with grid middleware, visualization systems and other applications to provide a rich environment for remote lectures and distributed

meetings including multimedia facilities for large format displays and high quality video and audio channels.

Over the past years Tele- immersive and Collaborative Virtual Environments (CVE) gained significant attention in the field of computer supported collaborative work. According to the scenario of a face-to-face meeting a virtual room metaphor with virtual tables and avatars is used to simulate the presence of remote partners.

With the *Carpeno* system Regenbrecht et al. [10] combine table-top interaction techniques with a CVE for supporting intuitive co-located and remote collaboration. Their approach is based on the *carPE* system [23] which simulates a 3D conference room with facilities for interacting with different kind of media. Bourke [3] explored the potential of Second Life⁴ for the remote collaborative visualization of scientific data sets. Similar to this approach Nakasone et al. [9] present with *AstroSim* an application for collaboratively visualizing astrophysics simulations. A Mixed Reality teleconferencing application based on Second Life and OpenSim [24] was recently published by Kantonen et al. [2]. Augmented Reality techniques and video based gesture detection are used for creating a shared augmented space with remote partners.

The enhancement of presence of remote partners and the provision of more intuitive interaction and communication methods are the most significant benefits of such immersive environments. However systems as Second Life typically exhibit some limitations in handling large amounts of objects as reported in [9] or in integrating new visualization features.

The collaborative visualization in distributed environments has been the subject to several research and development activities. Overviews on this field are given by Brodlie et al. [4] as well as by Grimstead et al. [6]. Basically the approaches can be distinguished into:

- 1) *Server side rendering*: All data resides on a central server or cluster, which perform the rendering and transfer the images to the clients.
- 2) *Client side rendering*: The data is distributed amongst the client hosts and rendered locally.

Besides higher graphic capabilities *Server side rendering* usually provides a more secure data access and transfer for remote collaborations. On the other side *Client side rendering* typically require less network load since only synchronization signals are transmitted.

A client side approach for collaborative direct volume rendering in distributed environments based on the COVISE⁵ framework was published in [12]. Mishchenko et al. presented in [7] the architecture of an open source framework for distributed and collaborative visualization including direct volume rendering capabilities. In [5] Zhao et al. presented a server side approach for web-based synchronized collaborative visualization of large scale data sets in a web browser via web services. Finally Al-Saidi et al. presented in [11] a platform for

¹ <http://docs.google.com>

² <http://grv.microsoft.com>

³ <http://www.cisco.com/Telepresence>

⁴ <http://www.secondlife.com>

⁵ <http://www.visenso.de>

distributed collaborative visualization by investigating Image-Based Rendering (IBR) techniques.

C. Collaborative Design

Different commercially available products appeared on the CAD software market, which explicitly address collaborative aspects of computer aided product development. Products such as PTC/CoCreate OneSpace [25] and CollabCAD [26] concentrate on providing collaborative tools for the design phase while other tools like Oracle/Cimmetry AutoVue [27] or Enovia DMU Navigator [28] focus on enabling distributed collaboration for design reviewing tasks. Beside such integrated solutions there are also software frameworks like HOOPS [29] and Java3D [30] which started providing basic functions for distributed collaboration.

Nevertheless the transfer performance and interoperability of these systems still needs to be considered for larger models and groups. The integration between different CAD systems is one of the most prominent expense factors in industry. As major software vendors only provide support for their own data formats, users have to rely on external translators and converters. This entails additional costs and time-consuming data preprocessing steps.

Another issue of many collaborative CAD tools available today is the visualization performance during distributed collaborative sessions. Most distributed visualization approaches are based on techniques which require high network bandwidths and thus are limited by the performance of the underlying communication medium. This bottleneck can be avoided by minimizing the amount of data to transfer over the network. Here, efficient geometry compression algorithms as presented in [13] and intelligent methods for identifying redundant data which can be omitted for distributed synchronization will have a significant impact on the overall performance.

III. DISTRIBUTED DESIGN WORKSPACE

The design of our distributed workspace is based on the elicited requirements and scenarios developed in cooperation with industrial partners in the EC funded project CoSpaces[31]. In general the requirements for a distributed collaborative working environment can be summarized as follows:

- Organization of meetings:
 - Selection of partners, data, applications and documents.
 - Plan and schedule meetings.
- Shared visualization:
 - Browse 3D assembly and part structures.
 - Navigate through 3D geometries.
 - Point to and annotate parts and assemblies.
- Online conferencing via chat and audio/video conferencing.

Our design is based on the assumption that different solutions are required in specific collaborative environments. Furthermore the creation of the collaborative working environment is not built from scratch - existing tools need to

be integrated, legacy applications have to be considered.

A. Collaboration Lifecycle

For the general concept we assumed that collaboration is undergoing a lifecycle, during which collaborative sessions are prepared, executed and finalized by a predefined group of partners. These sessions of active collaboration usually follow a well defined workflow as shown in Fig. 1.

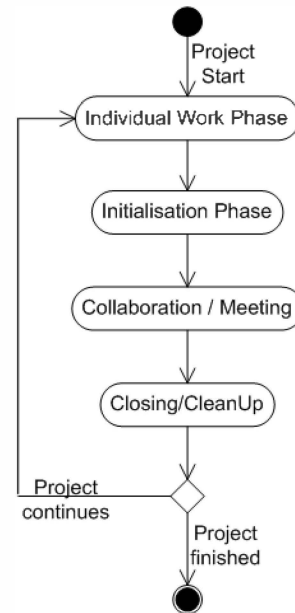


Fig. 1, The collaboration lifecycle

The collaborative tasks are preceded by the individual work carried out by the participants of the project. Problems may be encountered which need to be solved collaboratively. Therefore documents describing the problem have to be compiled to be used during the collaborative session or to be distributed beforehand.

During the initialization phase, one participant starts the scheduling process. This determines the date and participants and type of the collaborative session. Afterwards all participants need to agree on the settings. The session may be reconfigured until a common consent has been reached. Configuration includes identifying the availability of the required resources such as applications, hardware resources, rooms, experts, participants, documents like agenda, minutes of previous meetings, and decision documentation. Large data, which cannot be accessed on demand over networks, is distributed to the local hosts. The applications, which now were started on the local hosts for collaboration, can access the data without large delays.

After the applications are available, the session is ready and open for the users. Subsequently collaboration between the users takes place, during which data is produced and modified. At the end of the meeting these data and a collaborative written summary can be stored. This includes minutes of the meetings, enhanced by recordings, plus documentation of decisions. Finally these results will be sent to all authorized participants and the session will be closed, including all connections, applications, and processes.

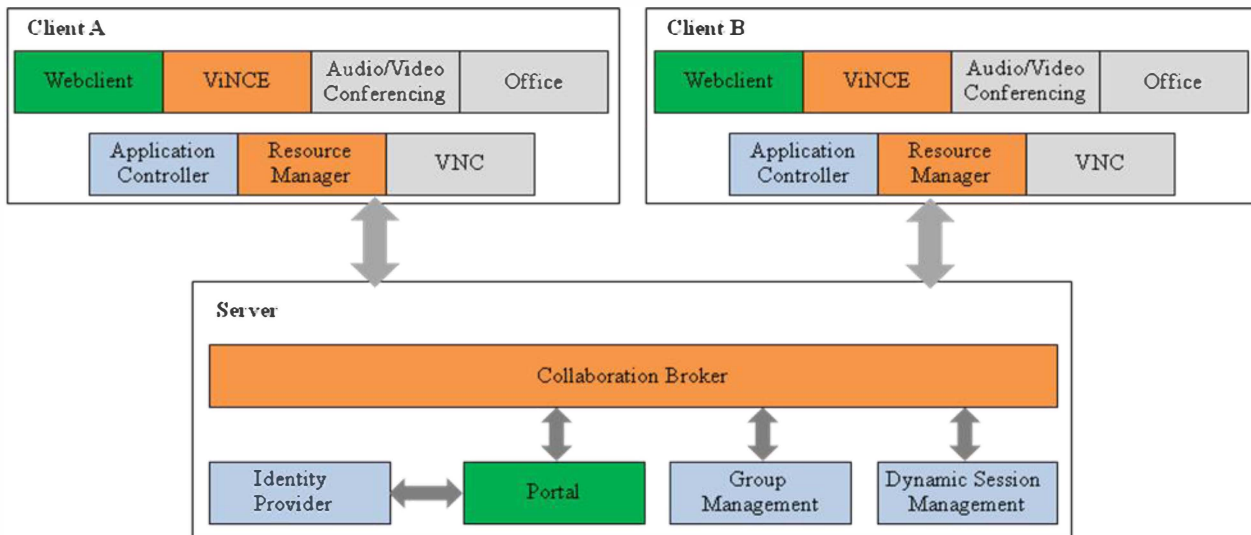


Fig. 2, Distributed workspace design.

B. Overall Concept

According to the previously described considerations we distilled a set of components encapsulating the required functionalities. A modular approach dividing the functionality into separate components has been chosen in order to create an adaptable software framework which can be easily adjusted to the specific requirements. The components are to be designed to collaborate on basis of well-established, open interfaces as largely as possible - a service oriented approach is intended. The interfaces should allow for platform independent communication methods, regardless of the programming languages used for the implementation of the individual components. For these reasons Web Services based on the SOAP protocol have been chosen as the preferred communication medium between components. Fig. 2 illustrates how this conceptual design has been applied in defining the collaboration services specific to the distributed workspace.

C. Workspace service components

As shown in Fig. 2 the workspace design is based on a client-server-architecture. Central supporting components are installed on a server and provide services for the client service components, which are installed on each client's host computer. These client services support the applications used in each workspace. A VPN tunnel is deployed for bundling the different in- and outgoing connections between client and server. This enables the user to easily open a secured, private, encrypted connection with all other participants of a session.

On the server side the following components are provided:

Portal

The general entry point into the software framework is the Portal. With its graphical user interface it provides the front end to the functionalities implemented by the collaboration broker. The portal is used during the entire lifetime of the collaboration lifecycle. It supports asynchronous tasks carried out during the individual work phase, as well as on the fly reconfiguration of ongoing sessions during the meeting phase.

Identity Provider

This component is used by the Portal for requesting services for managing users and authenticating users at the different components.

Group Management

The Group Management is based on the BSCW system. It provides data on available projects, workspaces, their users and the applications registered for the users.

Dynamic Session Management

The Dynamic Session Management (DSM) controls the execution of applications as well as the communication between the involved applications and users during synchronous collaborative sessions.

Collaboration Broker

The Collaboration Broker (CB) provides backend functionalities of the Portal. It provides means for orchestrating the different tasks as well as the inter-component communication between all other components to retrieve information, process data and to perform changes. It is a central service for preparing and setting up new collaborative sessions. In this context the Collaboration Broker provides the functionalities for administration and maintenance of collaborative workspaces. It supports asynchronous tasks carried out during asynchronous collaboration or the individual work phases. The Collaboration Broker is also active during the initialization phase, preparing the collaborative session and ending the session.

The design of the CB follows a modular approach as shown in Fig. 1. The different modules can be subdivided into specific services modules such as the *Portal connection module* and support modules such as the *I/O-module* which provide general services to access the PostgreSQL-database management system. Each service module is utilising a Webservice module for processing the SOAP-messages received and building the correct SOAP-responses. The tools

module contains a XML-parser. The main module finally initialises starts and connects the different modules.

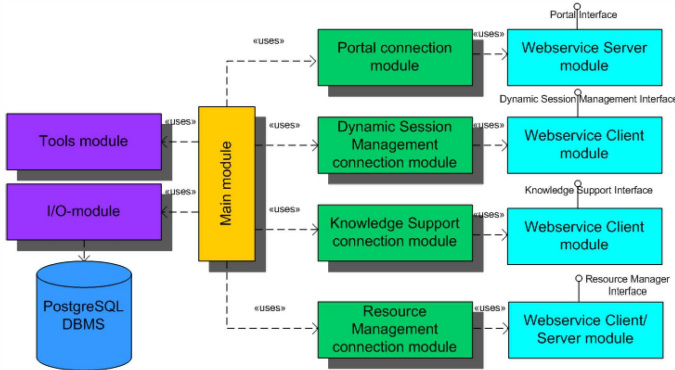


Fig. 3 Design of the Collaboration Broker

Requests from the Portal are received by a gSOAP-based Webservice Server module. The response is generated inside the Portal connection module. The actions range from using the I/O-module to retrieve locale data to using the Knowledge Support Connection module, or the Resource Management connection module to retrieve data from these sources. The Dynamic Session Management connect module provides information on sessions and reacts to requests like session configuration.

The Knowledge Support connection module provides access to the Group Management. Other modules can use the functions provided to manipulate the workspaces (user and group management, data and file management). Additionally information for synchronous collaborations can be stored and retrieved. Amongst them are applications and application controllers configured for users.

XML-configurations for synchronous collaborations are sent to the DSM by using the access functions provided by the connection module. This module provides information on configured sessions and the ability to sent commands to these sessions as well.

The I/O-module provides access to local data. On the one hand it reads configuration files and writes log files on the other hand it provides access to a PostgreSQL database.

The Tools module provides access to information inside configuration files. Logging information is processed by e.g. pre-pending a date before it is written by the I/O-module. The common functions for initialising endpoints and encryption are placed here. Methods for initialising and handling TLS-related function are included. These include requirements by OpenSSL and gSOAP to support TLS-secured connections. Furthermore support functions are included, e.g. to gather the usernames from the certificates to map users to Resource Managers. More functions will be included to support automated generation of certificates which are used by Resource Management and Application Controller for authentication.

On the client side the following services are executed on each local computer participating in a collaborative session:

Application Controller

The Application Controller is responsible for starting and steering the applications necessary for a collaborative session by accessing a standardised interface for the tools.

VNC

An installation of *TurboVNC* is used to provide easy ad-hoc screen sharing facilities. This enables users to share applications which are not collaborative or not integrated with the distributed workspace software. The steering of this tool is implemented in a Java-Applet inside the Portal.

Resource Manager

The Resource Manager (RM) provides means for sharing very large data like models from Computer Aided Design (CAD) applications, videos or design drafts. As uploading such large bulk data to a central storage would not be efficient, the RM was designed as a decentralised service located at each end user's machine which allows selecting local files for a collaborative session and distributing them directly to the participants as required. Each client who needs to share bulk data with other clients requires the RM service installed on his local machine.

The design of the Resource Manager is shown in Fig. 4. All RM instances are coordinated at the RM Directory which is a subcomponent of the CB. The CB displays all information about connected RM instances to the user via the Portal. The Portal is the central place for the user where he can interact with all RMs owned by him.

The communication and steering between the RM instances are done via Web Services function calls at the CB. This implies that RMs never communicate directly with each other but always use the CB as mediator. The only exception is the data transfer which is executed directly between the sender and receiver RMs to reduce delays and increase transfer speed.

The Data Layer of the RM is the place where all user data and metadata is stored. It is composed of two storage facilities. The local file system is used to store local user data as well as replicas of remote files shared by other clients. A XML database stores metadata and management information like users, groups, data access rights, etc.

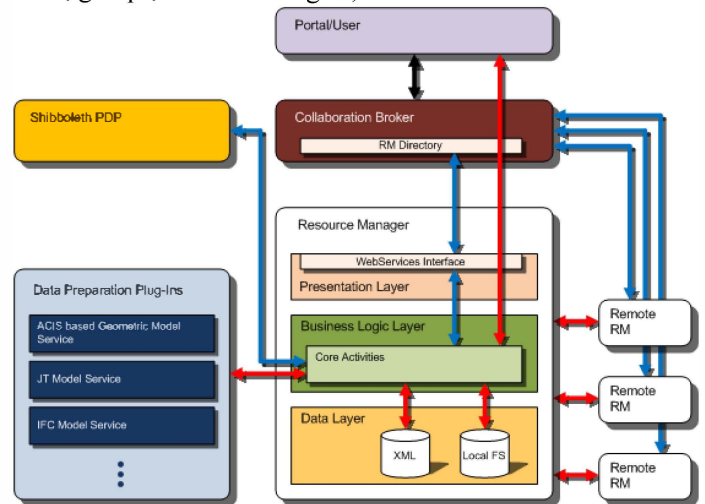


Fig. 4 Design of the Resource Manager

The Business Logic Layer is responsible for checking user access rights and for preparing user data for transmission. Two ways to manage access rights are foreseen in the RM architecture. Firstly, strict data-item level access rights checking, similar to common file system access rights management, which is performed locally for each data item. The second method is a rule-based and more general access rights checking mechanism which is performed remotely at the Shibboleth PDP (Policy Decision Point) via Web Services. Both are optional and can be disabled.

The Presentation layer provides a Web Services Interface to the CB where all management and usage functions are exposed. These functions are presented to the user in the Portal in a user-friendly and intuitive way. Moreover, the user logged in at the Portal can access any shared files directly on his local file system, provided that he has a RM instance running on his local machine.

Supported by these services components different applications are provided for each collaboration partner in the distributed workspace:

Audio/Video conferencing

An audio/video conferencing application is provided by a flash program, executed directly from the portal.

Office

Office applications are supported by specific plugins for accessing and collaboratively exchanging documents, spreadsheets and presentations.

ViNCE

ViNCE – *Visualization in Networked Collaborative Environments* is a collaborative DMU-Review tool which enables the user to upload and visualize native 3D-CAD-models. A screenshot from the user interface of this application is shown in Fig. 5. Beside the synchronous exploration of 3D models in distributed environments, this viewer provides different functionalities for collaborative design reviews, which include:

- A structure browser for displaying and navigating in the product structure.
- Facilities for highlighting, selecting and hiding parts of the 3D model.
- Clip-planes for inspecting internal structures.
- Tools for adding, saving and loading annotations to the 3D model.
- Methods for recording, saving and loading camera positions.
- A telepointer for pointing to specific parts in the 3D scene.
- A chat window for text-based communication with remote partners.
- A reference plane with soft shadows for supporting the 3D perception of 3D models.

In general this application follows a *client side rendering* approach. First, the 3D CAD models are uploaded in their native format and converted into polygon models. Conversion parameters and hence the quality and size of the polygon

model can be adjusted by the user. These polygon models are distributed amongst all partners and displayed on the local screens by using the graphics hardware at the client side. Afterwards only single commands for synchronizing the different views such as the position of the telepointer or the transformation matrices need to be transferred. A master-slave mode is provided for choosing between tight and loose synchronization of the partner views. The master role can be exchanged on request amongst the participants of a collaborative session.

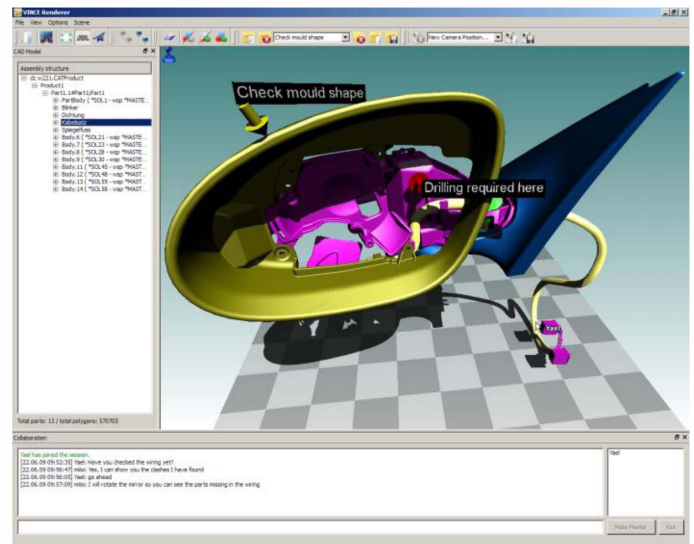


Fig. 5, a screenshot of the ViNCE DMU-Review tool.

IV. EVALUATION

We evaluated our distributed collaboration workspace by conducting a series of collaborative sessions with different users in distributed locations. The aim of this evaluation was to verify the conception, to gain a feedback on the usability of our implementation and to identify shortcomings as well as potentials for further improvements. The evaluation was organised to involve experts from the industrial partners as well as internal users to provide their opinion on the general usability of the software. The users from the industrial partner group contributed with their expertise in collaborative engineering whereas the internal users provided their IT experience. During the multiple collaborative sessions the partners participated from different sites located in France, Germany and Hungary.

A. Scenario and Method

Based on the partner requirements we developed a scenario which resembles all parts of the collaboration process in engineering but is focused on evaluating the properties of the software. This way we used a comparable setting for all evaluations, which is close to the industrial scenarios, but still easily understandable without expert knowledge. Further details of the more specific industrial scenarios and the context of certain actions were provided to the users on request.

As a reference data set for the different online collaboration sessions we used a CAD model of a car mirror composed of 13 parts and 346094 polygons. In a pre-processing step the model was transferred to the partner sites and uploaded into the ViNCE tools executed at the different local hosts. Synchronisation signals such as transformation matrices are distributed via the network to all partner hosts participating in the active collaborative session. Therefore the time for image rendering is solely determined by the graphic capabilities of the local host computer. For testing the rendering performance of the ViNCE tool we used a CAD model of an airplane wing and carrier with 897 parts. The conversion from the native CAD file format resulted into a surface geometry with 1.847.831 polygons which were rendered with real time shadows and high quality shading at interactive frame rates (> 25 Fps) on a 2.2 GHz Intel Core 2 Duo laptop PC equipped with 2GB RAM and a NVidia GeForce 8600M GS graphics board.

For a realistic rating of the software during the evaluation the users were located either resembling “real world” conditions or utilizing network emulators. In the first case the evaluation test were conducted between locations in Germany, Southern France, and Hungary. The emulation was accomplished by WANem [32] resembling low broadband connections (2Mbit/s downstream, 1Mbit/s upstream, 50ms average latency, 50ms average jitter, 1% average package loss).

During the evaluation the user should take on creating and executing a meeting with another user. After selecting the participating users and configuring the applications and data, the meeting is started.

The videoconferencing facilities are used for an introduction and conversation throughout the meeting. The application sharing functionality can be used to ad-hoc-share application, like text documents or presentations.

The ViNCE tool is used during this synchronous collaborative session for interactively exploring 3D CAD models. The task was to present the opened CAD models to the remote partners by interactively moving the camera position and selecting pre-defined positions. This resembles the presentation of progress to a customer or the presentation of the parts of a model where further actions need to be taken. Next annotations need to be placed on the model. The annotations are stored in an XML file and a screenshot is saved which shows the problem(s). Usually this function is used to mark identified problems or request changes to certain parts.

For DMU-Reviews it is essential to explore the internal structure of a model. ViNCE supports selecting parts of the model and hiding them as well as focussing on a single part, by setting it to be the only one visible.

Afterwards the master control is switched over. During the evaluation this time is used to present the concept of the master-role and to present the clipping plane. Finally the meeting is closed by logging out of the portal.

B. Results

Users were provided with a questionnaire including statements on the framework and its components. They were asked how much they agree with the statements. Table 1 shows the mapping of the answers to numbers. For achieving preferably meaningful results, the questionnaire included positive as well as negative statements. Thereby a (strong) agreement on a positive statement (e.g. q1) denotes a positive rating and a negative rating for negative statements (e.g. q6). Vice versa a (strong) disagreement on a negative statement (e.g. q7) denotes a positive rating and a negative rating for positive questions (e.g. 3).

A total of 17 users answered the questionnaires. Since some components provide background services our evaluation focused on the visible applications and tools such as the Portal and ViNCE.

Table 1: Mapping on ratings to values

Strongly disagree	1
Disagree	2
Neutral	3
Agree	4
Strongly agree	5

The first part of the questionnaire addressed the overall impression of the users on the distributed collaboration workspace. The users agreed that the system is easy to use (question 3(q3), see 2.3) and they do neither need technical support to use it (q4) nor a lot to learn (q7). The system was agreed to be quite comfortable to use (q6), not unnecessarily complex (q2) or inconsistent (q5). From their first impression, all users stated that they would frequently use the system (q1). The results are summarized in Table 2.

The second part of our questionnaire addressed the Portal since this component is the central entry point into the distributed workspace. Table 3 provides an overview of the users’ answers.

Table 2: Answers on questions on the overall system

No.	Question	Average
1	I would like to use this system frequently	4.3
2	I found the system is unnecessarily complex	1.7
3	The system is easy to use	4.3
4	I need technical support to use this system	2.0
5	There is too much inconsistency in this system	1.7
6	The system is very uncomfortable to use	1.7
7	I needed to learn a lot of things before I could get going with this system	1.3

Table 3: Answers to questions on the Portal

No.	Question	Average
8	Log on to the framework is simple	3,7
9	The graphical interface requires additional improvements	3,6
10	The graphical interface was easy to use	3,6
11	The content was clearly structured and could be easily understood	3,5
12	Participating into a pre-configured session was easy	3,8
13	I needed additional assistance for joining into a collaborative session	3,2

14	The process flow was easy to understand and to handle	3,5
15	I had difficulties in understanding the required steps	2,5
16	Configuration of a new session requires expert knowledge	2,3
17	Managing and configuring sessions is simple	3,7

According to the answers, the portal is well fitted for the tasks to be carried out during the evaluation: Logging in is not considered too difficult (q8), configuring a meeting does not require expert knowledge (q16) and the steps are not difficult to understand (q15). Participating in a pre configured meeting was experienced being easy as well (q12) and no assistance is required when joining a meeting (q13). Nevertheless there still is room for improvements (q9), especially on the GUI and the structure (q10, q11). Obviously there are features which exceed the requirements, but are nevertheless anticipated (q9).

As a central component for synchronous collaboration tasks in engineering field we finally evaluated the ViNCE DMU-Review tool. The results are shown in Table 1.

Regarding the overall impression (question 22(q22) and q23) answers were different, yet show a positive tendency. We asked whether the layout and design of the user interface were appealing (q22). The answers ranged from neutral over agreement to strong agreement. Question 23 - whether the user interface would need to be revised - was answered mostly neutral – one disagreed. The users agreed that it was easy to switch the master role (q20) and to annotate the model (q21). Navigating through the 3D-scene (q24) and selecting parts of the model (assemblies) (q25) was regarded as being not difficult as well as hiding and un-hiding parts (q30). Yet when asked whether navigating through the product’s structure (q29) was intuitive, there was agreement with a trend towards a neutral opinion. The activation and handling of the clip plane was perceived as not difficult (q31). The telepointer was commonly regarded as helpful when discussing with remote partners (q34) and most users do not think it needed to be revised (q35). The reference plane and the shadows support the perception of the 3D-model efficiently (all ‘agreed’ to q32). The question whether the shadows on the reference plane distract from the content (q33) was answered nonuniformly ranging from neutral to strong disagreement. One of the most important features of ViNCE, the CAD-conversion, was subject of questions 18,19,26, and 27. While all agreed that the process of opening and converting a CAD-file was simple (q18), the time necessary for conversion was not completely satisfying (q19). Besides that, the rendering quality (q26) was recognised as sufficiently good. The answers for q27 support this, though with a trend towards neutrality. The response time for user interaction (q28) was assessed very differently. While one user strongly disagreed, that the response times were short, the others agreed or took a neutral stance.

Table 4: Answers on questions on ViNCE

No.	Question	Average
18	Importing / converting of native CAD files is simple	4.3
19	The conversion time was acceptable	3.7
20	It was easy to access / change the master role	4.0
21	It was difficult to annotate the model	1.7

22	The layout and design of the GUI is appealing	4.0
23	The GUI needs to be revised further	2.7
24	It was difficult to navigate in the 3D scene	1.7
25	It was easy to highlight parts/ assemblies	4.0
26	The quality of the model display was sufficient	4.3
27	The rendering quality needs to be improved	2.3
28	The response time on user interaction was short	2.7
29	The system supported an intuitive navigating in the product structure	3.7
30	It is easy to hide/unhide parts of the 3D model	4.0
31	I had difficulties in activating and handling the clipping plane	2.3
32	The perception of the 3D model is efficiently supported by the shadows and reference plane	4.0
33	The display of shadows distracts from the 3D content	2.0
34	The telepointer is helpful for discussing with remote partners	4.0
35	The telepointer needs to be revised for a more efficient perception and usage	2.3

V. CONCLUSION AND OUTLOOK

We presented the design and evaluation of our distributed workspace for collaborative design and engineering tasks. Analysing the results of the evaluation yields a positive impression. It was stated that the overall system was comfortable to use without the need for much additional support or training.

Users with professional background in the area of CAD-design provided positive feedback on the functionality integrated for the design review in the ViNCE tool. This shows that the development successfully fulfils the requirements. Elements such as the structure browser, the telepointer and the clip plane were evaluated from rather positive to neutral. Following our observation above, we can deduce that these components could be further refined.

The CAD-conversion-process was positively judged. Only the speed and quality of conversion were not described as fully satisfying. A possible reason for this cautious rating is that we used a CAD-model which was not known to the evaluators. It would be desirable to make further tests with models provided, and well known, by the CAD-designers such that they have a ‘feeling’ for speed and quality of the model. As a result, possible improvements like conversion optimisations or compression algorithms will have to be elaborated.

Summarizing, this evaluation has shown that our distributed workspace is useful for the end-users. We received a positive feedback on the evaluated components – in particular of the DMU-review tool. The reviewing tools and the very short response times during collaboration were important factors which were considered to have the potential for leveraging the user experience and productivity for different industrial sectors such as the aerospace and automotive industry. Still most important is the CAD support for opening and collaboratively exploring CAD models in their native format, combined with the possibility to easily and quickly cooperate over standard networks.

Besides improvements according to the user requirements for additional functionalities such as the inclusion of a measurement tool into ViNCE, future activities will address the CAD integration. For collaboratively handling large

models in a distributed environment the conversion and transfer times still require some optimization. Techniques for distributing the tessellation process amongst different nodes in a local network will significantly speed up the overall workflow and hence reduce the time required for preparing a collaborative session. Methods for compressing polygonal meshes will additionally improve the transfer times to the remote partners.

Further developments will focus on advanced user interfaces. Products and applications such as Apple's iPad or Microsoft Surface multi-touch table show the potential of intuitive interaction facilities. Techniques for camera based gesture recognition which gained increasing attention over the past years will provide a basis for a more natural and controller free 3D interaction in collaborative working environments.

For achieving optimal response times during real-time collaborations we selected a *Client side rendering* approach. Data conversions and transfers are executed during the offline (asynchronous) phases of collaboration. This approach surely minimizes the influence of network loads and latency times in distributed online sessions with partners in different countries.

On the one hand this concept raises additional security and confidentiality issues since the complete polygon models are distributed over the partner network and hence leave the owners control as mentioned in chapter II. On the other hand *Server side rendering* includes higher demands on reliable network bandwidth and latency times especially for synchronous collaborations across multiple countries. The real-time coding and streaming of polygons, similar to online video streams, might be a feasible compromise for getting out of the dilemma in either losing control or performance in synchronous collaborative sessions. Combined with temporal keys, solely verified partners will be able to decode and hence render the polygon data.

ACKNOWLEDGMENT

We would like to acknowledge the support of the CoSpaces Integrated Project (FP6-IST-5-034245) and the European Commission in funding this work, and CoSpaces user partners in contributing to the research.

REFERENCES

- [1] Skopik F., Truong H.-L., Dustdar S. (2008). Current and Future Technologies for Collaborative Working Environments. *Study*, ESA ITT Number AO/3-12280/07/NL/CB, May 2008.
- [2] Kantonen T., Woodward C., Katz N., "Mixed reality in virtual world teleconferencing", Proc. IEEE Virtual Reality 2010 Waltham, Massachusetts, USA, March 20 - 24, pp. 179-182.
- [3] Bourke P.D. "Evaluating Second Life as a tool for collaborative scientific visualization". Computer Games and Allied Technology. Singapore, April 28-30, 2008.
- [4] K.W. Brodrie, D.A. Duce, J.R. Gallop, J.P.R.B. Walton and J.D. Wood, "Distributed and Collaborative Visualization", Computer Graphics Forum, Volume 23, Number 2(2004), pp 223-251.
- [5] Yongwang Zhao, Chunyang Hu, Yonggang Huang, Dianfu Ma, "Collaborative Visualization of Large Scale Datasets Using Web Services," icwi, pp.62, Second International Conference on Internet and Web Applications and Services (ICIW'07), 2007
- [6] I. J. Grimstead, D. W. Walker, and N. J. Avis, "Collaborative visualization: a review and taxonomy," in Distributed Simulation and Real-Time Applications, 2005. Proceedings. Ninth IEEE International Symposium on Distributed Simulation and Real-Time Applications (DSRT'05), 2005, pp. 61-69.
- [7] Oleg Mishchenko, Sundaresan Raman, and Roger Crawfis: "Distributed visualization framework architecture", Proc. SPIE 7530, 75300K (2010), DOI:10.1117/12.838702
- [8] Applegate L. M., "Technology Support for Cooperative Work: A Framework for Studying Introduction and Assimilation in Organizations", Journal Organizational Computing 1 (1991), 11-39. 3.
- [9] Arturo Nakasone, Helmut Prendinger, Simon Holland, Piet Hut, Jun Makino, Ken Miura, "AstroSim: Collaborative Visualization of an Astrophysics Simulation in Second Life," IEEE Computer Graphics and Applications, pp. 69-81, September/October, 2009
- [10] Regenbrecht, H., Haller, M., Hauber, J. and Billinghurst, M. Carpeno: Interfacing Remote Collaborative Virtual Environments with Table-Top Interaction., *Virtual Reality- Systems Development and Applications* (2006), ISSN 1434-9957
- [11] A. Al-Saidi, N.J. Avis, I.J. Grimstead, O.F. Rana, "Distributed Collaborative Visualization Using Light Field Rendering," Cluster Computing and the Grid, IEEE International Symposium on, pp. 609-614, 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009
- [12] P. Benölken, R. Niemeier, U. Lang., Collaborative Volume Rendering in a Distributed Virtual Reality Environment", Eurographics Workshop Proceeding Series, pp. 47-1 - 47-10, Stuttgart 1998.
- [13] P. Benölken; A. Stork: "Geometry Compression for Collaborative CAD Applications". In: Stelzer, Ralph H. (Ed.) u.a.; Gesellschaft für Informatik (GI), Group 4.2.1 Computer Aided Design (CAD) u.a.: CAD 2002. Proceedings : Corporate Engineering Research. Bonn : GI, 2002, pp. 121-129.
- [14] IBM Lotus Notes: <http://www.ibm.com/developerworks/lotus>
- [15] Microsoft Exchange: <http://www.microsoft.com/exchange/2010/en/us/default.aspx>
- [16] Alfresco: <http://www.alfresco.com>
- [17] Microsoft Sharepoint: <http://www.microsoft.com/sharepoint>
- [18] BSCW: <http://www.bscw.de>
- [19] <http://www.google.com/chat/video>
- [20] <http://www.skype.com>
- [21] <http://www.webex.com>
- [22] <http://www.accessgrid.org>
- [23] Regenbrecht, H., Lum, T., Kohler, P., Ott, C., Wagner, M., Wilke, W., Mueller, E. (2004). Using Augmented Virtuality for Remote Collaboration. Presence: Teleoperators and virtual environments, 13(3), pp. 338-354.
- [24] <http://opensimulator.org>
- [25] <http://www.ptc.com/products/cocreate>
- [26] <http://www.collabcad.com>
- [27] <http://www.oracle.com/us/products/applications/autoVue/index.html>
- [28] <http://www.3ds.com/products/enovia>
- [29] <http://hoops3d.com>
- [30] <https://java3d.dev.java.net/>
- [31] <http://www.cospaces.org/>
- [32] WANem: The Wide Area Network Emulator, <http://wanem.sourceforge.net/>