



Distributed Architecture for Remote Collaborative Modification of Parametric CAD Data

Yujiro Okuya, Nicolas Ladeveze, Olivier Gladin, Cédric Fleury, Patrick Bourdot

► To cite this version:

Yujiro Okuya, Nicolas Ladeveze, Olivier Gladin, Cédric Fleury, Patrick Bourdot. Distributed Architecture for Remote Collaborative Modification of Parametric CAD Data. IEEE VR International Workshop on 3D Collaborative Virtual Environments (3DCVE 2018), Mar 2018, Reutlingen, Germany. hal-01849289

HAL Id: hal-01849289

<https://inria.hal.science/hal-01849289>

Submitted on 25 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed Architecture for Remote Collaborative Modification of Parametric CAD Data

Yujiro Okuya¹

Nicolas Ladeveze¹

Olivier Gladin²

Cédric Fleury²

Patrick Bourdot^{1 *}

¹ VENISE group, LIMSI/CNRS, Université Paris-Saclay F-91400 Orsay, France

² LRI, Univ. Paris-Sud, CNRS, Inria, Université Paris-Saclay F-91400 Orsay, France

ABSTRACT

Companies are now using Virtual Reality (VR) for collaborative design reviews on digital mock-ups. These meetings often involve remote collaborators due to current trends towards decentralization of work organization. While lots of previous works proposed distributed architectures for implementing Collaborative Virtual Environments (CVE), modifying native CAD parts in such environments is challenging. There are two main difficulties: (i) how to directly modify native CAD data (i.e. data used internally in CAD software) from the virtual environment, and (ii) how to manage collaborative modifications of such data by remote users. Most common VR-CAD applications require data conversions before the VR session and post-modifications of original CAD data afterwards. Only a few VR applications allow direct modifications of native CAD data, but they do not support remote collaboration. In this paper, we propose a distributed architecture allowing collaborative modifications of native CAD data from remote and heterogeneous platforms. Technically, a VR-CAD server embedding a CAD engine is included in our architecture to load and modify native CAD data according to remote requests. A proof of concept uses our architecture to connect a wall-sized display and a CAVE-like system.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality; Human-centered computing—Collaborative and social computing—Computer supported cooperative work

1 INTRODUCTION

Collaborative design reviews using Virtual Reality (VR) platforms are becoming common in industrial companies [4]. Such meetings involve various specialists (e.g. stylists, engineers, ergonomists) who aim to examine different aspects of a product, such as aesthetic, functionality and usability. In today's globalized world, these experts can be geographically spread and these meetings often require collaboration between remote locations.

In the context of industrial product reviews, products are modeled by parametric CAD software to support the whole manufacturing processes. These CAD data are difficult to load and modify in an immersive virtual environment because of their complexities. Consequently, collaborative meetings with remote participants do not allow live modifications of such data, and all modifications must be applied after the meeting by a CAD engineer on a workstation.

Although, industrial CAD systems (e.g. 3D CAD® from SolidWorks or CATIA V6® from Dassault Systèmes) support collaborative product reviews between remote users, they do not support such collaborative meetings in VR. Previous work [8] proposed a solution to import and modify native CAD data in an immersive environment, however, without support for remote collaboration.

In this paper, we describe a distributed architecture allowing collaborative modifications of native CAD data from remote platforms. This architecture deals with heterogeneous platforms which suit

different users' needs well: stylists might prefer a high resolution visualization, while ergonomists might prefer an immersive rendering. To support modifications of native CAD data in a Collaborative Virtual Environment (CVE), we propose a distributed architecture including a VR-CAD server which embeds a commercial CAD engine. This server is an extension of the work by Martin et al. [8].

Section 2 reviews previous work. Section 3 details the current industrial design process and presents a use case for our system. Section 4 describes the system. Section 5 reports a proof of concept of our architecture. Section 6 concludes the paper by discussing the limitations of our approach and presenting future work.

2 RELATED WORK

While VR is currently used in industry for product reviews without any live modifications of CAD data, most of the VR-CAD applications are still research works.

2.1 Collaborative Applications for Product Reviews

Some collaborative applications focus on product reviews even if they do not include any CAD data. CALVIN [6], for example, focused on collaborative architectural design for multidisciplinary experts, such as architects and engineers. The experts could explore and manipulate objects in a shared virtual environment. They were supported with distinct viewpoints according to their expertise: an inside-out view for engineers, and an outside-in view for architects. Lehner et al. [5] developed a collaborative review system for vehicle design. This work focused on the awareness of other collaborators: the system provided video and audio streams for communication between users in the CVE to improve awareness.

2.2 Collaborative VR-CAD Applications

A few previous works allowed multiple users to interact with CAD data in a CVE. MAS (Multi-Agent System) [7] addressed collaborative product design meetings across a VR platform and workstations. This system used a commercial CAD software. Engineers and ergonomists could manipulate the global position/orientation of CAD objects (sets of CAD parts) from each platform, but they could not modify the shape of these objects (i.e. parameters of the CAD parts). DVDS [2] enabled users to create a 3D model with hand gestures in a virtual environment. It used a customized CAD system. Consequently, it could not load and modify existing native CAD data designed by a commercial CAD software. A collaborative architecture was also discussed in this paper, but not implemented.

2.3 Native CAD Data Modifications in VR

cReaVR [8] proposed a solution to load native CAD data and modify it: it loaded the data by parsing whole Constructive History Graph (CHG) and Boundary Representation (B-Rep). A user could thus edit parameter values of a CAD part with a horizontal hand motion in a CAVE. The modifications were directly applied to the native CAD data by a commercial CAD engine (CATIA V5®, Dassault Systèmes). This system could overcome the issue of the VR-CAD data integration, however, it did not support collaboration.

While previous work enabled remote users to perform immersive product reviews in a CVE, collaborative modification of CAD part parameters was not supported, especially for native CAD data.

*{okuya, ladeveze, pb}@limsi.fr, olivier.gladin@inria.fr, cfleury@lri.fr

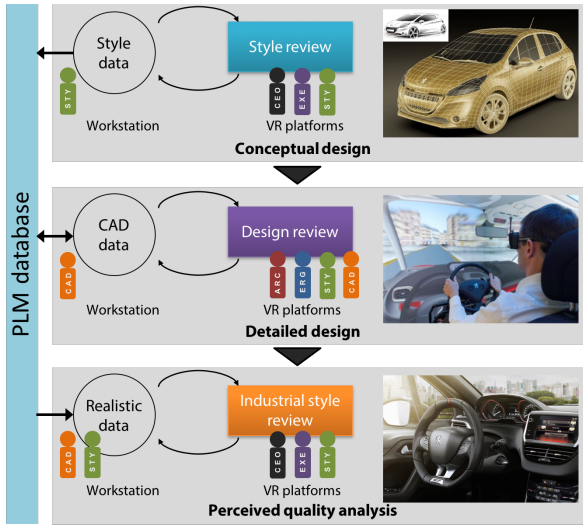


Figure 1: Collaborative product design meetings all along the industrial design process. CEO: Chief Executive Officer, EXE: Other executives, STY: Stylist, ARC: Architect, CAD: CAD engineer, ERG: Ergonomist.

3 COLLABORATIVE DESIGN IN INDUSTRY

In this section, we present an example of industrial product design process and demonstrate the needs for collaborative live modifications of CAD data in an immersive environment.

3.1 Industrial Product Design

Industrial product design is a sequential process. For example, some automotive companies split this process in 3 main stages (see Fig. 1):

- Conceptual design stage: style designers sketch a preliminary draft of the product based on expectations and requirements of end users. They create a conceptual model using dedicated tools specialized at aesthetic design (e.g. Alias®, Autodesk).
- Detailed design stage: CAD engineers build a digital mock-up using parametric CAD software (e.g., CATIA®, SolidWorks®) from the conceptual model designed at the previous stage.
- Perceived quality analysis stage: stylists and CAD engineers tune rendering materials and textures on high quality meshes exported from the CAD data to create a realism-oriented virtual scene using a high-rendering system (e.g., Deltagen®). Stylists, CEO and other executives analyze the perceived quality of the car to validate the final digital mock-up. If modifications are required, it is mandatory to go back to the detailed design stage to apply these modifications on the CAD data.

At each design stage, review meetings are often conducted within immersive VR platforms to allow experts to review the virtual prototype at full scale and/or in a realistic environment.

The digital mock-ups are stored and exchanged through a PLM (Product Lifecycle Management) database under various formats according to the dedicated software used at each product design stage. Consequently, this data heterogeneity imposes conversions to pass from one stage to another. Also, the use of VR environments usually imposes time consuming conversions and/or transcriptions between the virtual environment and the PLM database.

3.2 Detailed Design

During review meetings of the detailed design stage, digital mock-ups based on CAD data are presented in the virtual environment to verify or compare different versions. However, as highlighted in section 2, current systems did not support live modifications of CAD

part parameters during these meetings because of the complexity of CAD data. In particular, CAD objects are defined by a set of operations (e.g. Extrusion, Sweep) applied on primitives and 2D sketches and based on a number of parameters and geometrical constraints. These CAD objects are generated by a commercial CAD engine outside the virtual environment and there is usually no direct link between the CAD engine and the VR system. Consequently, review meetings of the detailed design stage require data conversion before the VR session, creation of several versions of the 3D model, and post-modification of original CAD data afterwards. This back-and-forth between the virtual environment and the workstation is time consuming and a source of errors.

To avoid these issues, we want to provide a direct way to modify CAD data parameters during the collaborative review meetings of the detailed design stage. Consequently, we will focus on this stage as the main context of our collaboration system.

3.3 Collaborative Modification Scenario

Our system aims to make the following scenario possible. An industrial company is divided into several departments over the world. Stylists work at the parent company in Germany, while CAD engineers and ergonomists work in China. With our system, stylists could check the design of a car through a large screen with a high-resolution visualization system, while ergonomists evaluate the customer comfort (e.g. driver seat, cockpit space, field of view) in an immersive VR system. Both of them can discuss with CAD engineers also present in the CVE and ask them to achieve some modifications on the digital mock-ups. CAD engineers can directly modify the CAD data from the immersive system with a haptic device. Communication between remote sites is enhanced by streaming video and audio in the virtual environment.

4 SYSTEM ARCHITECTURE

To implement the proposed system, we designed a distributed architecture which allows a direct connection to the commercial CAD engine which manages the CAD data. This architecture should meet the following requirements:

- Modification of native CAD data at run time: the system should be able to load and modify existing CAD data without any data conversion before/after the VR session.
- Collaborative modification of CAD parts: multiple users should be able to modify either different parameters of a part, but also a same parameter of this part.
- Distribution of CAD data among remote locations: the system should dynamically update the CAD data on each remote platform.
- Interconnection between heterogeneous platforms: the system should deal with the various visualization systems and interaction devices of each platform.

To support these features, we developed an external server, named *VR-CAD Server*, which loads and modifies native CAD data according to user modification requests, and then transmits the resulting meshes of the CAD model to each platform. We also use a hybrid network architecture which combines both a centralized architecture to connect remote platforms to the *VR-CAD Server* and peer-to-peer connections to allow fast communication between platforms. This architecture is independent of the technical specifications of each platform and can connect heterogeneous platforms with different visualization system and interaction devices.

4.1 VR-CAD server

The *VR-CAD server* extends the cReaVR approach proposed by Martin et al. [8], and makes its remotely functional when a distributed architecture is required. cReaVR is implemented with C++

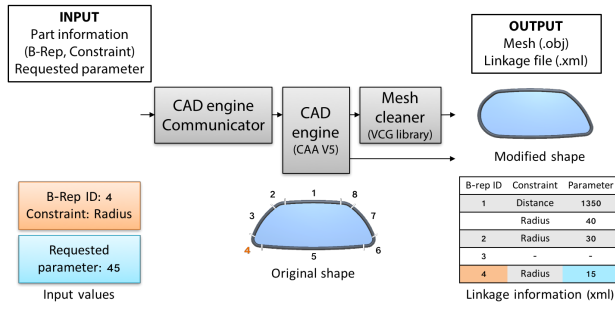


Figure 2: Data flow of the VR-CAD server. Meshes (.obj) and linkage information (.xml) are generated based on modification requests.

embedding CAA V5 (API for CATIA V5®). It can parse CHG of native CAD data and tessellate meshes for visual rendering. The base concept of cReaVR is *labeling* [3]: a direct linkage between 3D meshes of each B-Rep and each CHG node of the CAD object. With this linkage, users can implicitly access to the parameter values of the CHG nodes, which contribute to define the shape of CAD parts, by selecting one of the displayed meshes in the virtual environment.

The VR-CAD server relies on the data flow illustrated in figure 2. When the system starts, the VR-CAD server loads a specified CAD data (.catpart) and generates related meshes and a linkage information file (.xml). The meshes tessellated by the CAD software often have defects (e.g. non-manifold geometry), we thus clean these meshes using VCG library¹. The meshes are converted to Wavefront standard format (.obj) before deployment. For the modification of a CAD part, the VR-CAD server receives as input the information of selected part (B-Rep ID and constraint ID) and the new parameter value for the targeted constraint. Then, it sends back as output the computed meshes and a new linkage file.

For example, figure 2 details the behavior of the VR-CAD server when a user wants to change the curvature of a left bottom corner of the rear-view mirror of a car from 15 mm to 45 mm. First, the information “B – Rep ID = 4”, “Constraint = Radius” and the new “parameter value = 45” are transmitted from the user’s platform to the VR-CAD server. Then, the CAD engine communicator asks the CAD engine to edit the parameter of the specified CHG node with the new value. The CAD engine updates the whole CHG (e.g. relevant operators) and generate new meshes from the computed B-Reps. This process takes 700 ms on average for the rear-view mirror. After all this processing, the new meshes and the updated linkage file are delivered to each platform.

4.2 Distributed Architecture

We propose a hybrid network architecture to connect the remote platforms together and with the VR-CAD server (Fig. 3). All information about non-CAD objects (e.g. avatars, pointers) are transmitted with a peer-to-peer architecture between each platform. Whereas, all the data relative to CAD objects (meshes and linkage files) are transmitted with a client/server architecture between the platforms and the VR-CAD server.

A *Workspace* server (WS server in Fig. 3) is used for authentication and for establishing the connection between each platform and the VR-CAD server. This notion of *Workspace* allows to store a particular configuration of a work session and to retrieve it later in order not to have to redo the network configuration for each similar work session. Each user just has to connect to a specific *Workspace* to be connected to the other user. The WS client is the communication layer which manages the network communication on each platform and on the VR-CAD server. The communication between

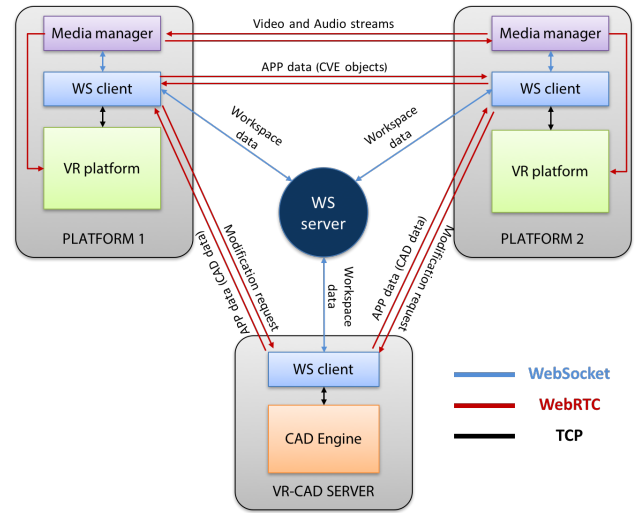


Figure 3: Distributed architecture for collaborative CAD data modification between remote VR platforms.

the WS server and the WS client is managed by WebSocket protocol, which allows real-time full-duplex communication.

The communication between each WS client is handled by WebRTC² protocol allowing peer-to-peer connections. We choose this open protocol to be able to connect various remote locations with different network architecture and deal with network constraints such as firewall and security issues. The WS client running on each platform streams application data to all the other platforms for synchronizing interaction events and non-CAD object positions in the CVE. The VR-CAD server also exchanges the CAD data or the modification requests with each platform through a WS client.

For the internal communication inside each platform, the WS client uses the Transmission Control Protocol (TCP) to interact with the VR platform or the CAD engine. The application sends information about the local elements (e.g. positions/orientations of user’s head and hand, events) to the local WS client which transmits update about the states of the CVE to the remote platforms. Conversely, the information of the remote platforms are transmitted to the local application by the local WS client. If a VR platform is composed of computer clusters (such as for CAVE systems), the WS client only communicates with a master node of its local clustering system.

Each platform can also stream media data (audio and/or video) with WebRTC protocol through a Media manager. The Media manager is connected via WebSocket to the local WS client which controls the media streams. The audio and/or video stream can thus be delivered to all the other remote platforms.

As the communication module is independent from the platform, this architecture can easily be adapted to several visualization systems and interaction devices. Figure 3 describes the architecture with two platforms, but additional platforms can be included with the same communication scheme.

4.3 Collaborative User Interaction

To achieve the parameter modification of a CAD part in the virtual environment, users must perform the following actions:

- **Selection:** the users start by picking a specific part of the 3D model using a virtual pointer. Once they have selected a part, they need to pick a specific constraint in a list of constraints (e.g. distance, radius) linked to the part.

¹<http://vcg.isti.cnr.it/vcglib/>

²<https://webrtc.org/>

- **Parameter modification:** the users can then modify the selected parameter value using a dedicated manipulation metaphor to increase/decrease the value.
- **Validation:** when the users validate their modification, the resulting modification request is sent to the *VR-CAD server* which updates the CAD data and distributes the new version over the network. If the value is not acceptable for the CAD engine (i.e. exceeds the limit of the constraints), the users receive an error and have to continue the parameter manipulation.

Each remote user can independently act on different CAD parameters, but we also want to allow multiple users to simultaneously modify the same parameter to support different cooperative activities. This cooperative manipulation could be adjusted by considering the expertise and roles of the collaborators. For example, a CAD engineer could assist the parameter manipulation conducted by a stylist to refine its modification by either providing possible discrete solutions or constraining its manipulation range in real time. The system can thus support two kinds of collaborative modifications:

- *Independent modifications* which occur when each user modifies a different parameter of the CAD data at the same time. Each user can see the visual proxy of the other users' hand, and can thus check their actions. When one of the users validates its parameter value, the CAD data will be updated regardless of the ongoing parameter manipulation on the other platforms.
- *Cooperative modifications* which happen when more than one user simultaneously select the same CAD data parameter. In our current implementation, we use a *Mean* technique by averaging the parameter values of each user after their validation. This is a simple way to combine users' action [10], but some other techniques have been proposed [1,9] and we can envision the use of a more sophisticated technique in the future.

5 PROOF OF CONCEPT

As a proof of concept, we use our distributed architecture for enabling two remote users to achieve a collaborative CAD data modification from two different VR platforms: a wall-sized display and a CAVE-like system (Fig. 4). The wall-sized display and the CAVE-like system have distinct visualization and interaction capabilities: a 2D high-quality rendering system with a touch interaction for the first one and a stereoscopic rendering system with a force feedback device for the second one. While both VR platforms are located in our campus, they are not on the same local network.

In this example, the remote users collaboratively modify the shape of a rear-view mirror while standing in the virtual cockpit of a car, instead of using the interface of a CAD software. As illustrated in section 3.3, a stylist could use the wall-sized display to check the quality of the product with a wide field of view and a high resolution visualization, while a CAD engineer could use the CAVE-like system to check the shape in 3D with a stereoscopic rendering.

6 CONCLUSION

This paper presents a distributed architecture allowing collaborative modifications on native CAD data from heterogeneous VR platforms. This architecture enables remote users to modify the parameter values of CAD parts during immersive reviews. Our system is based on a *VR-CAD server* which embeds a commercial CAD engine for loading and modifying native CAD data in a CVE. It implements a hybrid network architecture: the CAD data (meshes and linkage files) are distributed to each platform by the *VR-CAD server*, while information about non-CAD objects and media streams are directly transmitted between each platform. The *VR-CAD server* also manages concurrent access to CAD data parameters. It implements a cooperative modification mode in which several users can simultaneously modify the same parameter values of a CAD part.

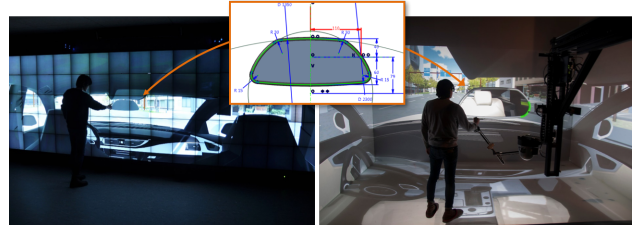


Figure 4: A proof of concept: native CAD data modification between a Wall-sized display (left) and a CAVE-like system (right).

We believe that direct modifications of CAD data in a CVE will facilitate and shorten review meetings at the detailed design stage of the industrial process. However, our system also has some limitations. In particular, it introduces a delay during modification of the CAD part because the 3D-mesh generation on the *VR-CAD server* and their distribution to all platforms takes time. After requesting a modification, users need to wait a few seconds before receiving the updated CAD data. We need to reduce this delay as much as possible, but the time required to generate meshes will likely remain proportional to the complexity of the CAD model. Consequently, we also need to work on appropriate feedback to enable users to understand and assess the delay.

As future work, we will investigate different interaction techniques for parameter modifications considering user skills and capabilities of their VR platform. We also want to explore the usage of media streams to enhance the communication between remote users.

ACKNOWLEDGMENTS

This work was partially supported by EquipEx DIGISCOPE (ANR-10-EQPX-26-01) operated by the French Agence Nationale de la Recherche.

REFERENCES

- [1] L. Aguerreche, T. Duval, and A. Lécuyer. Comparison of three interactive techniques for collaborative manipulation of objects in virtual reality. In *Proc. of Computer Graphics International (CGI)*, 2010.
- [2] R. Arangarasan and R. Gadh. Geometric modeling and collaborative design in a multi-modal multi-sensory virtual environment. In *Proc. of ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 10–13, 2000.
- [3] T. Convard and P. Bourdot. History based reactive objects for immersive CAD. In *Proc. of symposium on Solid modeling and applications*, pp. 291–296. Eurographics Association, 2004.
- [4] G. Lawson, D. Salanitri, and B. Waterfield. VR Processes in the Automotive Industry. In *International Conference on Human-Computer Interaction*, pp. 208–217. Springer, 2015.
- [5] V. D. Lehner and T. A. DeFanti. Distributed virtual reality: Supporting remote collaboration in vehicle design. *IEEE Computer Graphics and Applications*, 17(2):13–17, 1997.
- [6] J. Leigh, A. E. Johnson, C. A. Vasilakis, and T. A. DeFanti. Multi-perspective collaborative design in persistent networked virtual environments. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*, pp. 253–260. IEEE, 1996.
- [7] M. Mahdjoub, D. Monticolo, S. Gomes, and J.-C. Sagot. A collaborative design for usability approach supported by virtual reality and a multi-agent system embedded in a PLM environment. *Computer-Aided Design*, 42(5):402–413, 2010.
- [8] P. Martin, S. Masfrand, Y. Okuya, and P. Bourdot. A VR-CAD data model for immersive design. In *Proc. of Int. Conf. on Augmented Reality, Virtual Reality and Computer Graphics*. Springer, 2017.
- [9] M. S. Pinho, D. A. Bowman, and C. M. Freitas. Cooperative object manipulation in collaborative virtual environments. *Journal of the Brazilian Computer Society*, 14(2):54–67, 2008.
- [10] R. A. Ruddle, J. C. Savage, and D. M. Jones. Symmetric and asymmetric action integration during cooperative object manipulation in virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 9(4):285–308, 2002.