Scenario-based VR Application for Collaborative Design

Romain Terrier[1,2], Valérie Gouranton[1,2], Cédric Bach[1,3], Nico Pallamin[1] and Bruno Arnaldi[1,2]
[1] *IRT bcom, Cesson-Sévigné, France*
[2] *Univ Rennes, INSA Rennes, Inria, CNRS, IRISA, Rennes, France*
[3] *Human Design Group (HDG), Toulouse, France*

Keywords: VR, Framework, Scenarios, Collaborative Design.

Abstract: Virtual reality (VR) applications support design processes across multiple domains by providing shared environments in which the designers refine solutions. Given the different needs specific to these domains, the number of VR applications is increasing. Therefore, we propose to support their development by providing a new VR framework based on scenarios. Our VR framework uses scenarios to structure design activities dedicated to collaborative design in VR. The scenarios incorporate a new generic and theoretical collaborative design model that describes the designers' activities based on external representations. The concept of a common object of design is introduced to enable collaborations in VR and the synchronization of the scenarios between the designers. Consequently, the VR Framework enables the configuration of scenarios to create customized and versatile VR collaborative applications that meet the requirements of each stakeholder and domain.

1 INTRODUCTION

Virtual reality (VR) is largely used to support design in industry (e.g., product design, interior design). However, depending on the designers and the sector needs, each VR application has to be personalized and has to offer various design functionalities. The development time is becoming longer due to the requirement to script new behaviors from the beginning and devise new rules for each need. As an alternative to building each application from scratch, in this paper, we propose a unique VR Framework based on scenarios to create various VR multi-user applications for collaborative design.

VR is an appropriate tool that the designers can use to cooperate. The shared and interactive environments enable the designers to work on complex systems (Wang et al., 2019). There is a great variety of VR applications that can be utilized to work on these complex systems. This variety is explained by the necessity to take into account the knowledge and the work tools of multiple stakeholders in VR, and by the integration of the different types of collaborations that occur during the design process (Falzon et al., 1996). The reasons for this diversity highlight the complexity of building VR applications for effective collaborative design and underline the necessity of supporting the development of such applications.

To support the development of VR applications for collaborative design, we propose a new VR multi-user Framework that offer several services to the developers (Fig. 1). The Framework is based on scenarios to sequence and to define the services in compliance with our theoretical design model.

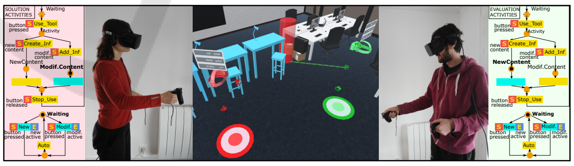

Figure 1: Virtual reality (VR) framework based on scenarios for collaborative design applications.

First, we describe the generic design activities and the levels of collaboration in a theoretical model. Then, the theoretical model is incorporated into various scenarios "to depict all the possible sequencing" of design activities in VR (Claude et al., 2015). The construction of the scenarios is based on Petri-Net, which enables the developers to adjust this process in the event of changes to the theoretical model. Finally, the new concept of a common object of design is implemented in the scenarios as a service to the developers to enable different collaborations during the design process, including functionalities to display design state information in VR.

Terrier, R., Gouranton, V., Bach, C., Pallamin, N. and Arnaldi, B.
Scenario-based VR Application for Collaborative Design.
DOI: 10.5220/0010268602370244
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 1: GRAPP*, pages 237-244
ISBN: 978-989-758-488-6

In this way, the developers are provided with a tool to create VR multi-user and personalized applications for collaborative design.

2 RELATED WORK

This section presents the state of the art of the various design activities, that are part of a general process, and the collaboration between designers. These activities, collaborations and processes are structured into theoretical design models that are subsequently outlined. Finally, multiple Frameworks are presented to facilitate the development of VR applications based on activities.

2.1 Design: Activities and Process

The design process could be defined either as a problem-solving process (Simon, 1995) or as a situated and reflective practice (Schön, 1991). Regardless of the definition adopted, the design process is composed of a succession of *design activities* (Gero and Neill, 1998) during which the designer's knowledge is reflected in practice (Schön, 1992). These activities act on and occur in the designer's mind or in the physical world, that is, the activities are internal or external (Zhang, 1991). The designers use the external activities to materialize their internal representations (Eastman and Computing, 2001), and the external activities are easier to implement in a VR application than internal activities.

In the literature, scholars mostly divide the design process into two main spaces, namely the problem space and the solution space (Lonchampt et al., 2006). A third space, called the evaluation space, has been introduced to characterize the activities involved in the evaluation of a design (Terrier et al., 2020). The designers perform multiple sequences of activities to refine the design (cos, 2003). The three spaces are linked (Brissaud et al., 2003), and they co-evolve. Thus, all the spaces are refined together, and the design process ends when the final state of compromise is reached (Simon, 1995). Regardless of the design process and the design domains involved, the problem, the solution, and the evaluation are shared concepts among the design domains.

The users' design actions are described by activities that produce external representations that are categorized into generic design spaces.

2.2 Collaboration and Cognitive Synchronization

In the collaborative design process, the design activities involve several designers cooperating simultaneously. In VR, this cooperation is classified into three levels (Margery et al., 1999). In level 1, the users are able to see each other and to communicate. In level 2, the users are able to act on the scene by changing the scene individually. In level 3, the users are able to act on the same entities in the scene. But these levels are never used to describe the collaborative design process. The users' success depends on their ability to establish a cognitive synchronization (Détienne, 2006). Thus, in VR collaboration, the design systems have to support communications between the designers and provide a mutual workspace that enables a "shared understanding of the design artifact among a design team" (Saad and Maher, 1996). A deficiency in shared understanding may lead to misinterpretation, or annoyance, consequently slowing down the design process (Valkenburg, 1998). Representation and interaction metaphors can be utilized to avoid this drawback. The relevant ideas and knowledge are externalized through the metaphors, thus facilitating and supporting communication between the designers (Perry and Sanderson, 1998). For these reasons, common descriptions and explanations of a system are key factors for effective collaboration, which emphasizes the necessity of synchronicity and communication between the designers (Smart et al., 2009).

External representations are important for sharing information to establish a shared understanding between the designers in VR and for easing communication between them, which could also be supported by the use of a verbal channel in virtual environments (VE) (Gabriel and Maher, 2002).

2.3 Theoretical Design Models

Different models have been proposed to structure and understand the design process. During the design process multiple iterations occur, and these aspects of design can be described by models. Some models envisage design as consisting of multiple activities (Girod et al., 2003) without defining it as involving a series of events. Others group activities by family and link the activities with events to describe a generic design process (Gero and Kannengiesser, 2004). All of these models enable iterations and describe activities, but none depict collaboration.

To overcome this gap in the existing models, the individual function-behaviour-structure (FBS) model has been extended for collaborative design (Gero and

Milovanovic, 2019). The junction between the individual activities occurs in the external world, thereby allowing the development of shared understanding. The FBS model is a powerful tool for understanding and encoding the collaborative design mechanism, but it describes internal and external activities. The internal activities occur in the mind of the people, thus the development of these activities are not possible. A generic model has been proposed for individual design describing the co-evolution of the problem, solution, and evaluation space (Terrier et al., 2020). This model describes generic activities categorized according to the three spaces delineated in Section 2.1. Moreover, the described activities acts on external representations and have been used to structure VR activities. Nevertheless, this model does not describe collaborative activities, the common understanding process, and the three levels of collaboration. A solution to this limitation is to extend this individual model following the multi-user FBS extension.

2.4 Framework for VR

The development of VR applications requires support provided by generic and reusable systems (Mollet and Arnaldi, 2006). Several frameworks exist. MASSIVE (Greenhalgh and Benford, 1995) enables the developers to immerse the users into a shared virtual environment with rules to define different collaborative states. Another framework (Gonzalez-Franco et al., 2015) enables the developers to build multiuser application that engages the users in three levels of collaboration according to their proximity of each other. But the developers cannot use these frameworks to drive and to sequence the activities of the users, and to implement personal tools into the application. VHD++ (Ponder et al., 2003) enables the developers to create and use personal tools.

Another solution is to use scenarios. The scenarios are able to fully constrain, partially constrain, or completely free the users' actions by listening to and interacting with the VE. A graphical representation can be used to model scenarios, for example, the Petri-Nets in #SEVEN (Claude et al., 2014). In this way, communication is facilitated between the developers and the users. An advantage of the scenarios-based approach lies in the fact that the actions described are similar to the external activities depicted in the theoretical design models. Here, the actions described are the interactions between the users and the objects or the virtual world. In addition, the scenario of each user can run independently from one user to another, allowing the users to temporarily work individually (Jota et al., 2010). Consequently, the scenarios are able to reflect the collaborative design activities of multiple users and enable iterations. Solutions exist to create a Framework for collaboration in VR, but none of these solutions have been used for collaborative design based on generic activities occurring in VR.

The related works surveyed in this section reveal the need to propose a new theoretical collaborative design model that depicts external activities categorized in the three spaces. To the best of our knowledge, the scenarios have not previously been used to describe a collaborative theoretical model with the goal of structuring collaborative design activities in a synchronized VR application.

3 SOLUTION OVERVIEW

In this paper, the VR Framework supports the developers' work towards building personalized, multitool, and multi-user VR applications intended to meet the needs of multiple design domains. The solution needs to drive the generic users' actions in VR independently of the domain to structure the collaborations of and to support the communication between designers in VR.

Consequently, the solution we propose here uses a theoretical design model to define the generic design activities of several users. The collaboration is modelled through the common object of design (Fig. 2, part 1). Next, the theoretical model is naturally described by the three-level scenarios in a VR Framework, with each level managing functionalities making possible specific activities in the collaborative design process (Fig. 2, part 2). For example, a user is able to select an object for the team and each user is able to perform evaluation or to generate new solutions. Finally, the implementation of the scenarios in a VE leads to the production of VR applications for industrial design (Fig. 2, part 3).

For example, the developer configures the scenario to provide access to specific tool metaphors to enable the creation or modification of a 3D object in VR (Fig. 1). Scenarios are the cornerstone of our VR Framework. All the design activities are correlated to the use of a set of design tools. Each user has their own independent instance of the scenarios. The scenarios can be adapted to depict other models and alternative spatial or conceptual organizations of the VE.

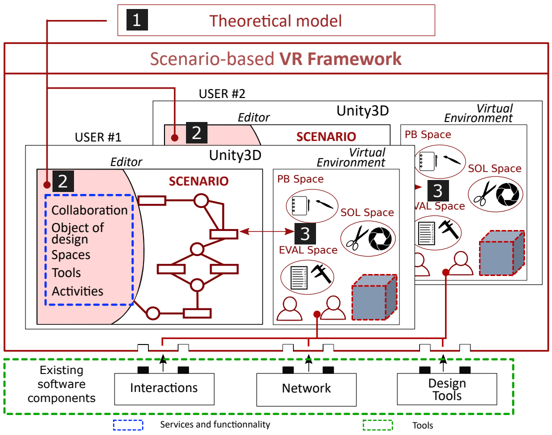

Figure 2: In red, our contribution: (1) a new theoretical collaborative design model defining generic design activities and collaborations, (2) the three-level scenarios depicting the new theoretical model, providing services to the developers (blue dots) and enabling the implementation of tools (green dots), and (3) an implementation of the scenarios in a virtual environment (VE).

4 THEORETICAL COLLABORATIVE DESIGN MODEL

Our multi-user VR Framework is driven by a generic and theoretical collaborative design model to cover the needs of various domains. It is composed of a multiple single-user theoretical model linked by design objects and logical space for collaboration.

4.1 Single User Dimension

The model depicts the three logical design spaces: problem, solution, and evaluation (Terrier et al., 2020). In each logical space, two activities are distinguishable: the creation and the iteration of contents. In addition, the evaluation logical space includes the act of evaluating and the act of defining or modifying the criteria of evaluation. The next step is to depict the multi-user dimension in the theoretical model.

4.2 Multi-user Model

As the collaborative design implies multiple designers, we propose to use one instance of the theoretical single user design model for each designer. Since the activities are based on the creation of external representations, the internal mental process of building of a shared understanding is not represented in our multi-user model. However, the collaboration and the shared understanding can be facilitated. To this end, the model introduces the concept of a *common object of design*. This concept describes the object of design on which the designers are working in unison. Once a common object is defined, all the activities of each designer focus on this specific object. In this way, the collaboration level 3 occurs between the designers (Fig. 3, top). For example, two designers are able to modify the same mockup of a room to propose new solutions. The model also describes the collaboration level 2, which occurs when no common object is selected, thereby allowing the designers to interact with any object of the mockup. In this situation, the designers still collaborate to refine the design but are not constrained to work on the same object (Fig. 3, down). For example, one designer is able to modify a workstation while the second moves screens around the room to adjust together the overall layout of the room.

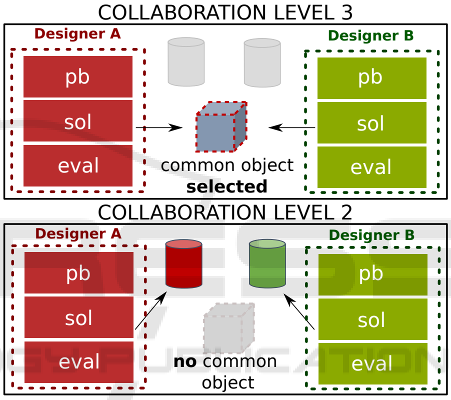

Figure 3: The theoretical collaborative design model introducing the common design object. It depicts, top, the collaboration level 3, and down, the collaboration level 2.

Thus, our theoretical collaborative design model is able to depict various collaborative design situations and their associated activities. The next step is to depict this model with a scenario model that will be used by developers and interpreted by a scenario engine to play the events in the VE.

5 SCENARIO MODEL FOR COLLABORATIVE DESIGN

5.1 Scenario Details

Our VR Framework uses a scenario creation tool integrated into Unity (Claude et al., 2014). Based on a Petri-Nets (Petri, 1962) language, the scenarios are edited into Unity with graphical representations to model user events, environment states, and object behavior. The scenario model is a graph depicting a series of events made up of places, transitions, and subscenarios. The scenario model is able to listen to and modify the state of the VE through sensors and effectors. The associated scenario engine is able to play the events in the VE and, thus, to sequence the design actions (e.g., collaboration level and common object selections) depending on conditions (e.g., roles).

5.2 Three-levels Scenario Model

The theoretical collaborative design model is split into three levels of scenarios (see Fig. 4). An instance of the three-levels scenarios is attached to each user.

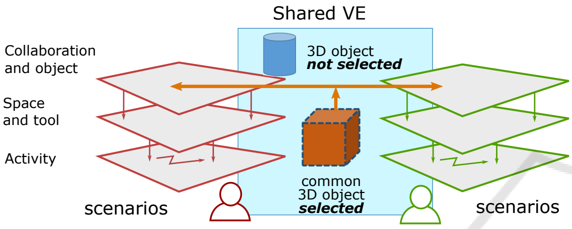

Figure 4: Three-Level scenarios depicting the theoretical collaborative design model. An instance of the three-level scenarios is attached to each user. The synchronization between each instance is performed through the common 3D object in the VE.

The first level manages the functionalities that the developers can use to enable selection of the common object of design in the VE by the designers. The framework provides a service of synchronization that occurs only between each first level and that depends on the state of the selection of a common design object in the shared VE. As described in the theoretical model, the collaboration level 3 occurs when a common object is selected. All of the activities are performed on this object. The framework integrates a service based on roles for developers to enable the functionality of selection only for the “supervisor”. The other role can only select a design object when the collaboration is level 2 (i.e., no common object is selected). For example, the service enable only the “supervisor” to select the collaboration mode level 3 or level 2. Once the selection is executed, the selection state is detected by all the instance of scenarios at the same time engaging all the designers to work jointly or independently. If the level 3 is activated by the scenarios for each designers, only the “supervisor” is able to select the common object of design. The services enable the designers to change their selections (i.e., collaboration mode and object).

The second level manages the functionalities that the developers can use to enable the users to access one of the three logical spaces of design (i.e., problem, solution, or evaluation) and to pick a tool. For example, independently of the choice of the other designers, each scenario is able to provide the capacity for a designer to select the solution space and a virtual pen to later propose a new solution of a workstation. This level is activated only when an object is selected. The scenario enables the developers to allow the designers to switch from one space to another and to switch from one tool to another.

The third level manages the functionalities that the developers can use to enable the users to access the design activities (i.e., creation and modification) of the space. The level is activated only when a space and a tool are selected. The developers are able to constrain the user to select one activity at a time. The scenario enables the designers to switch between activities for a single tool.

5.3 Technical Details

Many items need to be synchronized between each user’s applications: the scenario, the VE state, the avatars, and the common design object state. Besides making sure that things are running smoothly, the synchronization supports the building of a shared understanding between the designers. The framework implement an existing software component to provide this service to the developers. The networking is managed by the component TNET3 [1]. Moreover, the framework meets the developers’ need to share information regarding each user’s own activity state to the others in the VE by adding effectors to provide these awareness information. This information takes the shape of a text displayed above the user’s avatar with the following information: the space of the activity, the tool in use, and the current activity performed by the user (see Fig. 5). All this awareness information is updated according to the user’s scenario. The networking setup also enables the developers to implement audio sharing among the designers. In this way, the VR Framework enables distant and co-localized collaborations.

6 USE CASE

The use case stems from the industrial partner’s activities, which deal with the collaborative design of new working spaces. In this use case, the developers immerse two designers in a VE and to reflect on a layout (see Fig. 5), considering that the surface of the room remains unmodified and that a workforce increase of

---

[1] TNET3: www.tasharen.com

50% is expected over the next 5 years. Based on these details, the designers propose and analyze new layout configurations. The following sections present: (1) the designers' series of events in the VE and (2) the functioning of the VR application, including the link between the VE and the scenarios.

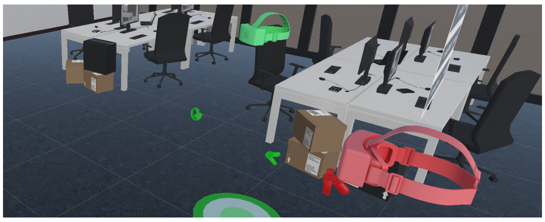

Figure 5: The collaborative design VE scene for the use case of the working space layout.

6.1 Designers Activities

In the application, the two designers propose multiple layouts according to the relevant specifications and norms. In this case, the problem is already defined and the designers focus on the creation or the modification of solutions. The designers focus also on the creation or the modification of evaluation according to the solutions they have proposed.

Considering the design activities, the designers need tool metaphors to interact in the VE. Thus, the developers are able to implement several tools in the application to move the objects, to navigate among space configurations, to change the material, to check the safety space between the furniture, to simulate the different users' heights, to add points of interest, and to have a top view.

Before building the application, the developers are able to attach the "supervisor" role to one instance of the application to only allow one designer to select a common object once the application is running. The designer is then able to show the different space configurations to the other designer. The collaboration level 3 is activated.

Meanwhile, the service in charge of the space management enables the developer to create an application that let the users to switch between evaluation and solution activities during the session, for example, to iterate on the materials of the common object and check the global harmony of the room.

By using the common object and collaboration services, the developers are able to build an application that let the "supervisor" designer switching from one common object to another or to change the collaboration mode for the level 2.

At that point, both designers are able to select distinct design objects. Each designer performs activities on a different design object with the same goal of proposing a new layout that takes into account the anticipated workforce increase and the relevant specifications.

The framework and the TNet3 component are able to synchronize all modifications of the VE between each instance of Unity.

6.2 Scenarios and VE Relationships

The following describes the step during which one designer switches between configurations, while the second evaluates the evacuation criterion.

Space and Tool Detection. The developers enable each user to navigate between the three logical spaces thanks to the second level of scenarios. The second level of scenarios and the space service enable the developers to allow each user to navigate between the three logical spaces : *Problem*, *Solution*, and *Evaluation*. Each logical space contains a batch of dedicated tools implemented by the developers. A sensor detects that the user has selected the solution space, and an effector activates the third level of scenarios (see Fig. 6). In this way, the tool selection functionality is activated enabling the user to pick the configuration tool. Simultaneously, the second user selects and activates the tool to evaluate the safety distance between objects.

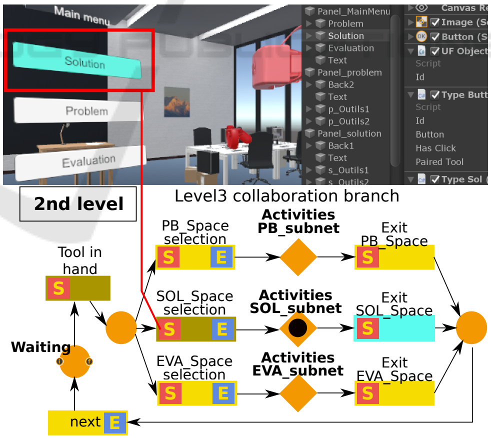

Figure 6: Left: the space selection functionality is displayed to the user. Right: the second level of scenarios that manages the selection of the space. Since the space is selected, the scenario activates the third level.

Depicting the Design Activities. The configuration tool is activated for the first user, and the possible activities are depicted in a branch of the third level of scenarios, *Solution Activities* (see Fig. 7). The functionalities of this level enable the user to modify the configuration of the object (e.g., a workstation). A sensor detects the action, while another sensor displays the activity over the user's head to inform the second user about this action. The position and the active state of each workstation configuration are synchronized between the Unity instances.

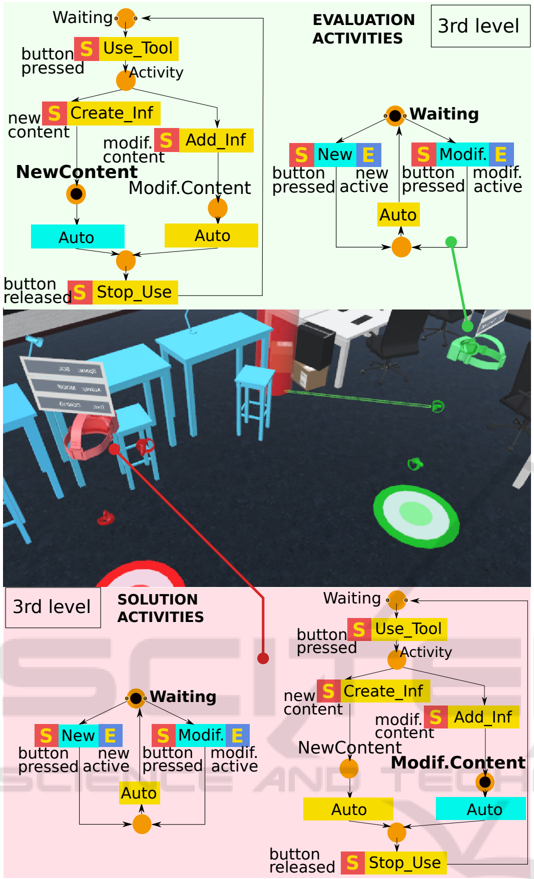

Figure 7: Up: the green user's third level in the *Evaluation Activities* branch is active. Down: the red user's third level in the *Solution Activities* branch is active. It has detected the use of a solution tool to modify contents. Thus, the token in the Modif. content place indicates that the user is modifying the current mock-up. The transitions are colored blue as they are activated but not triggered.

The scenario is thus able to discriminate between the designers' activities, their space, the tools involved, and the creation or modification of contents. Finally, the three-level scenarios enable design iterations on the same object for a level-3 collaboration.

In a customization context to meet the needs of an industry, new VR tools and/or activities can be implemented in the model. The developers are also able to use only a specific part of the scenario to constrain the users' activities during a design session (e.g., adding or deleting branches, transitions, etc.).

7 CONCLUSION

Our solution supports the development of VR collaborative design applications involving multiple designers. The Framework implements scenarios and services according to a theoretical collaborative design model to drive the users' activities in VR. In this paper, only the functionalities of the Framework are described and illustrated in a use case.

The collaboration is supported by two functionalities. The Framework works over the network by being implemented individually for each user. The VE, the communications, and the scenarios are synchronized between each instance of the Framework. Then, the concept of a common object of design enables the users to switch between types of collaboration. The users are consequently able to work on the same object or on different objects with the same goals.

Moreover, our solution meets the requirement to personalize the applications dedicated to design. The Framework enables the implementation of various design tools in VR without modifying the scenarios. The functionalities to access the activities remain operational independently of the design tools.

Presently, generic activities have been implemented and the developers are able to personalize the scenarios. However, a new functionality could be implemented to enable the selection of the activities needed by the designers, the selection of each user's role during the design session, and the selection of restricted access concerning specific activities according to the user's role. At this point, an evaluation of the use of the Framework is needed in comparison to other existing Frameworks. Moreover, the concurrent edition of the same object is not yet managed by our Framework and should be integrated in a future version. The persistence of the scene and the saving/loading are already implemented via the network asset TNET3 but they need to be implemented as functionalities in the Framework.

To conclude, our scenario-based Framework supports the development of various collaborative VR applications for design and can be upgrade to manage more functionalities.

REFERENCES

(2003). *Iteration in Engineering Design: Inherent and Unavoidable or Product of Choices Made?*, volume Volume 3b: 15th International Conference on Design Theory and Methodology of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.

Brissaud, D., Garro, O., and Poveda, O. (2003). Design process rationale capture and support by abstraction of criteria. *Research in Engineering Design*, 14(3):162–172.

Claude, G., Gouranton, V., and Arnaldi, B. (2015). Versatile scenario guidance for collaborative virtual environments. In *Proceedings of the 10th International Conference on Computer Graphics Theory and Applications*, GRAPP 2015, pages 415–422. SCITEPRESS - Science and Technology Publications, Lda.

Claude, G., Gouranton, V., Berthelot, R. B., and Arnaldi, B. (2014). Short paper: #seven, a sensor effector based scenarios model for driving collaborative virtual environment. In *Proceedings of the 24th International Conference on Artificial Reality and Telexistence and the 19th Eurographics Symposium on Virtual Environments*, ICAT - EGVE '14, pages 63–66, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Détienne, F. (2006). Collaborative design: Managing task interdependencies and multiple perspectives. *Interacting with Computers*, 18(1):1–20.

Eastman, C. and Computing, D. (2001). Chapter 8 - new directions in design cognition: Studies of representation and recall. In Eastman, C. M., McCracken, W. M., and Newstetter, W. C., editors, *Design Knowing and Learning: Cognition in Design Education*, pages 147 – 198. Elsevier Science, Oxford.

Falzon, P., Darses, F., and Béguin, P. (1996). Collective design process. In *Proceeding of COOP 96, Second International Conference on the Design of Cooperative Systems.*, pages 43–59.

Gabriel, G. C. and Maher, M. L. (2002). Coding and modelling communication in architectural collaborative design. *Automation in Construction*, 11(2):199 – 211. ACADIA '99.

Gero, J. and Milovanovic, J. (2019). The situated function-behavior-structure co-design model. *CoDesign*, 0(0):1–26.

Gero, J. S. and Kannengiesser, U. (2004). The situated function behaviour structure framework. *Design Studies*, 25(4):373 – 391.

Gero, J. S. and Neill, T. M. (1998). An approach to the analysis of design protocols. *Design Studies*, 19(1):21 – 61.

Girod, M., Elliott, A. C., Burns, N. D., and Wright, I. C. (2003). Decision making in conceptual engineering design: An empirical investigation. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 217(9):1215–1228.

Gonzalez-Franco, M., Hall, M., Hansen, D., Jones, K., Hannah, P., and Bermell-Garcia, P. (2015). Framework for remote collaborative interaction in virtual environments based on proximity. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 153–154.

Greenhalgh, C. and Benford, S. (1995). Massive: A collaborative virtual environment for teleconferencing. *ACM Trans. Comput.-Hum. Interact.*, 2(3):239–261.

Jota, R., de Araújo, B. R., Bruno, L. C., Pereira, J. M., and Jorge, J. A. (2010). Immiview: a multi-user solution for design review in real-time. *Journal of Real-Time Image Processing*, 5(2):91–107.

Lonchampt, P., Prudhomme, G., and Brissaud, D. (2006). *Supporting Problem Expression within a Co-evolutionary Design Framework*, pages 185–194. Springer London, London.

Margery, D., Arnaldi, B., and Plouzeau, N. (1999). A general framework for cooperative manipulation in virtual environments. In Gervautz, M., Schmalstieg, D., and Hildebrand, A., editors, *Virtual Environments '99*, pages 169–178, Vienna. Springer Vienna.

Mollet, N. and Arnaldi, B. (2006). Storytelling in virtual reality for training. In Pan, Z., Aylett, R., Diener, H., Jin, X., Göbel, S., and Li, L., editors, *Technologies for E-Learning and Digital Entertainment*, pages 334–347, Berlin, Heidelberg. Springer Berlin Heidelberg.

Perry, M. and Sanderson, D. (1998). Coordinating joint design work: the role of communication and artefacts. *Design Studies*, 19(3):273 – 288.

Petri, C. A. (1962). Kommunikation mit Automaten. Dissertation, Schriften des IIM 2, Rheinisch-Westfälisches Institut für Instrumentelle Mathematik an der Universität Bonn, Bonn.

Ponder, M., Papagiannakis, G., Molet, T., Magnenat-Thalmann, N., and Thalmann, D. (2003). Vhd++ development framework: towards extendible, component based vr/ar simulation engine featuring advanced virtual character technologies. In *Proceedings Computer Graphics International 2003*, pages 96–104.

Saad, M. and Maher, M. L. (1996). Shared understanding in computer-supported collaborative design. *Computer-Aided Design*, 28(3):183 – 192.

Schön, D. A. (1991). *The reflective practitioner: How professionals think in action*. Routledge, London.

Schön, D. A. (1992). Designing as reflective conversation with the materials of a design situation. *Knowledge-Based Systems*, 5(1):3 – 14.

Simon, H. A. (1995). Problem forming, problem finding and problem solving in design. *Design and Systems general application of methodology*, 3:245–257.

Smart, P. R., Mott, D., Sycara, K., Braines, D., Strub, M., and Shadbolt, N. R. (2009). Shared understanding within military coalitions: A definition and review of research challenges. In *Knowledge Systems for Coalition Operations*. Event Dates: 31st March-1st April 2009.

Terrier, R., Gouranton, V., BACH, C., Pallamin, N., and Arnaldi, B. (2020). Scenario-based VR Framework for Product Design. In *VISIGRAPP 2020 - 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 1–8, Valletta, Malta.

Valkenburg, R. C. (1998). Shared understanding as a condition for team design. *Automation in Construction*, 7(2):111 – 121. Models of Design.

Wang, P., Zhang, S., Billinghurst, M., Bai, X., He, W., Wang, S., Sun, M., and Zhang, X. (2019). A comprehensive survey of ar/mr-based co-design in manufacturing. *Engineering with Computers*.

Zhang, J. (1991). The interaction of internal and external representations in a problem solving task. In *Proceedings of the thirteenth annual conference of cognitive science society*, pages 954–958. Erlbaum Hillsdale, NJ.