Scenario-based VR Framework for Product Design

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

Keywords: Scenario, Framework, VR, Design, Model.

Abstract: Virtual Reality (VR) applications are promising solutions in supporting design processes across multiple domains. In complex systems (e.g., machines, cities, interior layouts), VR applications are used alongside Computer Assisted Design (CAD) systems which are (1) rigid (i.e., they lack customization), and (2) limit the design iterations. VR systems need to address these shortcomings so that they can become widespread and adaptable across design domains. We thus propose a new VR Framework based on scenarios and a new generic theoretical design model to assist developers in creating versatile and personalized applications for designers. The new generic theoretical model describes the common design activities shared by many design domains, and the scenario depicts the design model to allow design iterations in VR. Through scenarios, the VR Framework enables creating customized copies of the generic design process to fulfill the needs of each design domain. The customization capability of our solution is illustrated on a use case.

1 INTRODUCTION

In today's environment, the design of products, services, and software is omnipresent across sectors and each domain has its own design specificities. In complex systems, such as machines, cities, or interior layouts, multiple Computer-Aided Design (CAD) systems are used to assist designers. Instead of creating an application for each domain, in this paper we propose a solution for both developers and designers. First, our solution supports the developers in building versatile Virtual Reality (VR) applications for designers. Second, it supports the design iteration loops in VR.

The goal of design is "to give shape to an artifact–the product of design. This artifact is the result of a complex of activities–the design process" (Moran and Carroll, 1996). Design problems are complex and require vast knowledge and multiple stakeholders (Arias et al., 2000), causing an outbreak in the development of CAD systems.

VR is a promising solution, which could assist designers in their activities across multiple domains, including complex mechanical systems, architecture, interior layouts. VR can either be integrated into the design process and combined with CAD systems as an extension or used as a separate application. From concept formulation to component validation, VR can be used at different stages of product maturity. The advantage of VR over non-immersive displays is that it assists in more rapid error detection (Satter and Butler, 2015). Independently of the domain, VR is used in design to import and explore multiple configurations of 3D models to shorten the design cycle (Zawadzki et al., 2018). Moreover, the designers are easily able to modify the 3D models in VR, with regard to, for instance, size and position. Nevertheless, these VR solutions are dedicated to specific uses (e.g., bus configurator) and they cannot follow the evolution of designers' needs in the design process. Additionally, in some applications, designers are not able to modify the 3D model in VR, but they need to do so by returning to the CAD system on their desktop computers. Moreover, the description of the design process also assists application developers to structure their work by listing and sequencing the events and the actions of end users.

We thus propose a solution with a two-fold goal: (1) to support developers in the development of versatile VR applications independently of the targeted domains, and (2) to allow designers to perform their activities in virtual environments (VEs) using metaphors and tools.

Thus, we propose a new VR Framework based on

Terrier, R., Gouranton, V., Bach, C., Pallamin, N. and Arnaldi, B.
Scenario-based VR Framework for Product Design.
DOI: 10.5220/0008880602190226
In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020) - Volume 1: GRAPP*, pages 219-226
ISBN: 978-989-758-402-2; ISSN: 2184-4321

scenarios. In a VR context, scenarios are used "to depict all the possible sequencing of events allowed in the virtual environment" (Claude et al., 2015). In matching the design domains, the design activities are described in a scenario following a generic theoretical design model. In this way, the developers are able to use the generic base to build dedicated applications employed by designers in performing specific activities. The application, which is beyond the scope of the present paper, is personalized due to the implementation of specific VR design tools. Through scenarios, the VR Framework enables modifications for improving the VR design applications. We illustrate our solution by implementing the generic theoretical design model through scenarios that drive user activities, and through situating the activities in a VE with integrated design tools (see Fig. 1).

In this paper, we use the term *design process* for the overall action of designing, *design activity* for a particular action performed by a designer, and *design tool* for a tool used by designer to perform an activity (e.g., a pen to sketch). This paper is structured as follows. Section 2 presents related works. Section 3 describes an overview of our solution. Section 4 presents a new generic theoretical design model. Section 5 explains how the model of scenario is operating. Section 6 describes the scenario of our theoretical design model. Section 7 presents a use case. Finally, conclusions are drawn in section 8.

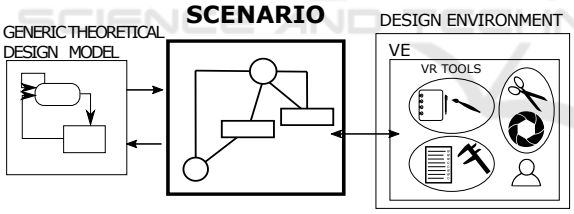

Figure 1: Framework for the development of VR applications for design. From a new theoretical design model, to its description by scenarios in order to manage user activities and VR tools.

2 RELATED WORK

2.1 Design: Activities, Process, and Models

The designing process is composed of activities (Gero and Neill, 1998). Two major paradigms describe this design process (Visser, 2006), namely (1) design as problem-solving (Simon, 1995), (2) design as a situated and reflective practice (Schön, 1991). The first paradigm is based on the division of a complex and ill-defined problem into multiple simpler ones. The second paradigm describes the design activities as context-dependent; the designers' knowledge is reflected in practice rather than verbally transmitted (i.e., knowing-in-action) (Schön, 1992). In each paradigm, the design process ends when a compromise regarding the solution is reached on the solution, which is then enacted through a succession of designers' actions, the so-called *design activities*. Thus, the design is not possible without activities. These activities are internal, i.e., conceptual, or, conversely, external,i.e., "in the world" (Zhang, 1991). The internal representation conjured by designers is materialized through external representation (Eastman and Computing, 2001). The activities impacting external representations (e.g., sketching a flat) are easier to implement than activities relying on internal representations (e.g., conceptual construction of a 2D plan).

Although there are multiple descriptions of the design process, to date, no standard has been defined. Many descriptions present the design process by splitting the activities in two spaces: the problem space and the solution space (Lonchampt et al., 2006). To reach a compromise, the problem and the solution need to be refined through multiple iterations. This process is referred to as the evolution of the spaces. Moreover, the spaces co-evolve (Maher et al., 1996): when the solution is refined, so is the problem. Solutions and problems are linked through evaluation (Brissaud et al., 2003). Multiple evaluations facilitate design refinement until the design process reaches its final state of compromise (Simon, 1995). The evaluation is based on criteria that can evolve during the design process. In this way, evaluation can be understood as the third space of design activities, next to the problem and the solution, however, no model presently describes it as such. Regardless of the design process, and the design domains, the problem, evaluation, and solution are shared concepts.

To gain a better understanding of the design process, researchers proposed its multiple descriptions usually in terms of the process activities or phases (Evbuomwan et al., 1996). Whereas some models describe the process as a succession of juxtaposed phases (Pahl et al., 2007), such descriptions are subject to several limitations. First, the procedural plan should not be followed strictly, as iterations often occur between phases. Second, simultaneous activities are not described. The models based on concurrent engineering overcome the latter limitation (Jo et al., 1991). Nevertheless, the iteration limitation remains if the model describes sequential phases due to their "linearity and their rigidity" (Pugh, 1986). An alternative description of the process envisages de-

sign as multiple activities (Girod et al., 2003) without defining it as a series of events. In this way, iteration is possible. Some models group activities by family and link the activities with events to describe a generic design process (Gero and Kannengiesser, 2004). Such models, do not presuppose linearity, iterations are possible, and the activities are described. These models can subsequently be used to structure the actions of designers in design applications. Nevertheless, the theoretical models do not suffice without implementation. In the following sub-section, we propose facilitating and structuring the development of VR applications.

2.2 Framework for VR

The development of VR applications, which can be fastidious, requires support (Kessler et al., 2000). Generic and reusable systems (Mollet and Arnaldi, 2006), such as system-independent applications (e.g., VRJuggler (Bierbaum et al., 2001)) or systems describing the objects' behavior and their relation (e.g., MASCARET (Buche et al., 2003)), seem to be the most suitable solutions in this context. These systems can be further enriched by reusable interaction techniques (e.g., VirtualHand (Mine et al., 1997)). Another possibility is to describe the events and actions performed in the VE through scenarios. They can be predefined, whereas all possible scenarios are initially described (e.g., HCSM (Cremer et al., 1995)), or they can be emergent (e.g., VRaptor (Shawver, 1997)). In emergent scenarios, the actions are not sequenced; the users' behavior depends on the state of the environment (e.g., the pump is closed, the users need a wrench to open it), or the users have goals (e.g., assembly an engine) and they an individualized list of actions. Many of these scenarios are built in a specific language, which complicates their comprehension. To ease their understanding, a graphical representation can be used instead (Sugiyama et al., 1981). In this way, some solutions employ a graphical representation of Petri-Nets to model scenarios (e.g., #SEVEN (Claude et al., 2014)). Thus, the users' actions are either fully constrained, semi-constrained, or free and the scenario acts either as a listener or as a constraining system. The actions in scenarios are similar to the external activities depicted in the theoretical design models. Here, the actions described are the interactions between the user and the objects or the world. Consequently, a scenario is able to reflect the design activities of users, to model the iteration and to model the links between the activities.

3 SOLUTION OVERVIEW

The VR community needs solutions to build multi-function and personalized VR applications supporting design in VEs. The solution needs to support the three design concepts shared among the design domains, namely, the problem, solution, and evaluation. The solution needs to depict generic design activities to allow the developers to use this generic base in building personalized applications for designers. The application that aims to cover all the design activities needs to be versatile.

Consequently, our solution uses a new theoretical design model to define generic design activities (see Fig. 2). Then, the theoretical model is naturally described by scenarios in a VR Framework. Finally, the scenarios, which are implemented in a VE, lead to the production of VR applications for industrial design.

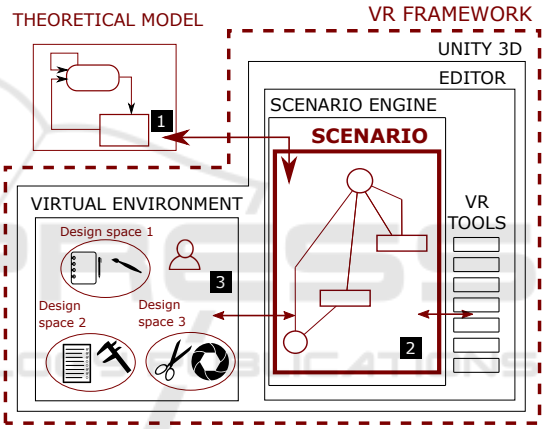

Figure 2: In red, our contribution: (1) a new theoretical design model defining generic design activities, (2) a scenario depicting the new theoretical model, and (3) an implementation of the scenario in a VE.

For example, the scenario is able to activate a 3D VR pen because a user is in a sketching area (see Fig. 3). Scenarios are the cornerstone of our solution. All the design activities are correlated to the use of a set of design tools. The scenarios are malleable and they can depict other models or spatial organizations of the VE.

4 A NEW THEORETICAL DESIGN MODEL

Our VR Framework uses a theoretical design model to assist VR developers in building VR applications that support the design activities, i.e., the users' actions in producing design-related content. Each activity produces an external representation, such as a

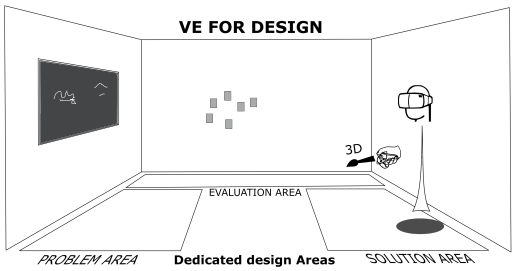

Figure 3: A VE depicting three areas dedicated for design activities (see Sec. 6). The design activities are linked to the use of VR tools (e.g., the 3D pen).

sketch. Since the domains have different design activities, process, and tools, we decided to build our solution around a generic theoretical design model. The aim is to enable the developers to use this model in creating a new, personalized version that fulfills designers' needs within their domain.

To the best of our knowledge, no generic theoretical design model has been defined yet for VR use. Our aim here is to fill the respective gap by illustrating three common concepts of the design process within our model. Since three concepts are shared in many design domains, our model is based on the following three spaces of design activities: problem, solution, and evaluation. Some of the existing design models depict the problem and the solution as spaces, leave out the evaluation, which is considered to be an activity. Moreover, this new three-partite theoretical design model is the result of collaboration with industrial partners (co-authors of this paper) and of the industrial design process analysis. We distinguish two types of design activities in each space: the creation of a new content and the iteration of the content. Moreover, in evaluation, the model distinguishes the act of evaluating and the act of defining/modifying criteria of evaluation.

The three-partite theoretical model is conceptualized as follows (see Fig. 4). First, the *problem space* includes two activities: the *formulation*, the first representation of a problem (new content), and the *reformulation*, an iteration of the problem representation (modification of the content). Second, the *solution space* includes two activities: the *conjecture*, the first representation of a solution (new content), and the *definition*, an iteration of the solution representation (modification of the content). Finally, the *evaluation space* includes three activities: the *definition of evaluation criteria*, the first representation of a set of criteria (new content), the *modification of evaluation criteria*, an iteration of the criteria (modification of the content), and the *evaluation*, the act of evaluating (new content).

A user switches from one activity to the next when

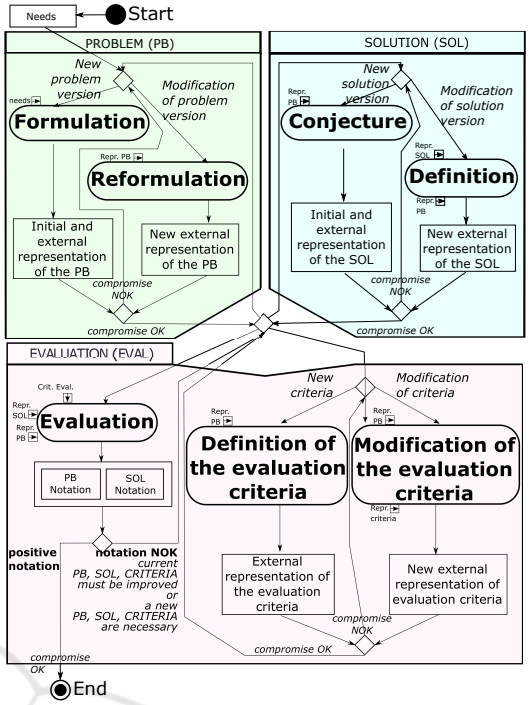

Figure 4: Our generic theoretical design model representation: three spaces (i.e., problem, solution, evaluation) with seven activities. The model is based on creating external representations.

a compromise is reached. For example, the first formulation of the problem takes place when the user explains the needs of the client. When the user considers the problem to be adequately described, a compromise is reached and the user continues the design process by defining evaluation criteria. Thus, our theoretical design model is sufficiently generic to cover a range of design domains. Each domain is able to use a version of the model and adapt it by providing rules for the evaluation, for the definition of a problem, or for deciding on how to reach a compromise. With defined generic theoretical design model, the next step is to depict the design model with a scenario model.

5 SCENARIO-BASED VR

Our VR Framework uses a Petri-Nets language based on the scenario model (Claude et al., 2014), which enables the graphical representation of scenarios. Moreover, Petri-Nets (Petri, 1962) are able to model user events, environment states, and object behavior. Another advantage of the Petri-Nets is that parallelism and concurrency can be described. In this way, several design processes can be depicted, and played simultaneously or independently.

The scenario model is based on a series of events and the event model uses a token, places, transitions, sensors, and effectors (see Fig. 5). The following explanations present the standard event functioning in order to understand the next sections. An event occurs when a transition is triggered, and when a token transitions from an upstream to a downstream place of the transition. First, a transition is ready to be triggered when the upstream place holds a token. Sometimes, events occur only if a condition is fulfilled. For example, the problem tools should only be activated when the user is in the problem area. In the event model, the sensor defines the condition. Then, the sensor of the transition examines the VE to check whether the condition is fulfilled. When the state of the VE meets the requirement, the transition is triggered and the effector interacts with the VE (e.g., to activate tools in the problem area). Finally, the token is consumed and it transitions to the downstream place. The initial and the final place define where the scenario starts and stops. The scenario ends when the token reaches its final place.

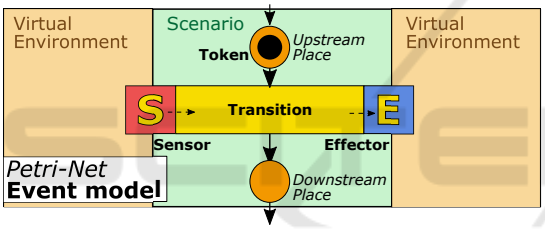

Figure 5: Petri-Net event model of the scenario.

As the scenario model is able to describe several activities and to interact with a VE, depicting our generic theoretical design model is a valid solution. In the following section, the Petri-Net language is used to depict the theoretical model.

6 DESIGN MODEL SCENARIO

The scenario is divided in three sub-scenarios (see Fig. 6) corresponding to the three spaces of the theoretical design model. The sub-scenario *Problem Activities* depicts the *Formulation* and the *Reformulation*. Then, the *Solution Activities* depict the *Conjecture* and the *Definition*. Finally, the *Evaluation Activities* depict the *Definition of Criteria*, the *Modification of Criteria*, and the *Evaluation*. Thus, each of the theoretical design activities is included in the scenario.

The main scenario determines which sub-scenario needs to be activated according to the user's location (see Fig. 7). For example, the main scenario examines the VE, detects whether the user is in the problem area, and activates only the problem sub-scenario. In the meantime, the two others sub-scenarios are in standby. Thus, only the problem activities can be triggered. Due to their distinctive labels, sensors are able to differentiate among different areas.

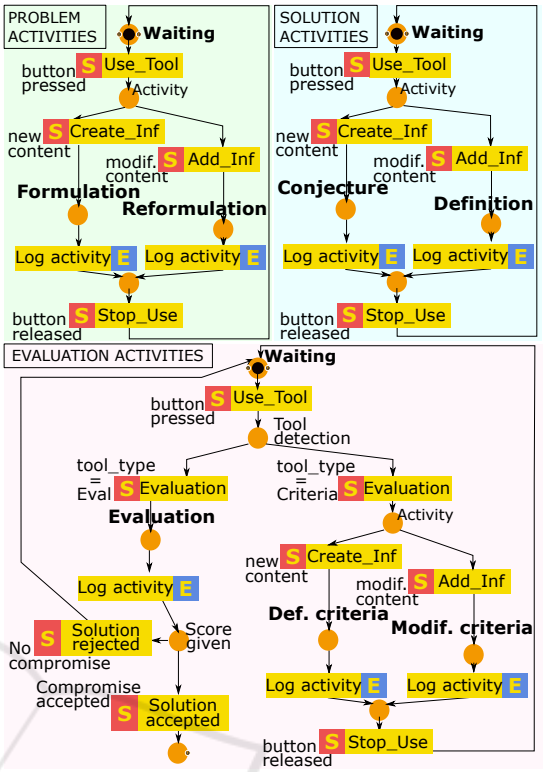

Figure 6: Three sub-scenarios depicting the theoretical design activities. The activities are split according to the three spaces of the theoretical design model: problem (green), solution (blue), evaluation (red).

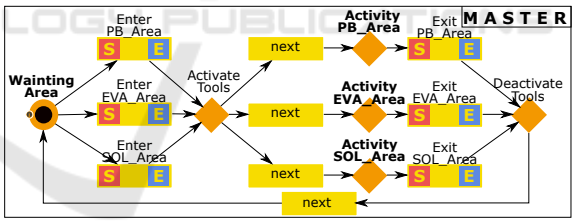

Figure 7: The main scenario with the three sub-scenarios included in the sub-nets (orange diamond) *Activity PB_Area*, *Activity EVA_Area*, and *Activity SOL_Area*.

According to our theoretical model (see Sec. 4), the design activities produce external representations. In the VE, the use of a design tool enables this production. Thus, the scenario detects an activity when a design tool is used. For the same category, the scenario also uses the labels to differentiate among tools. Additionally, the theoretical model differentiates between the activities that produce new content from those that simply modify content. In the sub-

scenarios, the sensors detect when the designer interacts with an empty or with an existing field. Thus, the sub-scenarios are able to distinguish *Formulation* from *Reformulation*.

The scenario model proposes a scenario engine and a graphical editor. Moreover, seeing that the scenario engine is already integrated in Unity3D, the link between the scenario, the editor, and the VE is already functional. This solution was used to build the use case presented in the following section.

7 USE CASE

We present the operation of our VR Framework through a use case, namely, the design of a working room layout. One user is immersed in a VE which is divided into three areas (see Fig. 8), and where dedicated design tools are situated. The following presents: (1) the designer's series of events in the VE, and (2) the functioning of the application, and the link between the VE and the scenarios.

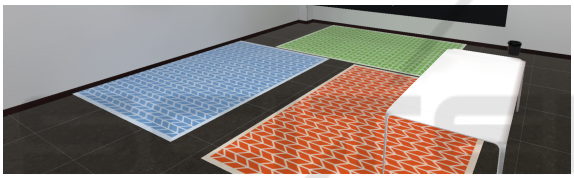

Figure 8: The VE organization: three theoretical spaces represented by three different areas. Left (blue): the problem area. Right (orange): the solution area. Background (green): the evaluation area.

7.1 Designer's Activities

This sub-section describes the designer's action sequencing. The designer's goal is to propose a layout considering the norms and the operators' needs. The proposal, based on the theoretical model and the scenario, is produced by iterations and evaluations.

First, the designer, who is immersed in the VE, heads for the problem area, and sketches out the needs on the white board: a working room includes several operators, others for support, and usable floor areas.

Then, the designer moves to the evaluation area to define the evaluation criteria based on problem formulation. The designer thus creates several sticky notes, one per criterion (e.g., usable floor area = $9m^2$/station in reality).

Next, the designer goes into the solution area to create the first layout by using a VR mock-up tool. They place small-scale desks, operators, and furniture on a table. The mock-up tool enables the designer to add or remove content of the mock-up. The mock-up can always be cleared and a virtual pen is available to sketch over the mock-up.

Once the solution seems at least to fill one criterion(e.g., the usable floor area criteria), the designer needs to evaluate the initial idea. Here, the designer needs to choose an evaluation tool in the evaluation area to annotate the solution in view of the given criterion. For example, the criterion is "$9m^2$/station" and the evaluation is "$7m^2$/station, need more space".

The designer has thus completed one design loop consisting of the problem formulation, the definition of the criteria, the solution conjecture, and a negative evaluation. The design process continues and the next sub-section details the functioning of the application.

7.2 User Scenarios and VE Relationships

The following describes the process of a designer working on the solution to extend the usable floor areas, and to satisfy the "$9m^2$/station" criterion.

Preparing the Working Area. During this step, the designer enters the *Solution Area* to modify the existing version of the working room layout with a specific design tool. For this, the user has to go into the *Solution Area*. The transition *Enter SOL_Area* (see Fig. 9) of the main scenario identifies, with its sensor, that the user has entered the respective area.

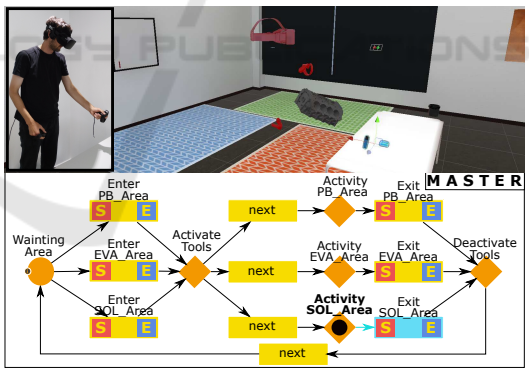

Figure 9: The scenario has detected the user's location: *Solution Area*. The situated design tools are activated and the sub-scenario including the solution activities can be triggered: the token is in the sub-scenario *Activity SOL_Area*. The blue transition is activated but not yet triggered.

The effector stores the user's location together with three lists of VR design tools in the token. In this way, the effector in the sub-scenario *Activate Tool* is able to activate the labelled design tools of the *Solution Area* (see Fig. 9). At this stage, the solution design tools can be used, and the sub-scenario including the solution activities can be triggered.

Depicting the Design Activities. In modifying the layout, the designer chooses the mock-up tool to add or move furniture. These activities are depicted in the sub-scenario *Solution Activities* (see Fig. 10). As the activity relies on the use of VR tools, the transition *Use_Tool* detects when a button is pressed. The sub-scenario is able to distinguish the *Conjecture* from the *Definition*. The transition *Add_Inf* thus detects whether a content is already in the solution space. The token is now in the *Definition* place, which indicates that the *Definition* activity is performed. The user is adding information or modifying the mock-up to propose a solution iteration.

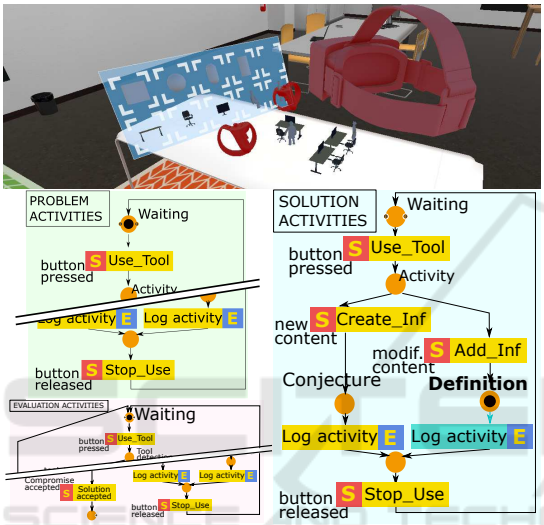

Figure 10: The sub-scenario *Solution Activities* is triggered. It has detected the use of a solution tool, and that the space is not empty. Thus, the token is in the Definition place, which shows that the user is modifying the current mock-up. The transition is blue as it is activated but not triggered.

The scenario is thus able to distinguish among activities performed by the designer depending on the area, the tool used, and the creation/modification of content. Next to that, the scenario is able to log the designer's activity (i.e., transition *Log activity*, Fig. 10) which enables versioning. The scenario enables iterations on the problem, the solution, or the evaluation until the evaluation fails.

New VR tools can be added as each domain is able to customize an implementation of the model by integrating specific tools and/or by specifying an activity (e.g., adding branches, transitions).

8 CONCLUSION

In this paper, we presented a new VR Framework created to support the development of VR design applications. Our VR Framewok implements a new, generic and theoretical design model through scenarios. It depicts the design activities as users' actions, which are built for implementation into VR. It uses three key concepts shared by many design domains: the problem, solution, and evaluation space. The theoretical model describes a design process with design iterations through the co-evolution of the three spaces. The VR Framework uses a Petri-Nets language and a scenario model in implementing our theoretical design model. The scenario is used to manage the design activities based on the theoretical design model in interacting with the VE. The VR Framework forms the basis on which developers can customize our generic theoretical model for specific design domains. The application is personalized through the implementation of specific VR design tools. The creation of these specific VR tools is not in the scope of the presen contribution. We illustrated a dedicated VR application for a specific domain, the design of a working room layout, based on our VR Framework.

According to the designers, they presently need to return to a previous design state after multiple iterations. Thus, a versioning solution could upgrade our VR Framework. Seeing that the scenario is able to track user activities, this is the first step towards developing the versioning functionality. Although we presently focused on the activity of a single user, some design circumstances involve several stakeholders, which would lead to creating collaborative process. The scenario engine is able to manage multiple users in VR as well as collaborative activities. Consequently, future studies should investigate the use of scenarios in developing versatile and personalized VR applications that support co-design.

REFERENCES

Arias, E., Eden, H., Fischer, G., Gorman, A., and Scharff, E. (2000). Transcending the individual human mind-creating shared understanding through collaborative design. *ACM Trans. Comput.-Hum. Interact.*, 7(1):84–113.

Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A., and Cruz-Neira, C. (2001). Vr juggler: a virtual platform for virtual reality application development. In *Proceedings IEEE Virtual Reality 2001*, pages 89–96.

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.

Buche, C., Querrec, R., De Loor, P., and Chevaillier, P. (2003). Mascaret: pedagogical multi-agents systems for virtual environment for training. In *Proceed-*

ings. 2003 International Conference on Cyberworlds, pages 423–430.

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.

Cremer, J., Kearney, J., and Papelis, Y. (1995). Hcsm: A framework for behavior and scenario control in virtual environments. *ACM Trans. Model. Comput. Simul.*, 5(3):242–267.

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.

Evbuomwan, N. F. O., Sivaloganathan, S., and Jebb, A. (1996). A survey of design philosophies, models, methods and systems. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 210(4):301–320.

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.

Jo, H. H., Parsaei, H. R., and Wong, J. P. (1991). Concurrent engineering: The manufacturing philosophy for the 90's. *Computers & Industrial Engineering*, 21(1):35 – 39.

Kessler, G. D., Bowman, D. A., and Hodges, L. F. (2000). The simple virtual environment library: An extensible framework for building ve applications. *Presence*, 9(2):187–208.

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

Maher, M. L., Poon, J., and Boulanger, S. (1996). *Formalising Design Exploration as Co-Evolution*, pages 3–30. Springer US, Boston, MA.

Mine, M. R., Brooks, Jr., F. P., and Sequin, C. H. (1997). Moving objects in space: Exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 19–26, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

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.

Moran, T. P. and Carroll, J. M. (1996). *Design rationale: Concepts, techniques, and use*. L. Erlbaum Associates Inc.

Pahl, G., Beitz, W., Feldhusen, J., and Grote, K.-H. (2007). *Product Development Process*, pages 125–143. Springer London, London.

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

Pugh, S. (1986). Design activity models: worldwide emergence and convergence. *Design Studies*, 7(3):167 – 173. Special Issue: the design coalition team.

Satter, K. and Butler, A. (2015). Competitive usability analysis of immersive virtual environments in engineering design review. *Journal of Computing and Information Science in Engineering*, 15(3):031001.

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.

Shawver, D. M. (1997). Virtual actors and avatars in a flexible user-determined-scenario environment. In *Proceedings of IEEE 1997 Annual International Symposium on Virtual Reality*, pages 170–177.

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

Sugiyama, K., Tagawa, S., and Toda, M. (1981). Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125.

Visser, W. (2006). Designing as construction of representations: A dynamic viewpoint in cognitive design research. *Human-Computer Interaction*, 21(1):103–152.

Zawadzki, P., Gorski, F., Bun, P., Wichniarek, R., and Szalanska, K. (2018). Virtual reality and cad systems integration for quick product variant design. In Hamrol, A., Ciszak, O., Legutko, S., and Jurczyk, M., editors, *Advances in Manufacturing*, pages 599–608, Cham. Springer International Publishing.

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.