Virtual Reality manuscript No.
(will be inserted by the editor)

The Design Space of Dynamic Interactive Virtual Environments

Kristopher J. Blom · Steffi Beckhaus

Received: date / Accepted: date

Abstract Keywords Virtual Environments · Dynamic Interactive VEs · 3D User Interaction · VR systems

Virtual Environments have become a key component of many fields and the critical component of Virtual Reality applications. Due to their virtual nature they can accommodate an infinite number of possibilities. A theoretical work is presented, which decomposes those innumerous possibilities into concepts to help clarify the vast design space and provide insights for future applied research. We propose that what makes environments *interesting* and *engaging* is having worlds that are both active and reactive. This article explores the manifestations of those actions and reactions in what we term: dynamic components and interactions. We term worlds containing these Dynamic Interactive Virtual Environments (DIVE). An analysis of each component time was performed, with the purpose of providing a theoretical understanding of the respective design spaces. Initially we collected the myriad possibilities of each component, e.g. the possible kinds of interactions. We point to examples throughout the field to ground and explain concepts presented. We then categorized of each area into taxonomies. The result of the analyses provide insights into the design space of Virtual Environments, expose several avenues of research that are yet under-explored, and provide better understandings of ways in which DIVE creation can be supported.

K.J. Blom - S. Beckhaus
interactive media.virtual environments
Department Informatik
Universität Hamburg
Vogt-Kölln-Strasse 30
D-22527 Hamburg
Germany
E-mail: blomk@acm.org
E-mail: sb@steffi.beckhaus.de

1 Introduction

Statements like "in Virtual Reality anything is possible" and "the possibilities are only constrained by the imagination of the designer," have been heard for decades. In many senses, these statements are true, as it is all just virtual. However, this promise of everything and anything in Virtual Reality (VR) has never really come to fruition. The realizable possibilities are much more constrained than the imagination of almost any of us. Part of this is that we do not yet have an understanding of what the virtual environment, the center point of VR, is and can be. Tools have been developed to solve technical problems; yet, rarely, has there been consideration of what is needed to create the virtual environments that fulfill the endless possibilities of VR. The tools for creating "anything is possible" simply do not yet exist. We feel the largest hindrance is currently that we do not have an understanding of what the possibilities are, and, therefore, do not have appropriate system support to enable its creation.

The work presented here strives to advance our understanding, through an investigation of the design space of the possibilities of virtual environments. We focus in this paper on those virtual environments (VEs) that are *engaging* and *experiential*. Naturally, the creativity of the designer is an important factor, and what makes an environment interesting or engaging is difficult to precisely define and somewhat personal. However, as with other media, like film or books, it is possible to identify components that are likely to assist in creating engaging or meaningful experience. In this work, we focus on the aspects that can be supported from a computer science standpoint.

In this article, we put forth a series of taxonomies designed to make those interesting and engaging components explicitly visible and tangible. These taxonomies were created through a process of collecting potential components and categorization. By identifying the basic components that are able to create such engaging, experiential environments, we can gain valuable insight into what is possible. The taxonomies serve as a reference for getting ideas of what can be done to make environments more interesting. They also outline the design space, providing a new, more formal basis and understanding of what VEs can be. Moreover, this work exposes the time based nature of many of these different components; based on that knowledge new approaches to their application and implementation may be fruitful to explore. Components found near each other in the taxonomies are likely to share many design and implementation details. This can assist with finding methods for development as well as insight for future design patterns.

The following section will propose that the two basic components necessary for such environments are **dynamic components**, changes happening over time, and **interaction**, in general and with dynamic components. Both of these function to liven the environment and to make it more interesting, particularly over longer periods of time. While individually they are interesting, environments that combine interaction and dynamic components in differing ways are likely to be more interesting. After introducing the style of environment we refer to as *Dynamic Interactive Virtual Environments* (DIVEs), we explore the design spaces of the main identified components individually in Sections 3 to 7. A short discussion of work on DIVEs follows in Section 8.

2 Engaging Environments

Our goal in this section is to come to a basic understanding of the components that make up environments characterized as: engaging, exciting, affective, interesting, experiential. We take inspiration from an arbitrary selection of a few simple, yet illustrative, examples of well known environments that achieve such experiences. Considering those examples, we identify and define what we believe are the critical components that require system support to enable designers to create engaging experiences.

2.1 Example Environments

Virtual Reality Distraction Therapies are an example of the class of engaging environments. The goal of these environments is to capture the attention of the user so completely that they are able to remain detached from the physical realities surrounding them and even partially ignore painful procedures. The "Snow World" application is one example of this [32, 50]. Snow World has been used in various applications, but originally was for burn treatment. The environment is just a simple game, presented immersively, where the user shoots snowballs at snowmen, while moving through a snowy environment. The original version is of quite low fidelity, though newer versions have more up to date graphics. What is interesting in Snow World is that such a simple game, with limited interaction possibilities, can reduce the perception of pain by successfully engaging the user.

Another highly successful application has been a Virtual Reality Exposure Therapy (VRET) design for treatment of Post-Tramatic Stress Disorder (PTSD) [45]. In therapy sessions, the patient is exposed to scenarios from the traumatizing events. Early applications were for therapy of Vietnam veterans [31]. For this therapy to work, the patient has to feel that they are in the situation(s) that caused the trauma. In early works, a Vietnam scenario was used. Of particular importance were the actions in the environment, e.g. helicopters. VRETs are now used for war veterans and active duty personnel [37] as well as being used in setting such as the September 11 attacks [18]. VRET applications are different than distraction therapy, because at its core it is about transporting the patient to another place and having them experience some particular time and events. It is necessary that the scenario "comes to life." Although the scenarios are filled with action, interaction is typical minimal. They are also multi-sensory, moving beyond visual and audio to include haptic and olfactory (smell) cues, both as transient and static parts of the environments.

Finally, one of the classic virtual environments,"Crayoland," is a worthwhile example [41]. Crayoland is very polygonal, with textures that were made with crayons. The grass plane of the scene is contained by cutout mountains. A small shack with windows and a door is present as well as a tree and a pond. On the tree is a bee hive that, when swatted, releases bees that swarm around the user for a while. In a field of flowers is a butterfly, which patient users can get to land on their virtual hand. When Crayoland becomes reality is while batting of the beehive and the ensuing pursuit of angry bees – are they really angry? In our experience, the visitor to Crayoland tends to say they were angry, anthropomorphizing them similarly to what has been shown to happen with virtual characters [21]. Another case is getting the butterfly to land on your hand; it actually lands on a virtual hand representation that is in front of the control device; However, people recall the event as if it landed on their hand. Without those parts of the world that are active and reactive to the user, Crayoland would likely be long forgotten. Crayoland only functions because it reacts to the user and is alive itself.

2.2 Deriving Components

The examples above are only singular examples of the many possible environments. However, they demonstrate that environments can produce powerful experiences, even when extremely simple. In the distraction therapy example, a clever context and a simple set of interactions is enough to engage the user to an extent that it can reduce pain perception. The VRET example shows that extensive interaction is not necessary; instead, it creates a meaningful and intense experience through action in the environment. Finally, Crayoland demonstrates the power of the combination of interactions and action and provides important examples of those combinations. Certainly in these examples, content that interests the users and that affects the user's emotional state is important. Content designers and storytellers can create these human connections, but require a well defined framework of their design space in which they should work and mechanisms to support content creation within that design space. We believe two critical components that are fruitful to support from the CS perspective are reactivity, i.e. interaction, and activity, i.e. changes over time.

In the literature, we find support for interaction's importance. Interaction is usually included as part of the definition of virtual reality and is a definite requirement for games. Central to attempts to understand the experience of immersive virtual environments has been presence [48]. In presence research, interaction was consistently seen as a factor [28, 51, 56]. Other recent work has used interaction as a foundational component of experience [17, 22, 57, 61]. The distraction therapy example demonstrates this markedly, as the interaction helps place the user in a different environment than the therapy room.

There is some support in the literature for activity being an important component of experience in VEs. It has even been argued that action is what defines the reality of the environment [65]. Recent theories put forth by Slater et al. have held that the plausibility of the actions in an environment [53] and the "behavior-response" of the environment on user interactions are necessary components of presence [54]. Both of these imply further that the environment has to be believable. We contest that part of believability is that the world is not static. Slater's behavior-response concept requires this, and his examples are based on this. However, Slater's arguments seem to implicitly assume avatars as the actors and interactors, where we see this more broadly. These non-static environments are instead dynamic, such as the activities in the virtual Vietnam VRET or the bees and butterfly in Crayoland. These dynamic and interactive components often lead to more believable environments.

Our investigation of the design space followed a multi-step process. The first step was to formally define the components of interest outlined above. The second step was to identify manifestations of each different class of elements. This was done by posing questions like: What interactions can you imagine being possible in a virtual environment? We produced these lists over a number of weeks, holding small workshops with various participants with backgrounds ranging from VR to computer gaming. An immense number of examples were produced. To reduce that data into a manageable quantity we categorized the data into taxonomies. The leaves of the taxonomies are examples and the branches capture the common natures of those examples.

2.3 Defining the Components

Two key components of the type of environments we are interested in were identified in the previous discussion. In this section we formally define the components and further derive three additional components.

We define interaction as:

> **Interaction** is the user taking influence on the environment or its controlling structure.

This definition takes a different approach to interaction than has previously used in the VR community. The VR view of interaction has largely identified with Bowman's interaction taxonomy [11] and is still the prevalent view, cf. [35]. Bowman did not define interaction, but his focus was clearly on interaction techniques and was closely tied to implementation. This viewpoint holds that four interaction types (tasks) exist: selection, manipulation, navigation, and system. The community has largely focused on these interactions and techniques, devoting limited effort on other possible interactions.

The second component identified was action. These actions can be better understood as aspects that change over time. We refer to the portions of the VE that change over time as "dynamic components,"[1] as they create the opposite of a static environment, and define them as:

[1] In prior publications we had referred to these as 'dynamics.' A discussion of the choice of nomenclature can be found in Online Resource 1.

dynamic component is anything that changes over time that effects perceivable changes to the VE either directly or indirectly.

Our definition is purposefully general, covering an immense spectrum. We feel anything that changes over time potentially makes the environment more lively and interesting. However, we constrain the meaning slightly to anything that has a *perceivable* effect on the VE. Anything that is never perceived does not really make the environment more interesting. Perception in our understanding includes unconscious perception, allowing ideas like masking, but excluding phenomena like "change blindness." This perception clause does not, however, imply that an immediate or direct visibility of the changes is required. A *dynamic component* can also be an indirect and even invisible change to the actual user of the system, for instance a branch in a story due to the user 'choosing door number two.' This visibility issue is also a motivation for using the term *dynamic component*, as it does not imply a visible object change.

We are interested in VEs that contain both *dynamic components* and *interaction*, such as those in the Crayoland example. An important question that needs to be addressed is what happens when components are both interactive and dynamic? We feel this question is crucial to the development of better support of new types of interesting environments. Unfortunately, the areas at the conjunction of dynamic components and interaction are under-developed.

There are two ways in which such actions and reactions can be combined. Examples of these combinations are found in the Crayoland example. The butterfly is a *dynamic component*, and furthermore one with which the user can interact, when getting it to land on your hand. We refer the resultant combination as an *interactive dynamic component.* When the butterfly is on the hand, a new dynamic component is introduced through the interaction, i.e. slow movement of the butterfly induced by the user's hand. These we refer to this as a *dynamic interaction.*

We define dynamic interactions as:

Dynamic interaction is any interaction that either induces a dynamic in the environment or the interaction takes place over a period of time.

Dynamic Interactions are quite commonplace in classical VR. Most of Bowman's manipulation category of interaction techniques falls into this category. Manipulations cause an object to move, rotate, or scale over time. Although one does not often think of those manipulations as causing a dynamic, without seeing the physical manipulation — for instance in a distributed VE without an avatar — the changes would just be a dynamic component. An even more commonplace *dynamic interaction* is travel. *Dynamic interactions* exist beyond those that are part of Bowman's taxonomy and will be discussed in Section 5.

The other combination of dynamic components and interaction is a class of interaction involving dynamic components:

Interactive dynamic components are any interaction, where the "object" of interaction is a dynamic component.

This subset is those interactions that are with dynamic components that are changing at the moment of interaction. For instance, if we are to strike a virtual football that is traveling towards us, the ball would be an interactive dynamic component. Even the selection of the object would be an *interactive dynamic component.* In our definition, object is offset in quotes to indicate that the 'object' of an interaction may not be a traditional, concrete thing that is "manipulated." Having a conversation with an embodied agent would be a case of an *interactive dynamic component.* The 'object' might not even be visible; for instance, interactions with the storyline, via the user's actions, would also be an *interactive dynamic component.*

There remains one additional step along this path that we can take. When investigating interactive dynamic components, it becomes apparent that their combination with dynamic interactions is inevitable. These are dynamic interactions, where the target of the interaction is a dynamic component. We refer to these as *dynamic interactions with dynamic components.*

> **Dynamic interaction with a dynamic component** is a special subset of interactive dynamic components that is concerned with cases when the interaction performed is itself occurs over time. Hence, the object of interaction is changing over time as well as the interaction with that object.

Five types of components we believe contribute to creating interesting, engaging types of environments have been identified: *dynamic components, interaction, interactive dynamic components, dynamic interactions,* and *dynamic interactions with dynamic components.* In the coming sections, we investigate each of these component type in turn. For each, we develop catagorizations. The *interaction* design space is presented in the next section, followed by the *dynamic components* design space. Sections 5 and 6 explore the design spaces of *dynamic interactions* and *interactive dynamic components* respectively. Finally, the special case of *dynamic interaction with dynamic components* will be discussed, before moving onto a general discussion in Section 8.

3 Interaction

Interactions are a fairly obvious component for creating environments that can be interesting for longer periods of time. Although there are infinitely many conceivable interactions, they reduce nicely into a relatively small hierarchy of interaction types. The developed taxonomy of interactions can be found in Figure 1. Prominent among them are the classical VE interactions of object manipulation; however, there are other interactions not often considered in immersive VEs. These actions are somewhat more common in games, though still in limited capacities. Many of these interactions are complex. For instance, conversations with a virtual character are definitely interactions, but are hard to implement unless trivialized. An example is the classic "click on the response" method for avatar interactions seen in many computer games.

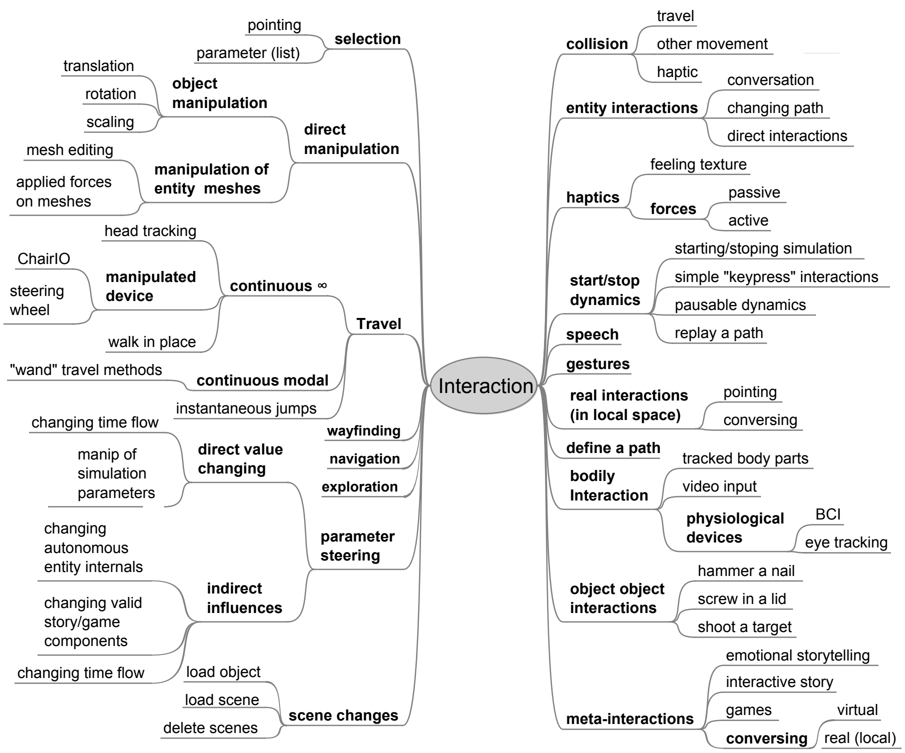

Fig. 1 Design space of interactions with a VE. Bold font indicates a category or sub-category, and standard font listings are concrete interaction examples.

Portions of the categorization merit additional explanation. Speech and gesture interactions are not further discussed here, as they are classes of interactions of their own right [27,62]. Interactions that occur in real space, though involving the virtual space, form a perceptually different class. For instance, two people pointing to virtual objects and talking about them. Though not purely VR interaction, they are important and have been researched by a number of groups, cf. [5,47]. When considering the possibility of distributed virtual environments, these person to person interactions take on a slightly different meaning, but are essentially the same.

An emerging type of interaction is non-intentionally controlled interactions. Bodily interactions of emerging types, like those using physiological signals, are an example of this type of interaction. For instance, various researchers are working on emotional coupled steering of games and VEs [7,26,64]. It is important to note, that many of the same signals may also be used for intentional steering. For example, the emerging area of passive BCI [66] uses brain activity to determine cognitive or emotional state, and intentional BCI, such as motor imagery [23], has been shown capable of controlling a VE.

The final two interaction categories on the right hand side have been only narrowly considered to date in the VR community. Meta-level interactions provide deeper meaning to the environment; this is vital for creating experiential environments. Creating such interactions is challenging, from conception to design and implementation.

The area of object-object interactions has had some attention in VE related communities, but has been limited in scope. The simplest example is shooting a gun, where the bullet then interacts with objects in the scene. This is common in many games, but also in immersive VR in the military, e.g. [12, 20]. The area of training applications, where interactions are based on some tool, has done some work in this direction. Conceivably any (virtual) tool could act upon another. Examples in VR include surgical simulators [38, 43], welding trainers [63], and automotive painting simulators [34]. Extensive research has been done in the computer graphics community on implementing complex object-object interactions, such as collisions [58] and soft-body deformations [40]. Object-object interactions might not even be initiated by user, but interactions between virtual objects independent of the user. For instance, virtual characters may interact with the simulated environment [33].

4 Dynamic Components

When considering the possible dynamic components that could be found in a VE, the number is overwhelming. As with interaction, when we collected ideas we strove to: not constrain ourselves by prior ideas, not get lost in the myriad of endless variations (e.g. by different objects being put together), and not to miss things by generalizing prematurely. The most important criteria we used when generalizing and grouping was that of observer perception. This is in contrast to a typical view of how they are implemented. We felt this was important, as thinking about the implementation at this stage tended to constrain one's view greatly. After collecting the potential dynamic components, we categorized them according to observer perception. A complete look at all levels of the categorization is not possible in this article. Instead, we present the highest levels and most important discoveries; a more complete view of the whole classification can be found in [6].

The dynamic components design space consists of six top level categories. Moving from the most overarching to specific, then to the most abstract, the categories are:

Overall Presentation artifacts of the presentation method

Scene Attributes artifacts of the scene presented as a whole

Singular Objects changes affecting a single item in the scene

Propagating Quantities time change effects that are not directly associated with the medium (object) in which they are present

System State artifacts of changes in the technology, either hardware or software, that is used in presenting the VE

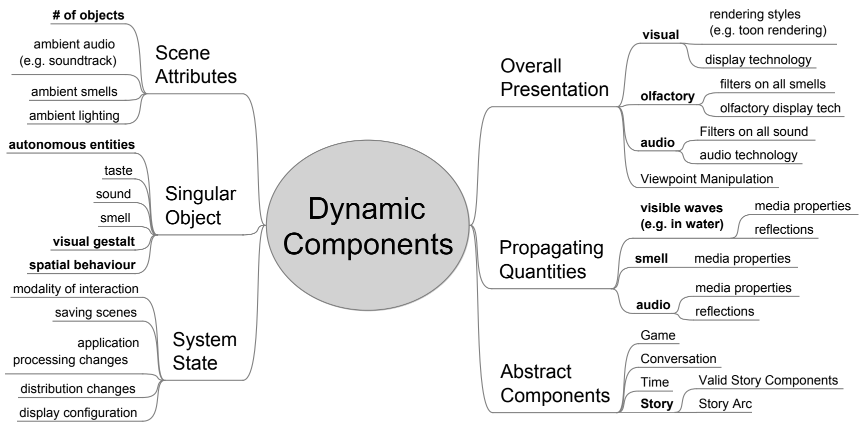

Fig. 2 A mindmap categorization of the highest level of the dynamic components taxonomy is shown. Bold font indicates a category or sub-category, and standard font listings are concrete interaction examples. The "Screen Attributes" and "Singular Object" hierarchies contain a lengthy sub-tree broken out in [6].

Abstract Components intangible and conceptual changes that can only be indirectly experienced

In the following text, each category is presented separately and more clearly delineated. Important sub-categories are mentioned. A mindmap of the highest levels of the taxonomy built on the perception direction is shown in Figure 2. The full taxonomy in [6] develops the areas of *Scene Attributes* and *Singular Objects* to deeper levels.

The *Overall Presentation* category deals with artifacts that are attributable to the presentation method. These components affect the entire VE, but are differentiated from the Scene Attributes category in that they are not strictly part of the scene. Examples include filters that change how the scene is presented. For instance, visual effects, e.g. cartoon shading, could be added to change the appearance completely [25]. Similarly, something as simple as the audio level fits in this category. A physical example of this is the warming of the projectors over time. This changes the color of the complete system.

Scene Attributes are dynamic components present at the level of the whole world. These dynamic components are part of the scene, but are not just singular objects within the scene. Ambient effects across the different modalities fall into this classification. The addition of ambient sounds, such as a "soundtrack," to make the VE more interesting is one example. A common dynamic component implemented in VR classes is changes to the number of objects in the environment. This can either be the introduction of new objects or the deletion of objects. We categorize them as scene attributes changes, because it is not a change of the object itself, but rather a change of the scene's composure. Similarly, the breaking of an object into multiple objects, a conceptually simple, physically based phenomenon fits into this category. The final dynamic component category is changes of the scene in its entirety. This typically reduces to a sub-worlds concept, as it is at some level still the same VE or at least the same program. The most common manifestation of this is the level paradigm common to computer games. Similarly, interactive storytelling environments may have different scenes as the story progresses.

The largest subset is dynamic *Singular Objects* and likely the most commonly thought of dynamic component. The two classical dynamic components of virtual worlds are in the sub-categories of *Spatial Behavior* and *Autonomous Entities*. The *Spatial Behavior* of an object is its movement through space. For instance, the classical manipulation of VEs is a spatial behavior [11]. *Autonomous entities* includes the classical virtual humans, which for many people is the quintessential dynamic component.

On account of their unique nature, *Propagating Quantities* are in their own category. These propagating quantities are manifest only through another medium. Examples include waves in an ocean, waves in a pool of water, sound waves, and smells. We chose to categorize these differently, because there is a perceptual difference for the observer. One doesn't think of the ocean moving, but instead of waves. One doesn't think of particles moving in the air, but instead of smells moving through the world. Here, the motion has its own importance that is largely independent of the transport medium, making the distinction important.

A number of potential changes are not so much a part of the VE, but rather of the *system state*. However, these impact the experience of the environment. Only one of these is a typical dynamic component of existing applications, modal interaction. Interaction with VEs is often performed with only a single device, which is overloaded in functionality. The changing between modes is an important dynamic component of the system. Related to this is the possibility of changing the display configuration or system configuration at run-time [4]. For instance, in certain Scientific Visualization contexts this may mean changing the number of processors or even computers involved in the computation [59, 29]. Similarly, in distributed applications this can be changes the connections or changes to the aspects of the VE that are distributed [52]. However, recall for these to fit our definition of a dynamic component, they have to have perceivable consequences, e.g. the speed of the simulation increases or new virtual characters appear.

Abstract Components of dynamic change are less tangible, but aspects that are implicit and potentially powerful components to making a VE into an experience. Story and game components of VEs are two of the major components at this level. Both of these necessarily unfold over time. In early games this was created primarily through levels; However, today's attempts try to handle this solely through a changing environment, without the explicit changes. Interactive storytelling environments are also implicitly dynamic VEs, particularly in that the stories are influenced by the user.

The final major potential abstract dynamic component of VE is time itself. As dynamic components are defined as being aspects that change over time, this may seem intrinsically wrong. However, in a synthetic world, time is not limited to moving in one direction or at a specific rate. More formally expressed, time does not have to be a homogeneous time arrow. Time in VR is flexible. This is a possibility that is often recognized and is often considered desirable [13,14,42,46]. An example use of this is for learning environments. In such cases the learner can experience the proper process at different speeds and review them freely [3]. Conceivable and even often suggested methods include: reversing time's flow, freezing time, changing time's "speed," heterogeneous time skewing, and an "undo" for time. However, these possibilities are rarely realized and, even then, usually in very limited ways.

The discussion of dynamic components up to now has been on classifying the potentials of the design space in terms of perceptual groupings. This is useful when thinking of the possibilities. We believe this will become a useful tool when considering how to make more interesting worlds, by providing a structured way to investigate possible dynamic components to add. It is also useful when evaluating systems for VE creation. Systems can be checked for support of each different class of dynamic component. Implementations within hierarchical categories can be expected to contain similarities or even exact methods. However, for implementation insights there is another way to classify the collected potentials, based on the decisive factor, time.

Classifying the dynamic component design space by how time is perceptually understood provides interesting insights into how to create the individual effects and how to support them. We have created a time based classification of dynamic components. The resulting taxonomy is shown in Figure 3. It is important to note that three of the four categories involve continuous or piecewise continuous time changes. This insight will be taken back up in Section 8.1.

5 Dynamic Interactions

Having looked at the design spaces of dynamic components and interactions, we can now move to the areas that combine them. The first we investigate is *dynamic interactions*. Recall that they are dynamic components induced through interactions. The design space of dynamic interactions provides interesting insight not only into what is possible, but uncovers a large research field that is largely unexplored and provides new insight into the necessary support mechanisms for such interactions.

The design space of *dynamic interaction* consists of various standard interactions and less conventional interactions. We present the design space in Figure 4 only in the form of a time based categorization, as the equivalent categorization based on perception contributes little additional information.

Considering first the different possibilities themselves, we see that a good number of the possible dynamic components are induced by interactions. This is interesting, as it supports the prevalent thought that interaction is of importance for making things interesting, as they also produce actions. We'll introduce the different dynamic interactions in three groups: classical VR in-

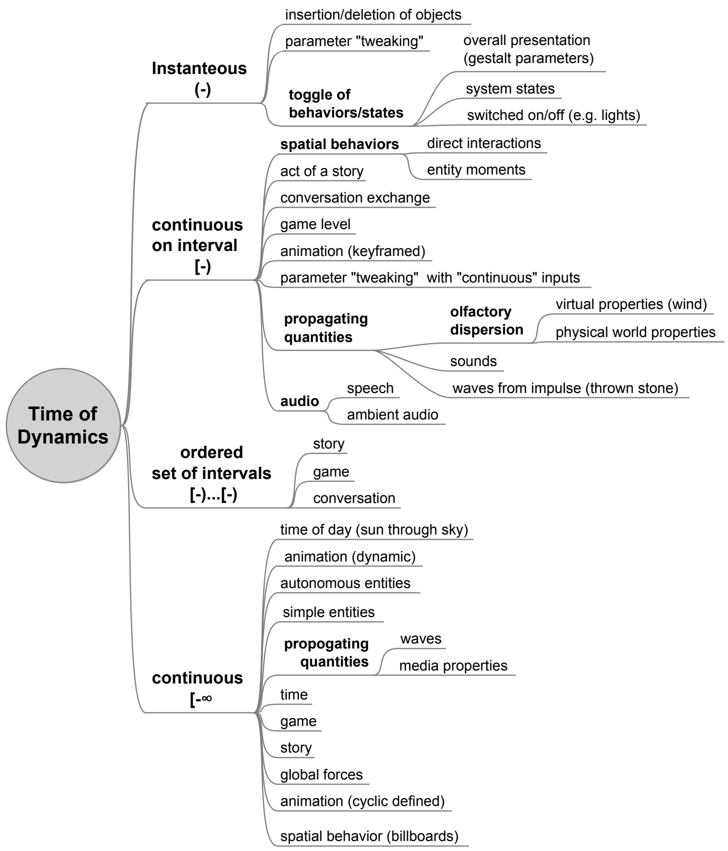

Fig. 3 A categorization of dynamic components based on a time representation criteria is shown. The boundedness of the representation is given below the descriptor. Bold font indicates a category or sub-category, and standard font listings are concrete interaction examples.

teractions, meta-level interactions, and those from “advanced” or “immersive” interfaces.

When we consider the interaction techniques that are typically used in VR environments, it turns out that most of them fall into the category of dynamic interactions. The ubiquitous interactions of object manipulation are a prime example as well as most of the travel methods that are crucial to VR. These classic VR interaction types are rarely considered with regard to their dynamic nature, exceptions include [30, 49, 55].

Meta-level interactions include a number of things like games and storytelling. However, other potential dynamic interactions can be included here. Conversations, gestures, and speech all can be include. Each is dependent on something occurring over time, which is part of what makes them challenging.

Finally, certain devices enforce a dynamic interaction. Basically, these are “continuous” sensors. Standard spatial tracking devices are this way, if values

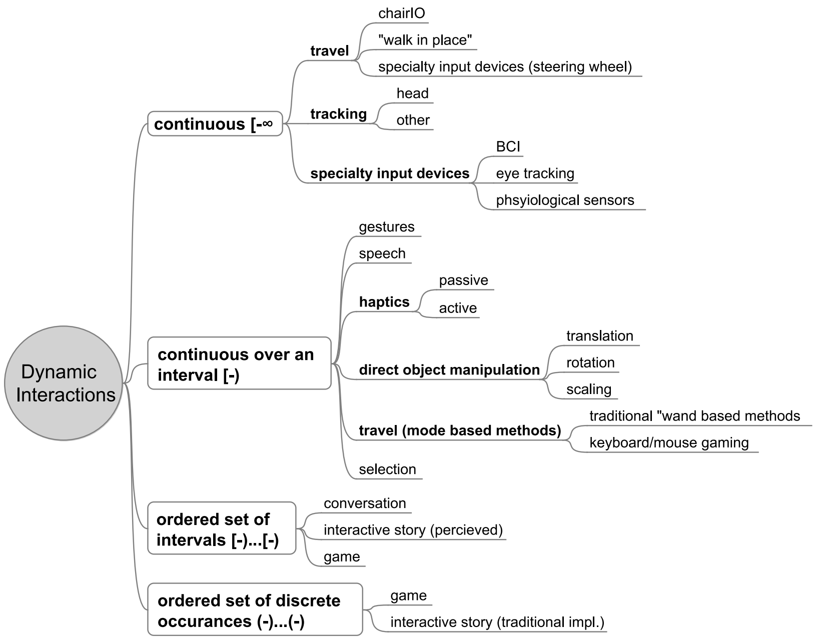

Fig. 4 The design space of dynamic interactions is shown in taxonomy form. Bold font indicates a category or sub-category, and standard font listings are concrete examples.

are continuously used. Haptics devices are also continuous in nature, because of the fidelity required. However, newer devices like Brain Computer Interfaces (BCI), eye tracking, and physiological sensors are potentially of this type dynamic interaction, even if they are rarely used as such.

When categorizing based on time representation, the dynamic interaction design space is divided into four categories. Two categorizes are continuous in nature, differing only by their boundedness in time, *continuous infinite* and *continuous over an interval*. The second set of interactions involves ordered sets of interactions that occur over time. We further differentiate between ordered sets that are composed of either continuous over intervals segments or discrete events.

The boundedness difference for the continuous time categories may seem trivial and irrelevant. However, this may prove to have an impact on the implementation, making it more than a conceptual difference. In the continuous infinite case, we expect the interaction to be active over the entire course of the application's life. A classic example of this is head tracking. Another example is the ChairIO travel interface [2], which is continuous for the length of program. On the other hand, tracking of an input device used for travel or manipulation typically is handled as a continuous interval, as they are almost exclusively performed as modal interactions.

The second set of categories is based on the premise that the interaction is composed of an ordered set of interactions that are distributed in time. The in-

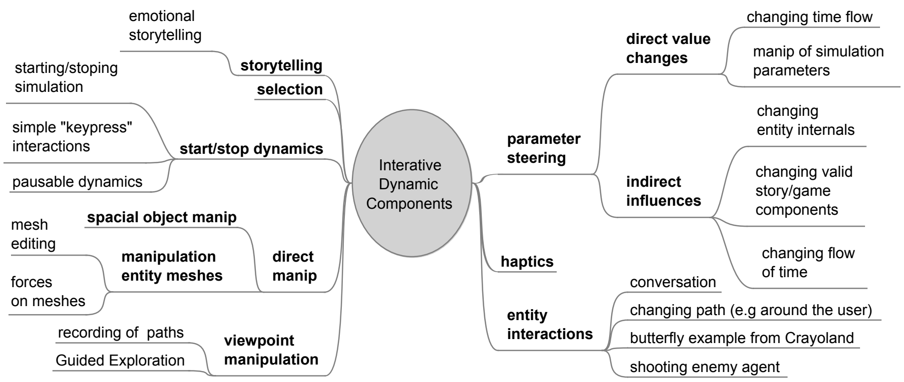

Fig. 5 The taxonomy of the interactive dynamic components' design space categorization. Bold font indicates a category or sub-category, and standard font listings are concrete examples.

dividual interactions take meaning through their association together. In this sense these are interactions that occur over time. For example, a conversation is a collection of speech components that occur in a specific order. Games and stories can be found as both sets of intervals and of discrete events. Although this deals primarily with implementation details, there is a conceptual impact on how the interaction is viewed. Classically, implementations view the interactions as a collection of discrete occurrences. This is partially because interactions in storytelling and games have been seen implemented as discrete interactions, but mostly because the implementation is based on a state machine approach. While the story of the experience is retold in this discrete way, the experiences are almost always continuous over some period of time, i.e. a continuous time interval.

6 Interactive Dynamic Components

The alternative combination of interaction and a dynamic component is an *interactive dynamic componen*. They are simply interactions that occur with a dynamic component. Conceptually this is quite straight forward; however, they are not as well understood and explored as prior ideas. Known examples of interactive dynamic components are yet limited in scope; however, we feel the larger area of interactive dynamic components will become important in the near future.

The design space of interactive dynamic components can be seen in Figure 5. The design space is seemingly small in size, though many of the categories encapsulate countless potentials. This small number may be more a result of the little exploration that has been performed in this direction than truly an indication of a small design space.

Examples of interactive dynamic components can be found in real life. Interactions with dynamic objects are performed in real life, though few cases are frequent for the average person. Many sports involve such interactions at their core, for instance kicking a moving ball. This often forms much of the challenge of the sport, particularly for youth and amateurs. While not everyone actively plays sports, another category is very commonplace; The “simple” act of having a conversation is built upon interactions that fall into this category.

One class of interactive dynamic components is simply taking dynamic components and interacting with them. While it is easy to say we will interact with a dynamic component, actually doing it or even considering and planning to do it more difficult. Computer games do this often as a way to increase the challenge. Because of implementation challanges, interaction with dynamic objects are typically instantaneous. However, as we saw in dynamic interactions, there are a whole class of interactions that are continuous and, potentially, more interesting than just “instantaneous” shooting and stomping interactions that are classic in computer games. These overlapping areas are what we have termed *dynamic interactions with dynamic components* and will be introduced in the next section.

A few of the categories shown in Figure 5 merit explanation. Parameter steering is a category, where the interactions are not directly with the dynamic component in question, but rather manipulation of some controlling mechanism in the simulation. Direct value changes are similar to what Bowman called “system” interactions. An example of this would be controlling the passage of time, e.g. stop, play, slow play, fast forward, and reverse buttons. Another could be controlling gravity's defined value. Indirect parameter steering is likely to be found in interactive storytelling. Here, the user's actions cause parameters in the story manager to be modified, e.g. the user ignores a character in the Interactive Drama Façade [36]. This in turn modifies the next act or task that the story manager picks out, which in turn modifies the behavior of the actors, e.g. the ignored character accosts the user. Such effects could be more subtle and long term, as seen in various current role playing games.

Most other interactions highlighted happen more at a meta-level. The object of interaction is not so concrete here. For instance, most meaningful interactions with an “entity” (avatar, person, animal/thing) involve some abstraction. A conversation implies that there is a change happening in the internal state of the conversant. Similarly, interactions in an interactive storytelling application imply that the interactions of the user influence in some way the flow of the story. Although, this may not actually be the case in the programming, perceptually successful applications provide the illusion of such deeper interactions.

7 Dynamic Interactions with Dynamic Components

Dynamic interactions with dynamic components are a challenging area first formalized here. They are interesting both for the application user and as a research area. It turns out that a large portion of the identified interactive dynamic components in the previous section fit into this category. For the user of an application, the meta-level dynamic interactions with dynamic components (e.g. conversation, interactive-storytelling, entity relations) provide depth to a world and make them seem more realistic. Haptics have been extensively explored; however, the area, as a whole, is largely under-explored. We believe a large portion of this is simply that they are very challenging.

What exactly does a *dynamic interaction with a dynamic component* (DID) entail? The examples given in the prior section focused on interactions well known from our everyday life. Conversations involve “give and take” methods, such that one actor talks and the other(s) listen, in “turn taking.” Here, rules have emerged in culture to make the DID possible. However, body language may play a role in conversations even between those discrete intervals of speech. Similarly, haptics research largely implements well know sets of rules based on physical properties, i.e. we push on something and the force (an abstraction over time) is feed into the system.

Dynamic interactions with dynamic components that are not based in existing physical and/or cultural rules are not as clear. An abstract example that functions well is Beckhaus' Guided Exploration system [1]. Her system created a hybrid interactive and guided system for exploration of a virtual environment. A passive user would be guided through an environment, from one point of interest to the next. However, the user had the possibility of interrupting the guidance system to navigate on their own. When returning to a passive mode, the guidance system reengaged. The system was designed to be emergent, such that it “created” a different tour through the environment after the interaction intervention. The user's new position started the system anew, and it took into account the points of interests visited by the user while they were in control.

The guided exploration example presents one possible method of DIDs: an informed takeover of control. Conversation involves exclusion in the verbal modality in the turn taking convention. However, when trying to extrapolate from those examples to come to an “essence” of DIDs, one quickly runs into difficulties.

To illustrate this, an example will be made of classical direct manipulation in VR. This interaction is a good example, because it is both easy to visualize and also commonly thought of when first approaching this kind of interaction (though according to the authors' experience, always quickly disregarded when attempting to implement it). Assuming the manipulation being performed is translation, looking at the possible ways in which the dynamic component and the manipulation dynamic component can be combined is informative. Figure 6 visualizes the possibilities listed here:

- add the manipulation to the continuing object movement - changing the position directly
- add the manipulation to the continuing object movement - changing the dynamic at a higher-order level (e.g. velocity)

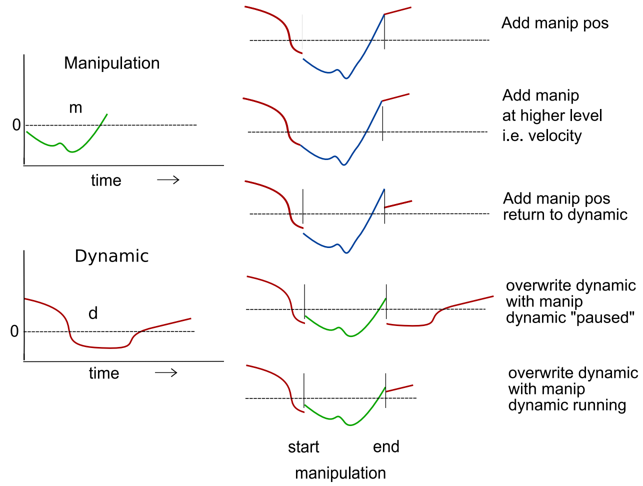

Fig. 6 This diagram shows the identified methods in which direct manipulation and existing dynamic component can be combined. For simplicity, the interaction is reduced to a 2D space. The timings of the interaction and the dynamic component are seen on the left hand side.

– add the manipulation to the continuing object movement - snap back to dynamics afterwards
– stop the movement and manipulate - continue as before (i.e. the manipulation has no effect after manipulation ends)
– stop the movement and manipulate - continue dynamics from the manipulated position

Implementations of each are possible; the larger question is, what manipulations make sense? We feel that the answer to that lies in the application for which it is needed, making it a difficult to answer question.

8 Discussion

As we performed this research, we were often reminded of the immense possibilities for what could be and, yet, so rarely is. This was true even for the areas of interaction and dynamic components. In this section, we discuss briefly some of the insights won during the process of this research. We also discuss briefly two topics of importance related to this work. The first is supporting the creation of DIVEs, a central motivation of this research. The second is some of the ongoing research into the newly identified research areas, in particular *interactive dynamic components* and *dynamic interactions*.

Interaction is the one area that has been extensively researched in the VR community, and yet this is the first work we know of that has asked what is possible. Here, we have shown that many potential avenues of interaction have not yet been deeply explored. Many of these interactions are related to dynamic components, another area that has had limited attention. The Web3D community has been the primary area of research on dynamic components in the recent years [10,16,39].

We believe, however, that the most interesting part of this work is the revelation of the very under-explored areas at the intersection of dynamic components and interaction; we have termed the different ways in which they can combine as *dynamic interaction*, *interactive dynamic components*, and *dynamic interaction with dynamic components*. These different combinations of interaction and action are likely to be compelling and interesting components of VEs. From a research viewpoint, each combination is interesting. A primary reason is the challenge. The combination of components that change over time and interaction with that same object is difficult. It is often difficult to conceive interactions that are functional and user friendly. In turn, these kinds of interactions are usually difficult to implement.

Given the challenges and the huge potential of these areas, it is surprising how little research exists on these topics. Most of the relevant prior research has been mentioned in sections introducing each concept. In the following subsection, we briefly present the system we have developed specifically to support the creation of *Dynamic Interactive VEs*. After that, we introduce some of the research we have been performing on such combinations of dynamic components and interaction.

8.1 Supporting DIVE Creation

The number of existing and previous VE systems is expansive, so it is surprising that only a single system has been designed with DIVE creation in mind. Reviews of the various systems can be found in [6,9]. We briefly introduce the issues surrounding supporting DIVE creation and introduce our system that was designed to fully support DIVE creation.

Systems that support interaction can be readily found, including various systems specifically for interactions. However, support for creating dynamic components is generally lacking in most systems. Most systems that support time based changes, do so by providing the programming language itself and a "timer." In all but a few exceptions, the support systems consider time to be instantaneous and occurring only at the frame update. If "continuous" dynamic components are supported, they are typically enabled only through pre-planned — typically keyframe — animations. Unfortunately, pre-planned animations exclude most interactions. *Interactive dynamic components* and *dynamic interactions* are rarely, if ever, supported in current systems at any level.

Based on the research presented here, we developed a system specifically to support the creation of DIVEs, called Functional Reactive Virtual Reality (FRVR) [6,8,9]. It directly and natively supports time based creation of dynamics of every class identified in Sections 4. It also provides provisions for reactive interaction with those dynamics, thereby supporting every category of DIVE described previously.

FRVR is built upon the Functional Reactive Programming (FRP) paradigm. In FRP, time is a first class part of the system and is considered continuous [19, 60]. What differentiates FRVR from other systems that can handle continuous time is that it: still handles instantaneous time changes well, supports interaction, and is VR system independent. This support of interaction can be for all classes of interaction: instantaneous, interactive dynamic components, dynamic interactions, and even dynamic interactions with dynamic components.

The FRP paradigm focuses on two things, expressing (time based) behaviors as mathematical functions and being reactive to input. Although FRP is not exclusively for time based usages, its strength lies therein. Time is explicit to the system, but hidden from the developer. Instead, typical dynamic components are programmed using integrals. FRVR uses the Yampa FRP implementation, which is written in Haskell using Arrows [15].

FRVR expands on the original FRP paradigm for the needs of Virtual Reality. FRVR adds diverse abilities for traditional usages, like keyframed animation. FRP's reactive system enables switching between "behaviors," and also for emergent and non-linear behavior of the system. FRVR adds support for exploration of the dynamic interaction through various functionalities. Even time is flexible in the FRVR system, manipulatable for all behaviors, independently per behavior, or as groups of behaviors. Time can even be frozen, while still allowing interactions. An undo functionality that is unique to FRVR can even step back through time. As with many of these abilities, undo requires only the addition of a single function.

FRFR is open source and freely available[2]. It is cross platform and can be integrated into any VR system, using any graphics libraries. Currently implementations of FRVR exist for two VR systems and have been demonstrated with five different graphics engines. FRVR is additionally designed to be multiply connected to various input and output generation systems, e.g. sound, that are running at different update rates. To achieve this, FRVR has only a loose coupling to each system, including the VR system, via a blackboard implementation. Additionally, FRVR has a variable simulation rate that can be determined as required by the application.

8.2 Active DIVE Research

A limited amount of research into the newer areas of the DIVE spectrum has been undertaken. We have started research into interactive dynamic components at various levels and introduce those directions here.

As this area is yet under-developed, we have started with the simplest of interactions, selection. In our initial work, we tested various methods of selection of dynamic objects [49]. We attempted various methods of selection

[2] http://imve.informatik.uni-hamburg.de/projects/FRVR

based on expectations gleaned from the developed taxonomies. We performed user tests with four different methods: standard ray picking, the snapping pointer, the time cone, and the trajectory based selections. The last three methods use a cone selection with specialized functions for determining the selected object, based loosely on the research of de Haan, et al. [24]. In a small study we found that the snapping pointer - closest object in cone - and time cone methods - selection based on length of time the object is present in the cone - worked best for single object selection and selection in a occluding group. This work on selection of dynamic objects has continued outside our group in [30] and by others [44].

We have also performed research exploring *dynamic interactions with dynamic components*. Our initial work was with defining and refining of paths through an environment. Here, we have been dealing with the classic issue of defining a “fly through” of an environment. The path is recorded in FRVR and then can be played back. The simplest interaction that can be implemented with FRVR is allowing dynamic changing of the flow of time for the playback (requiring the addition of only a single function). Developing interactions beyond the time manipulation requires some thought.

In order to edit the spatial changes, we experimented with the replacement of sections of the path [3]. This is probably the easiest DID method, at least with the FRVR implementation. We just select a starting time and ending time, replacing it with a newly recorded segment that roughly “matches up.” Currently one watches/experiences the path and triggers the start and end times during the playback (time manipulation can occur during this to increase accuracy if desired). Then the start position and end position are visualized. The user starts from the start position, recording a new path, ending at the end position. Classically such a method would cause discontinuities that are not acceptable, particularly for viewpoint paths. However, using FRVR’s different transitions functions we can smooth the transitions from one to the other, without having to intervene. Usage of transition functionalities requires only the replacement of the switching functionality with a special function that implements the desired transition method.

9 Conclusion

We have introduced and investigated the area of Dynamic Interactive Virtual Environments (DIVEs). These are a class of engaging VEs based on the idea that what makes them interesting are their dynamic and interactive components. This paper defines and explores the design space of this class of environments, providing insights into what is possible, identifying new areas for research, and providing insights into how to support their creation.

We identify five types of components to the environment that have to be regarded: *interaction*, *dynamic components*, *dynamic interactions*, *interactive dynamic components*, and *dynamic interactions with dynamic components*. For each area we have explored and categorized the design spaces of what is possible. This has made explicit the design space of possibilities for each and shed light unto a myriad of exciting areas of research. We have additionally provided an analysis of the dynamic component design space based on how the passage of the time based is perceived, providing insight into implementation requirements for these components. Among the more interesting areas for research formalized through this work are *dynamic interactions* and *interactive dynamic components*, two areas that have large potentials and are widely open for future research.

References

1. Beckhaus, S.: Dynamic Potential Fields for Guided Exploration in Virtual Environments. 6. Fraunhofer Series in Information and Communication Technology (2003)
2. Beckhaus, S., Blom, K.J., Haringer, M.: Concepts and Technologies for Pervasive Games: A Reader for Pervasive Gaming Research vol. 1, chap. ChairIO - the Chair-Based Interface, pp. 231–264. Shaker Verlag (2007)
3. Bernier, E., Chellali, R., Mouttapa, I., Blom, K.: The ICED plugin for virtual reality, immersive creation and edition of displacement. In: Software Engineering and Architectures for Realtime Interactive Systems (SEARIS) Workshop of the IEEE VR (2011)
4. Bierbaum, A., Cruz-Neira, C.: Run-time reconfiguration in VR Juggler. In: 4th Immersive Projection Technology Workshop (2000)
5. Billinghurst, M., Kato, H.: Collaborative augmented reality. Communications of the ACM **45**(7), 64–70 (2002)
6. Blom, K.J.: Dynamic, Interactive Virtual Environments. Sierke Verlag (2009)
7. Blom, K.J., Beckhaus, S.: Emotional Storytelling. In: IEEE Virtual Reality 2005 Conference Workshop "Virtuality Structure", pp. 23–27. Bonn, Germany (2005)
8. Blom, K.J., Beckhaus, S.: Functional Reactive Virtual Reality. In: IPT/EGVE '07: Short Paper Proceedings of the IPT/EuroGraphics workshop on Virual Environments, pp. 295–302. EuroGraphics Association (2007)
9. Blom, K.J., Beckhaus, S.: Supporting the Creation of Dynamic, Interactive Virtual Environments. In: VRST '07: Proceedings of the 2007 ACM symposium on Virtual Reality Software and Technology, pp. 51–54. ACM Press (2007). DOI 10.1145/1315184.1315191
10. Boussinot, F., Susini, J.F., Tran, F.D., Hazard, L.: A reactive behavior framework for dynamic virtual worlds. In: Web3D '01: Proceedings of the sixth international conference on 3D Web technology, pp. 69–75. ACM Press, New York, NY, USA (2001)
11. Bowman, D.A., Hodges, L.: Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. In: Journal of Visual Languages and Computing, pp. 37–53. Academic Press (1999)
12. Brooks, H., DeKeyser, T., Jaskot, D., Sibert, D., Sledd, R., Stilwell, W., Scherer, W.: Using agent-based simulation to reduce collateral damage during military operations. In: Systems and Information Engineering Design Symposium, 2004. Proceedings of the 2004 IEEE, pp. 71–77. IEEE (2004)
13. Bryson, S.: Time, data-time, and real-time interactive visualization. Computer Physics **11**(3), 270–274 (1997)
14. Cavazza, M., Hartley, S., Lugrin, J.L., Bras, M.L.: Alternative reality: a new platform for virtual reality art. In: Proceedings of Virtual Reality Software and Technology (VRST), pp. 100–107. ACM Press (2003). DOI 10.1145/1008653.1008672
15. Courtney, A., Nilsson, H., Peterson, J.: The Yampa Arcade. In: ACM SIGPLAN Haskell Workshop, pp. 7–18. ACM SIGPLAN (2003)
16. Dachselt, R., Rukzio, E.: Behavior3D: An XML-Based Framework for 3D Graphics Behavior. In: Proceedings of the ACM Web3D 2003 Conference, pp. 101–112. ACM Press (2003)
17. Dede, C.: Immersive interfaces for engagement and learning. Science **323**(5910), 66–69 (2009). DOI 10.1126/science.1167311

18. Difede, J., Cukor, J., Jayasinghe, N., Patt, I., Jedel, S., Spielman, L., Giosan, C., Hoffman, H.: Virtual reality exposure therapy for the treatment of posttraumatic stress disorder following september 11, 2001. Journal of Clinical Psychiatry (2007)
19. Elliott, C., Hudak, P.: Functional Reactive Animation. In: International Conference on Functional Programming, pp. 196–203 (1997)
20. Emond, B., Fournier, H., Lapointe, J., et al.: Applying advanced user models and input technologies to augment military simulation-based training. In: Proceedings of the 2010 Spring Military Modeling and Simulation Symposium, pp. 1–7 (2010)
21. Freeman, D., Pugh, K., Antley, A., Slater, M., Bebbington, P., Gittins, M., Dunn, G., Kuipers, E., Fowler, D., Garety, P.: Virtual reality study of paranoid thinking in the general population. Br J Psychiatry **192**(4), 258–263 (2008). DOI 10.1192/bjp.bp.107.044677
22. Gorini, A., Mosso, J., Mosso, D., Pineda, E., Ruíz, N., Ramíez, M., Morales, J., Riva, G.: Emotional response to virtual reality exposure across different cultures: The role of the attribution process. CyberPsychology & Behavior **12**(6), 699–705 (2009)
23. Guger, C., Groenegress, C., Holzner, C., Edlinger, G., Slater, M.: Brain-computer interface for virtual reality control. Cyberpsychology & Behavior **12**(1), 84–84 (2009)
24. de Haan, G., Koutek, M., Post, F.H.: IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection. In: Proceedings of the 9th IPT and 11th Eurographics VE Workshop (EGVE), pp. 201–209 (2005)
25. Haringer, M., Beckhaus, S.: Effect Based Scene Manipulation for Multimodal VR Systems. In: IEEE VR 2010: Proceedings of the IEEE Virtual Reality Conference (2010)
26. Haringer, M., Beckhaus, S.: Adaptive generation of emotional impact using enhanced virtual environments. Presence **21**(1), 96–116 (2012)
27. Hassanpour, R., Wong, S., Shahbahrami, A.: Visionbased hand gesture recognition for human computer interaction: A review. In: IADIS International Conference Interfaces and Human Computer Interaction, p. 125 (2008)
28. Held, R.M., Durlach, N.I.: Telepresence. Presence: Teleoper. Virtual Environ. **1**, 109–112 (1992)
29. Hernando, J., Martinez, J., Martin, V., López, M., Martin-Gago, J.: Svis: A computational steering visualization environment for surface structure determination. In: Visualisation, 2009. VIZ'09. Second International Conference in, pp. 36–39. IEEE (2009)
30. Herrlich, M., Walther-Franks, B., Schröder-Kroll, R., Holthusen, J., Malaka, R.: Proxy-based selection for occluded and dynamic objects. In: Smart Graphics, pp. 142–145. Springer (2011)
31. Hodges, L., Rothbaum, B., Alarcon, R., Ready, D., Shahar, F., Graap, K., Pair, J., Hebert, P., Gotz, D., Wills, B., et al.: A virtual environment for the treatment of chronic combat-related post-traumatic stress disorder. Cyberpsychology & Behavior **2**(1), 7–14 (1999)
32. Hoffman, H., Garcia-Palacios, A., Patterson, D., Jensen, M., Furness III, T., Ammons Jr, W.: The effectiveness of virtual reality for dental pain control: a case study. Cyberpsychology & Behavior **4**(4), 527–535 (2001)
33. Kallmann, M., Thalmann, D.: Direct 3D Interaction with Smart Objects. In: VRST '99: Proceedings of the ACM symposium on Virtual Reality Software and Technology, pp. 124–130. ACM Press, New York, NY, USA (1999). DOI 10.1145/323663.323683
34. Konieczny, J., Heckman, J., Meyer, G., Manyen, M., Rabens, M., Shimizu, C.: Automotive spray paint simulation. Advances in Visual Computing pp. 998–1007 (2008)
35. Kulik, A.: Building on Realism and Magic for Designing 3D Interaction Techniques. IEEE Computer Graphics and Applications **29**(6), 22–33 (2009). DOI 10.1109/MCG.2009.115
36. Mateas, M., Stern, A.: Facade: An Experiment in Building a Fully-Realized Interactive Drama. In: Proceedings of the Game Developer's Conference: Game Design Track (2003)
37. McLay, R.N., Wood, D.P., Webb-Murphy, J.A., Spira, J.L., Wiederhold, M.D., Pyne, J.M., Wiederhold, B.K.: A randomized, controlled trial of virtual reality-graded exposure therapy for post-traumatic stress disorder in active duty service members with combat-related post-traumatic stress disorder. Cyberpsychology behavior and social networking **13**(1), 3–11 (2010)

38. van der Meijden, O., Schijven, M.: The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review. Surgical endoscopy **23**(6), 1180–1190 (2009)
39. Mesing, B., Hellmich, C.: Using Aspect Oriented Methods to Add Behaviour to X3D Documents. In: Proceedings of the ACM Web3D 2006 Conference. ACM (2006)
40. Misra, S., Ramesh, K., Okamura, A.: Modeling of tool-tissue interactions for computer-based surgical simulation: a literature review. Presence: Teleoperators and Virtual Environments **17**(5), 463–491 (2008)
41. Pape, D.: Crayoland. In: SIGGRAPH '98: ACM SIGGRAPH 98 Electronic Art and Animation Catalog, p. 116. ACM Press, New York, NY, USA (1998)
42. Rehn, G.D., Lemessi, M., Vance, J.M., Dorozhkin, D.V.: Integrating operations simulation results with an immersive virtual reality environment. In: WSC '04: Proceedings of the 36th Winter Simulation Conference, pp. 1713–1719 (2004)
43. Reiley, C., Lin, H., Yuh, D., Hager, G.: Review of methods for objective surgical skill evaluation. Surgical endoscopy **25**(2), 356–366 (2011)
44. Richard, P., Naud, M., Inglese, F.X., Richard, E.: Modeling dynamic interaction in virtual environments and the evaluation of dynamic virtual fixtures. Presence: Teleoperators and Virtual Environments **21**(3), 321–337 (2012)
45. Rizzo, A., Rothbaum, B.O., Graap, K.: Combat stress injury: Theory, research, and management, chap. Virtual reality applications for the treatment of combat-related PTSD, pp. 183–204. Routledge/Taylor & Francis Group, New York (2007)
46. Roussos, M., Johnson, A.E., Leigh, J., Vasilakis, C.A., Barnes, C.R., Moher, T.G.: NICE: combining constructionism, narrative and collaboration in a virtual learning environment. Computer Graphics **31**(3), 62–63 (1997). DOI 10.1145/262171.262264
47. Salzmann, H., Moehring, M., Froehlich, B.: Virtual vs. Real-World Pointing in Two-User Scenarios. In: VR '09: Proceedings of the 2009 IEEE Virtual Reality Conference, pp. 127–130. IEEE Computer Society, Washington, DC, USA (2009). DOI 10.1109/VR.2009.4811011
48. Sanchez-Vives, M.V., Slater Mellers, B.: From Presence to Consciousness through Virtual Reality. Nature Reviews Neuroscience **6**(4), 332–339 (2005)
49. Schröder-Kroll, R., Blom, K.J., Beckhaus, S.: Interaction Techniques for Dynamic Virtual Environments. In: Proceedings of "Virtuelle und Erweiterte Realität" Workshop of the Gesellschaft für Informatik e.V. (GI), pp. 57–68. Shaker Verlag (2008)
50. Sharar, S., Miller, W., Teeley, A., Soltani, M., Hoffman, H., Jensen, M., Patterson, D.: Applications of virtual reality for pain management in burn-injured patients. Expert review of neurotherapeutics **8**(11), 1667 (2008)
51. Sheridan, T.: Musings on telepresence and virtual presence. Presence: Teleoperators and virtual environments **1**(1), 120–126 (1992)
52. Singhal, S., Zyda, M.: Networked Virtual Environments. Design and Implementation. Addison Wesley, Reading, MA. (1999)
53. Slater, M.: Place Illusion and Plausibility can lead to realistic behaviour in immersive virtual environments. Philos Trans R Soc Lond B Biol Sci **364**(1535), 3549–3557 (2009). DOI 10.1098/rstb.2009.0138
54. Slater, M., Lotto, B., Arnold, M.M., Sanchez-Vives, M.V.: How we experience immersive virtual environments: the concept of presence and its measurement. Anuario de Psicologia **40**(2), 193–210 (2009)
55. Steed, A.: Towards a General Model for Selection in Virtual Environments. In: proceedings of the IEEE Symposium on 3D User Interfaces (3DUI), pp. 103–110. IEEE (2006)
56. Steuer, J.: Defining virtual reality: Dimensions determining telepresence. Journal of communication **42**(4), 73–93 (1992)
57. Takatalo, J., Nyman, G., Laaksonen, L.: Components of human experience in virtual environments. Computers in Human Behavior **24**(1), 1–15 (2008)
58. Thomas, F., Torras, C.: 3D Collision Detection: A Survey. Computers and Graphics **25**, 269–285 (2001)
59. Van Dam, A., Forsberg, A., Laidlaw, D., LaViola Jr, J., Simpson, R.: Immersive VR for scientific visualization: A progress report. Computer Graphics and Applications, IEEE **20**(6), 26–52 (2000)

60. Wan, Z., Hudak, P.: Functional reactive programming from first principles. In: Conferenc on Programming Language Design and Implemetnation (PLDI'00), pp. 242–252. ACM SIGPLAN, ACM Press (2000)
61. Wu, W., Arefin, A., Rivas, R., Nahrstedt, K., Sheppard, R., Yang, Z.: Quality of experience in distributed interactive multimedia environments: toward a theoretical framework. In: Proceedings of the Seventeen ACM international Conference on Multimedia, pp. 481–490. ACM (2009)
62. Xian-Yi, C., Yan, P.: Review of modern speech synthesis. In: W. Hu (ed.) Electronics and Signal Processing, *Lecture Notes in Electrical Engineering*, vol. 97, pp. 517–524. Springer Berlin Heidelberg (2011)
63. Yang, U., Lee, G., Kim, Y., Jo, D., Choi, J., Kim, K.: Virtual reality based welding training simulator with 3D multimodal interaction. In: Cyberworlds (CW), 2010 International Conference on, pp. 150–154. IEEE (2010)
64. Yannakakis, G.N., Hallam, J.: Real-time Game Adaptation for Optimizing Player Satisfaction. IEEE Transactions on Computational Intelligence and AI in Games **1**(2), 121–133 (2009)
65. Zahorik, P., Jenison, R.: Presence as being-in-the-world. Presence **7**(1), 78–89 (1998)
66. Zander, T.O., Kothe, C.: Towards passive brain-computer interfaces: applying brain-computer interface technology to human-machine systems in general. Journal of Neural Engineering **8**(2), 025,005 (2011)