Int J Interact Des Manuf (2008) 2:17–32
DOI 10.1007/s12008-007-0032-2

ORIGINAL PAPER

A taxonomy for the design and evaluation of Networked Virtual Environments: its application to collaborative design

Miranda Baladi · Henry Vitali · Georges Fadel · Joshua Summers · Andrew Duchowski

Received: 11 September 2006 / Revised: 10 October 2006 / Accepted: 13 October 2006 / Published online: 20 December 2007

Abstract Networked Virtual Environments (NVEs) are virtual environments that are distributed across two or more physical locations and connected over a network, thereby forming one virtual shared workspace. In the past decade, research in the field of networked virtual environments has become active, quickly growing out of infancy into a diversity of applications. As is common in any new field, this growing assortment of applications lacks an overall classification, making the design of NVEs cumbersome and the comparison of existing environments difficult at best. The goal of this paper is to provide a means of classification through the development of a taxonomy of NVEs. First, an extensive literature review is conducted investigating the characteristics of virtual environments in general, as well as issues specific to networked virtual environments. This search leads to the development of a taxonomy for describing NVEs, which is used to compare current NVE systems and applications and find weak areas in which future work might be most beneficial. Finally, the taxonomy is used for the development of networked virtual environments for collaborative mechanical engineering design.

Keywords Networked Virtual Environments (NVEs) · Taxonomy · Virtual reality

M. Baladi
Fluor Corporation, Greenville, SC, USA
e-mail: Miranda.Baladi@fluor.com

H. Vitali
New South Companies, Columbia, SC, USA
e-mail: henry.vitali@newsouthcompanies.com

G. Fadel (✉) · J. Summers
Department of Mechanical Engineering,
Clemson University, Clemson, SC, USA
e-mail: fgeorge@clemson.edu

J. Summers
e-mail: jsummer@clemson.edu

A. Duchowski
Department of Computer Science,
Clemson University, Clemson, SC, USA
e-mail: aduchow@clemson.edu

1 Introduction

Networked Virtual Environments (NVEs) are virtual environments, which are distributed across two or more physical locations, connected over a network thus forming one virtual space. NVEs have a potential to enhance concurrent engineering and the collaborative design process. In contrast to most computer applications, which rely on the use of symbols and icons for the user to interact, Virtual Reality (VR) provides more naturalistic user interfaces [1]. VR applications are able to present the user with large amounts of information in an easily understandable format [2]. This is important during the design process, where the presentation of information may affect the creative generation and evaluation of ideas. NVEs enable the collaboration of remote participants, allowing them to communicate with each other about the objects and models in the virtual environment. This has the potential to impact travel (time and cost) required for collocated collaboration, resulting in a reduced time to market and development cost.

Research in the field of networked virtual environments has become active only in the past ten years, as high performance networks using T1 lines or ASDL and improved multicasting methods have become available. Despite the relative youth of the field, there is now a diversity of NVE applications; yet, no unified terminology has emerged. Even the definition of VR is under dispute [3]. This is compounded further with a lack of general agreement on what constitutes a NVE. In some cases, a networked virtual environment refers to any semi-immersive environment supported by a network of computers, even if only for 2D gaming over the Internet. For the purposes of this research, a virtual environment implies immersive, interactive VR in 3D stereo. A NVE therefore refers to a virtual environment connected over a network between multiple sites.

The lack of an overall classification makes the design of NVEs complicated and their comparison and evaluation difficult at best. To help deal with this challenge, this research explores the issues relating to NVEs and develops a classification taxonomy to aid in design and development. This classification is then used to describe current NVE systems, compare them, and make recommendations for the design of a NVE for collaborative engineering design. This taxonomy represents a first step toward the use of structured methods for designing NVEs. Additionally, this taxonomy has the potential to help determine weak areas in the field for future work and facilitate the use of prescriptive methods for the design of future applications.

2 Literature review

2.1 Taxonomy: the science of classification

The term, taxonomy, coined by the famous French botanist de Candolle [4], is derived from two words: taxis, meaning arrangement, and nomos, meaning study. Thus, taxonomy is the study of the methods and principles of classification. Jeffrey [5] describes classification as the assignment of similar objects to distinguishable groups. In this way, classification facilitates reference to the objects and dissemination of information about them, enabling knowledge to be more easily summarized. Derr [6] provides a simpler explanation of classification, stating simply that "classification adds order and clarity to information". As systems become more complex, order and clarity are much harder to achieve. This has led to the fact that taxonomies are commonly used to classify large bodies of information [7]. Jeffrey [5] has also made comments that large size and great diversity are causes of a need for classification. A few definitions involving the quality and the structure of taxonomies must be clarified in order to ensure a good understanding of the problem. These next sections describe the basic concepts of taxonomy: perceptual orthogonality, completeness, and parallel structure.

Perceptual orthogonality preserves the uniqueness of every classification or taxon and every category. Slaughterbeck-Hyde [8] used the example of a taxonomy of trees to illustrate perceptual orthogonality. If trees are divided into two categories, deciduous and conifers, orthogonality is satisfied if a tree could be classified as one or the other, but not both. Not all authors have used the term perceptual orthogonality to describe this important attribute, but all comment on its importance. Derr [6] specifies that classes should be mutually exclusive, while Jeffrey [5] states more generally that "all members of any taxon should resemble one another in the sum total of all their characters more closely than they resemble the members of any other taxon in the same rank". Dunn and Everitt [9] refer instead to distinctness, setting the requirement that a taxon must be "sufficiently distinct to be worthy of being assigned to a definite category". If taxons are not distinct, it will be unclear where to store or retrieve requirements, and the taxonomy will no longer aid the development process [7]. Therefore classes must be perceptually orthogonal at the time that the classes are named, meaning that the classes may be subject to change as the field evolves.

Completeness is a simple term meaning that a taxonomy should include all known aspects of the field within the domain being classified. It should be noted that some also refer to completeness as the characteristic of being exhaustive [6,8,10]. Yet, just as is the case with perceptual orthogonality, completeness is a relative measure as well as an important goal. Gershenson and Stauffer [7] explain this point well when they state.

"It is never possible to classify a whole subject; it is only possible to classify our knowledge of it. This implies that our knowledge of any subject is neither exhaustive nor objective."

Parallel structure considers the hierarchical nature of a taxonomy. In biological taxonomy, groups of taxons are classified together as Kingdoms, Phylum, Classes, Families, Species, plus additional levels in between. The goal of a parallel structure is to be able to have similar levels of abstractness for taxons of the same rank. This goal may be attainable in some cases, but it is not always so clear. In the case of biological taxonomy, it is clear only for species, which have the requirement of reproduction amongst members of the same taxon. Parallel structure is not an issue that has been well addressed in the first trial taxonomy for NVEs, which is proposed in this paper. Future versions should consider this aspect.

Perceptual orthogonality, completeness, and parallel structure are the defined goals, but there is no one right way to approach them, and no way to achieve them perfectly. In the development of a taxonomy, there are many ways in which a domain can be classified [11]. The taxonomy developed in this research classifies NVEs in an organized manner, but it does not take credit for being the only solution to classification. Its value can only be assessed by its usefulness to others [9]. This taxonomy provides a starting point for classification of NVEs. It is not meant to be a final complete tool, but instead a living one, which will evolve as the field matures.

2.2 Previous work in NVE classification

The creation of a virtual environment requires expertise in many areas (e.g. graphics programming, networking, rendering). VR has a wide range of applications, including medicine (e.g. [12]), education (e.g. [13]), military (e.g. [14]), and mechanical engineering (e.g. [15]). With such diversity, it is imperative that there be a system of classification for researchers to be able to communicate and compare their efforts. How does one begin to compare the features of a tele-surgery application with a virtual classroom? Are these systems too distinct to be compared on the same level? On which level can they be compared? Are there any commonalities? The need for a tool to aid in answering these questions makes the development of a taxonomy critical for understanding distributed virtual environments.

This task is compounded as a result of the lack of a common vocabulary for describing NVEs. A NVE incorporates multiple sites into a single virtual environment via a network; however other definitions of NVEs imply only the integration of multiple participants [16]. "Collaborative virtual environment" is typically meant to be a specific type of NVE connecting multiple users for cooperative object manipulation but may also describe multi-user environments with no distribution [17]. Often, each researcher introduces their vocabulary, resulting in multiple terms to describe the same feature. As an example, the term "networked collaborative virtual environment" (NCVE) is used to describe a typical "collaborative virtual environment" [18]. These discrepancies make evaluating accomplished work and ascertaining areas of research that are heavily studied from areas in which much research remains to be done (i.e. where the best contribution can be made) difficult.

Taxonomies have been proposed or are in development for VR User Interfaces (VRUIs) [2], computer graphics [19], interactive graphics systems [20], and Distributed Object Models (DOMs) [21]. Each of these frameworks is derived from different perspectives and contains elements that are important to VR systems. There is still a need for a "coherent framework for understanding distributed virtual environments" say Macedonia and Zyda, [22] when they propose an initial NVE Taxonomy. From their work and others, we see that a preliminary taxonomy for NVEs centers on issues of large scale environments such as scalability and network communications [22]. Scalability may be a major issue for environments such as for military simulation, which include large numbers of participants, but is not of great concern for other limited participant applications such as tele-surgery. Further, certain critical issues (e.g. quality of service and persistence) are not accommodated. This paper attempts to go further and build on Macedonia and Zyda's work to provide a more complete view, beneficial to anyone involved with NVEs.

3 Deriving the taxonomy

The following sections describe the development of the taxonomy based upon the available literature.

3.1 Taxonomy of virtual environments

3.1.1 Defining VR

One of the largest challenges in developing a taxonomy is sorting through the reuse, misuse, and original use of terminology. The fields of VR and NVEs are relatively new, resulting in quickly changing and often conflicting terms and definitions within these fields. These difficulties must be overcome at all levels of the taxonomy. Definitions of VR will be used as a baseline for developing the taxonomy, extracting requisite and distinguishing characteristics. This classification of VR is expanded for NVE.

One approach which many authors have taken to define VR is to consider it from a technology-based standpoint:

- "VR is an alternate world filled with computer-generated images that respond to human movements" [23]
- "VR can be described as the science of integrating man with information. It consists of three-dimensional, interactive, computer generated environments" [24]
- "A virtual environment is the representation of a computer model or database which can be experienced and manipulated by the virtual environment participant(s)" [25]
- "A virtual environment is a computer-simulated world consisting of software representations of real (or imagined) agents, objects, and processes; and a human-computer interface for displaying and interacting with these models" [26]
- "VR is a set of computer technologies which, when combined, provide an interface to the computer with which the user can believe he or she is actually in a computer-generated world" [83]

These definitions highlight the computer-generated, -simulated, -oriented aspect of virtual environments, attempting to define VR through the technology used to create it. These descriptions help to describe VR systems and the virtual experience, however from a design standpoint they are not as useful. They focus the designer's attention on the final product, without attention given to how to achieve that goal. In addition, these definitions do not uniquely characterize VR, as many modern computer-aided design (CAD) applications would also fit under their descriptions. Only the last definition excludes CAD applications with the requirement that the VR should provide interfaces that immerse the user in the environment.

Other authors have attempted to define VR by the functions and experience that it provides to the user, considering the concepts of immersion and presence:

- "VR is defined as a real or simulated environment in which a perceiver experiences telepresence where

tele-presence is 'the experience of presence in an environment by means of a communication medium"' [27]
- VR is the use of various computer graphics systems in combination with various display and interaction devices to provide the effect of 'immersion' in an interactive three-dimensional computer-generated environment in which the virtual objects have 'spatial presence' [28]

These definitions provide a more functional view. They also more uniquely describe VR, as no other systems have the ultimate goals of providing the user with a sense of presence. CAD applications, while allowing the user to interact with 3D models, do not attempt to immerse the user in the 3D world. In terms of collaboration in mechanical design, presence also differentiates collaborative VR from multi-user CAD applications.

Most characteristics of VR applications have been described in terms of presence, to the degree where some propose a framework for research in VR with presence as the central feature [29]. One definition of VR is based entirely upon the concept of presence [27]. Presence is examined in the next section.

3.1.2 Presence

Presence is defined as the perception that a user has of being immersed in the VR world [30]. Three dimensions of presence are personal, social, and environmental [31]. Personal presence, which is subjective and dependent upon each user, measures the extent to which the user feels present in the virtual world. Some aspects that contribute to personal presence include: user embodiment, sense of place, persistence, consistent recognizable patterns of behavior, and simulation of intensity and range of senses (similar to the concept of perceptual position [32]). Social presence relates to the interaction with other participants (real or virtual). Social presence is a component of personal presence, but as it such an important aspect to enhancing presence, it is addressed separately [31]. Environmental presence relates to the interaction with objects in the environment. Environmental presence includes the user's ability to make changes to the environment (e.g. moving objects) as well as the environmental responses to the user's presence (e.g. a door opening for the user).

Two dimensions of presence are vividness and interactivity [27]. Vividness is the extent (breadth and depth) to which sensory information is provided. Breadth is the number of senses engaged and depth is the resolution of the engagement. Interactivity is the extent to which the user can make changes to the environment in real time. This includes the speed (system response time), range (number of simultaneous interactions), and mapping (naturalness of the relation of the system controls to changes in the environment).

One taxonomy has been proposed for presence that is divided into external and internal factors [32,33]. External factors deal with how the environment is generated: display quality, consistency, interaction, anthropomorphism, and clarity of causal relationships. Internal factors deal with how humans sense the environment internally including the visual, auditory, and kinesthetic senses and the perceptual position (egocentric, exocentric, and external).

Lombard and Ditton [34] looked toward the literature to determine how researchers from various schools of thought consider the concept of presence. Their search resulted in six conceptualizations of presence. They admit that the six are interrelated, yet assert they are nevertheless distinct:

- Presence as social richness—the extent to which the environment 'is perceived as sociable, warm, sensitive, personal or intimate'
- Presence as realism—the extent to which the environment provides accurate representations of real world entities
- Presence as transportation—further decomposed into three forms: transporting the user to another place ('you are there'), transporting places and objects to the user ('it is here'), and bringing participants together in a shared space ('we are together')
- Presence as immersion—considers the aspect of immersion of the users senses leading to a sense of psychological immersion. Presence as social actor within medium—comes when the users begin to overlook or forget the artificial nature of the entities within the environment and attempt to interact with them
- Presence as medium as social actor—has to do with the environment exhibiting social cues that lead the users to treat the environment itself as a social entity

Upon examining this last type of presence, Barfield et al. [26] further stipulate that the environment should be constant and continuous across time, thus, introducing persistence as an additional aspect of presence as a medium as social actor. Kalawsky [35] identifies field of view, display resolution, level of physical interaction, and seeing parts of one's own body as factors (proprioception), which have an effect on the feeling of presence.

Zeltzer [36] presents a cubic model of virtual environments with three axes: autonomy, interaction, and presence (AIP model). Autonomy is the extent to which the objects in a virtual environment are able to actively change their own states, including their response to events. Interaction is the ability of the environment to respond to inputs from the user in real-time as well as the means by which interaction takes place. Presence deals with the range, number and quality of sensory modalities engaged. Some have found his taxonomy to be useful in describing, comparing, and contrasting virtual

			Personal presence	Social presence	Env. presence	Breadth	Depth	Speed	Range	Mapping	Ext. of sens. info.	Control of sensors	Ability to modify	Display quality	Consistency	Interaction	Anthromorphism	Clarity of causal relationships	Representation	Perceptual Position	Social richness	Realism	Transportation	Immersion	Soc. actor in med.	Medium as social actor	Autonomy	Interaction	Presence
	Personal presence		X																										
	Social presence		4	X																									
Heeter	Environmental presence		4	0	X																								
		Breadth	5	0	4	X																							
	Vividness	Depth	5	0	4	0	X																						
		Speed	5	4	5	0	0	X																					
		Range	5	4	5	0	0	0	X																				
Steuer	Interactivity	Mapping	5	4	5	0	0	0	0	X																			
	Extent of sensory information		5	0	0	2	2	0	0	0	X																		
	Control of sensors		5	0	0	0	0	2	2	2	0	X																	
Sheridan	Ability to modify the environ.		4	0	5	0	0	2	2	2	0	0	X																
		Display Quality	5	0	0	2	2	0	0	0	2	0	0	X															
		Consistency	5	0	0	0	0	0	0	0	0	0	0	4	X														
	External Factors	Interaction	4	4	5	0	0	2	2	2	0	3	1	0	0	X													
		Anthromorphism	5	0	0	0	0	0	0	0	0	0	0	4	4	0	X												
		Clarity of causal relationships	5	0	5	0	0	0	0	2	0	0	5	0	0	5	0	X											
	Internal Factors	Representation	5	0	4	5	5	0	0	0	5	5	0	5	4	4	4	0	X										
Slater		Perceptual position	5	0	0	0	0	0	0	5	0	5	0	0	0	0	5	0	0	X									
	Social richness		0	0	0	3	3	0	0	3	0	2	0	3	3	0	2	3	0	2	X								
	Realism		5	2	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	X							
	Transportation		5	2	4	2	2	2	2	2	2	2	2	2	2	2	3	2	2	2	0	2	X						
	Immersion		5	0	0	2	2	0	0	0	2	0	0	2	3	0	0	0	2	0	0	3	3	X					
	Social actor within medium		4	0	5	0	0	2	2	2	0	2	2	0	0	5	0	2	0	0	0	4	0	3	X				
Lombard	Medium as social actor		0	0	3	0	0	0	0	3	0	0	0	0	0	3	0	3	0	0	4	0	0	0	4	X			
	Autonomy		4	4	2	0	0	0	0	0	0	0	3	0	0	5	0	0	0	0	0	0	0	0	0	5	X		
	Interaction		4	4	5	0	0	2	2	2	0	2	2	0	0	5	0	0	0	2	0	5	4	0	5	0	0	X	
Zeltzer	Presence		5	0	0	2	2	0	0	0	1	0	0	2	2	0	2	2	2	0	5	5	5	5	4	4	0	0	X

Guide to reading the table: left column (x), top row (y); scale of 1-5, 0 => no overlap, 1 => x and y are essentially the same, 2 => x entirely encompasses y, 3 => x particially encompasses y, 4 => x is partially a member of y, 5 => x is entirely a member of y

Fig. 1 Orthogonality matrix for characteristics of virtual environments

environment systems [26,35]. Zeltzer's model could also be used to compare CAD applications with VR applications. Each of these might have varying degrees of interaction and autonomy (such as objects changing color when activated), but most CAD applications would be located on the zero presence axis, while VR applications strive for higher presence values.

3.1.3 The orthogonality test

A literature search has been conducted in order to create a complete representation of VR using definitions from multiple sources as discussed in the previous sections. In the development of a taxonomy, orthogonality (overlap between definitions) poses the next challenge. A test for orthogonality has been conducted by placing all of the characterizations of VR given by the literature in a matrix against each other, and assigning numerical values to the degree of overlap between definitions. The resulting orthogonality matrix can be seen in Fig. 1.

The orthogonality matrix does not compare the relationships between the characteristics, but instead only shows overlap in the definition of terms as each author has defined them. For example, consistency between displays has a large effect on the user's ability to interact, but there is no overlap in the meanings of these two terms. Once overlapping terms have been identified, the next challenge is to reduce the many categorizations down into a single taxonomy. A new taxonomy of virtual environments is derived by eliminating broad categories, entirely overlapping categories, and non-distinct categories and hierarchically structuring categories contained within other categories.

The next step is to adapt this taxonomy for NVEs, by incorporating the additional requirements that networking and multiple participants might impose on a system. The current use of protocols and other networking technologies are discussed, as they relate to the issues described within the taxonomy framework.

3.2 Types of NVEs

A taxonomy provides an objective approach that would allow anyone with sufficient knowledge of a system to classify the latter according to its distinguishing characteristics. In order to accomplish this goal, various types of NVEs are discussed, and their special features elaborated. There are five general types of NVEs found in the literature: Virtual-conferencing [37–39]; Collaborative Virtual Environments [40–42]; Teleoperation [43,44]; Teleimmersive [45,46]; and large-scale systems [47–52]. These terms are taken from the literature where authors have attempted to self-categorize their applications. Each type of NVE has distinguishing characteristics with which they can be compared and classified. However, these five categories are not entirely exclusive. This section provides descriptions of each of these categories, offers insight into their distinguishing characteristics, and discusses the overlap that exists between these groups.

Virtual-conferencing environments connect multiple users together in a virtual environment so that they may socially interact and communicate with each other. Often these environments make use of real time audio and video, which are mapped onto virtual screens and/or avatars, in order to provide realistic user embodiment and a sense of presence. For virtual-conferencing systems, the quality of social interaction is a priority. Therefore, many of these systems focus on characteristics that enhance social interaction, such as portraying gaze direction and facial expression. Facial expression provides emotional cues; a smile indicates happiness or raised eyebrows show surprise. Two methods of implementation include texture-mapping video [53] and capturing facial movements using sensors on the skin [37]. In contrast to video conferencing systems, virtual-conferencing applications have the potential to support gaze awareness. During a videoconference, attendees at remote location communicate through the use of streaming video and audio. Yet when a participant looks into the camera, the participant appears to be looking at everyone at the remote locations simultaneously. This inhibits participants from speaking on a one to one basis or knowing who is speaking to whom. Virtual environments on the other hand can support gaze direction through the use of mobile avatars [54].

Collaborative Virtual Environments (CVEs) also incorporate multiple users, while additionally providing users with modes of interaction with objects in the virtual environment. Their generic functionality provides mutual awareness, communication, and collaborative object manipulation [55]. These environments are collaborative in the sense that they support group work, yet the technological capabilities of current CVEs might be better described as cooperative. Cooperation is the joint editing of shared objects, with only one user having access at a time while collaboration implies concurrent, multi-user access [56].

Teleoperation environments allow a participant to experience a sense of tele-presence by conducting work at a physically remote location. Typically the user's senses and actions are coordinated with a robot at the remote sight, thanks to force feedback. The sense of touch is particularly important for teleoperation environments to realize tasks requiring a precise manipulation of objects [57]. Research in teleoperation applications has been conducted for work in hazardous environments like space [43], or in otherwise inaccessible locations such as desired for tele-surgery [58] and micromanipulation (microteleoperation) [44].

Teleimmersive applications integrate supercomputing resources and massive data stores into virtual environments across a distance over high-speed networks [45]. These are commonly used for systems with intensive simulation or visualization requirements. The Electronic Visualization Lab at the University of Illinois Chicago has developed televirtuality systems for collaborative viewing of massive data sets [46,59].

Large-scale NVEs involve large numbers of participants. Some of the greatest challenges in accommodating these large numbers of users are in scalability of the environments, admission and withdrawal of members, and maintaining consistency of states among users. Examples of large-scale virtual environments are Simnet [47,50], NPSNET [22], AVIARY [49], Bamboo [52], and Spline [52].

In examining the types of NVE's that are "self classified" in the literature, several different characteristics are extracted that appear to be distinguishing factors. These features include: gaze awareness, facial gestures, multiple users, cooperative editing, remote manipulation, and intensive real-time computation resulting in the need for a networked supercomputer. These features are integrated into the final NVE taxonomy in the following sections.

3.3 Additional requirements of NVEs

There are some concerns that go above and beyond that which is required for single site virtual environments, yet are common to all NVEs to varying degrees. These include aspects such as latency, bandwidth, reliability, and consistency.

3.3.1 Responsiveness

For networked applications, latency, bandwidth, reliability, and consistency are intimately related. Together they determine the system's responsiveness to changes in the environment. Considering this for the taxonomy, Steuer's concept of speed is replaced with responsiveness, in order to include the effects of reliability and consistency [27]. Incorporating these aspects, responsiveness is defined as the ability of the systems to respond to user inputs while providing consistent views to all participants in real-time.

The speed with which the user can witness his or her interactions in the virtual world is an important aspect. For real-time systems, the latency, or the time it takes between input from the user and its associated effect to take place in the virtual environment [22], should be both less than 0.1 second and no longer than the time of a single graphics frame [28]. In a single processor system, latency is the time it takes for changes to be acknowledged by the operating environment, computed, and the new state rendered. When the environment is distributed on a network between multiple sites, latency becomes an even larger concern. Not only must the system deal with the latency at each computer, but it must also be able to handle large network latencies.

Network latency is introduced into the system in a number of ways. First, information traveling between networked sites, like those distributed across wide area networks (WANs), takes time to travel from one site to the other. Thus, the larger the distance between sites and the slower the connection, the larger is the induced latency. Unfortunately the path along a network is not a straight line. Secondly, all information must be processed and transferred by switches and routers at each intersection, resulting in routing latency. The third source, caused by an influx of information passing over the network, is intimately related to bandwidth.

Bandwidth is the rate that information can be transferred over a network [22]. It depends both on the type of network cables as well as the connecting hardware. For example, traditional phone lines have a maximum bandwidth of 64 kbps, but a modem user is limited to 14.4–56 kbps. Ethernet carries 10 mbps up to 1 Gbps, and fiber optic cables can carry 10 Gbps or more [48]. In addition, the distance between sites may also have an indirect effect on the available bandwidth, as sites, which are very close together, may be connected on a local area network (LAN), and have higher available bandwidth than wide area networks (WANs) or even long-distance leased lines. However, continual technology advances bring ever-increasing amounts of available network bandwidth, which will open new opportunities for advances in NVEs.

In networked applications, bandwidth has a huge impact on the speed of interactions. Data, which is sent over a network must be broken down into packets at the sending site, transmitted in some sequence, and reassembled at the receiving sites. Low bandwidth networks have much larger system delays than their high bandwidth counterparts, since they do not allow much information to pass through the network at one time, and thus create 'long lines' of information waiting to be sent. Note that the user cannot sense low or high network bandwidth directly; instead, low bandwidth is perceived as longer latency. Therefore, the choice of network and corresponding amount of available bandwidth is not a functional requirement of the system, but a system design option.

Reliability is a measure of how much of the data sent by one host is lost or corrupted before another host receives it. In other words, it is the amount of data lost as it travels across the network from one source to another destination [22]. When data is sent across a network, network failures, and exchanges at routers can result in lost data [16]. In real-time applications, sequencing errors are also common, causing data to be processed in a different order than intended [L1]. The reliability requirements of an application determine the type of Internet protocol used to send information across the network, such as User Datagram Protocol (UDP) or Transfer Control Protocol (TCP), and how often data must be resent [22]. Systems requiring high reliability have higher network delays, as resending packets introduces lags in the system.

Consistency in a NVE deals with maintaining coherence in the shared database among all clients. In a networked multi-user environment, information regarding the environment is replicated at each site for quick access and in order to reduce the amount of communication over the network. Multiple participants may be accessing and making changes to the shared database at one of the replicated sites at any given time. Since modifications must be sent across the network with induced latency, a possibility exists that another participant might attempt to edit a parameter that has already been changed at another site, but has not yet been received over the network, or has been lost (reliability). For example, one user might attempt to change the diameter of a hole that should no longer be there, because another participant at another site had deleted it a few milliseconds before.

In order to ensure that each participant views the same environmental state at any moment in time, there must be methods in place to resolve inconsistencies, i.e. determine the state of the virtual object in the midst of conflicting manipulations. Consistency mechanisms must either be able to decide which changes to accept and which to discard, such as accepting the modifications of the first user to access the hole and ignoring the second person's attempt, or know how to combine simultaneous manipulations, thereby leading towards collaborative operations [60]. Thus, a NVE is consistent, when all processes receive events in the order in which they occur, so that the state of the virtual environment is the same at every host at any given time. Each participant may view the environment from a different perspective, but the state of the environment should be consistent with respect to space and time.

There are multitudes of consistency mechanisms in the literature, from the strictest methods such as sequential consistency [61] and processor consistency [62] to less strict methods like weak [63], release [64], and entry consistency [84]. Approaches to maintaining consistency in distributed multi-user systems can be classified either as conservative or optimistic [65].

Conservative approaches apply the strictest consistency mechanisms, attempting to avoid any type of inconsistency between participants' views. However, this is often at the expense of interactivity [60], causing longer latencies and huge bandwidth consumption. Some conservative approaches use locking mechanisms to avoid concurrent interaction. The DIVE system [66], based on the ISIS toolkit [67], issues requests for locks on object access when one participant attempts modification. The locks remain on the object until the participant finishes modifying the object in its local database, and a message is sent to all other clients, who update their replicas accordingly.

Optimistic approaches attempt to be less obtrusive to the user, enabling faster interactions, but often only correct for inconsistencies after they arise. Large-scale environments often employ optimistic approaches to reduce the already large amounts of network communication required to maintain consistency between large numbers of participants.

Maintaining consistency mechanisms, whether conservatively or optimistically, requires additional network communication and has noticeable effects on latency and bandwidth consumption. Therefore many systems employ additional methods to balance networking requirements with consistency. These methods include active replication and dead reckoning [66].

3.3.2 Scalability

Scalability determines how easily a NVE can be expanded. There are two common measures of scalability [16]. First, scalability can be measured by the number of separately modeled entities that are simultaneously participating in the networked system. Alternatively, scalability can be a measure of the number of hosts that may be connected at any given moment. For many applications, such as military battlefield simulation, one of the most distinguishing characteristics is the incorporation of large numbers of participants. Therefore, for the purposes of the taxonomy, scalability is defined by the number of hosts connected simultaneously.

Historically, large scale has been very difficult to achieve for applications using networked video, including video conferencing, because of the large amounts of bandwidth consumed to stream video to each host. However, large scale becomes easier to achieve when only the changes to the environment need to be transmitted to each host [54]. Therefore, large scale NVEs, whose priority is to connect large numbers of users, do not typically send video updates to each host.

Three types of scalability have been presented [68]: upward scalability, sideways scalability, and downward scalability. Upwards scalability refers to the number of objects and users that the system can support, similar to Singhal and Zyda's traditional definition for determining large and small scale environments [16]. Sideways scalability enables the dynamic grouping of these participants into crowds or smaller awareness or interaction groups. In order to group participants, there must first be support for increasing numbers of participants; therefore upwards scalability must precede sideways scalability. Downward scalability deals with parts of the NVE being connected to systems with large computational power, necessary for example to support the rapid transmission of massive data sets required by tele-immersive applications [46].

Another issue in scaling the environment deals with the compatibility of hardware and software devices and applications. An application that is scalable in terms of hardware is extensible, supporting multiple interaction paradigms for various types of hardware. For example, a user in a multi-walled projection room may interact with the environments through trackers on his head and hands, while a desktop user is limited to a mouse, but has easier access to a keyboard. Supporting heterogeneous software applications and systems makes an application portable. It aids in scalability, as users at various sites may have access to workstations with differing operating systems, such as from Linux or Unix to Windows.

3.3.3 Persistence

In a virtual environment, persistence can offer the user an enhanced sense of presence, by providing a place with which the user may gain familiarity and revisit. In addition, if there is work to be done, persistence of the environment allows the user's progress to endure for later modifications on another visit. Yet, persistence becomes even more important in a multi-user scenario. When participants communicate and interact in a common environment, a place where they can part and meet again, they develop a feeling of copresence. Whether at any one time they work alone or together, labors from their collaborative efforts will not be erased upon exit, but will persist. The persistence of an environment could also lead to forms of asynchronous communication such as that proposed by the virtual mail system [42], where work can be saved and transferred to other sites for future modification.

3.4 A taxonomy of NVEs

These additional requirements of NVEs have been incorporated into the taxonomy, resulting in Fig. 2. The iterative nature of the development of this taxonomy is not fully detailed here, but may be found in [69]. Further, this proposed taxonomy has been evaluated with respect to perceptual orthogonality, completeness, and parallel structure [69]. This evaluation is as subjective and arbitrary as taxonomies themselves. It is argued that this taxonomy's best evaluation might be done through application or the use of the taxonomy. Therefore, this taxonomy is used to classify NVE systems and applications in the following sections. Further, this taxonomy is used to guide the design of an idealized NVE for mechanical engineering design.

3.4.1 The orthogonality test

These additional requirements of NVEs have been incorporated into the taxonomy, resulting in Fig. 2. The iterative nature of the development of this taxonomy is not fully detailed here, but may be found in [69]. Further, this proposed taxonomy has been evaluated with respect to perceptual orthogonality, completeness, and parallel structure [69]. This evaluation is

Autonomy	Collision detection			
	Gravity			
	Kinematics			
	Behaviors			
Interaction	Environmental Interaction	Responsiveness	Bandwidth	
			Latency	Computation
				Network
			Reliability	
			Consistency (b/w users)	
		Range		
		Mapping	Natural	
			Arbitrary	
		Scalability	Upwards	
			Sideways	
			Downwards	
			Hardware	
			Software	
		Remote Manipulation		
		Persistence		
	Social Interaction	Gaze direction		
		Facial Expression		
		Body Gestures		
Presence	Vividness	Representation	Visual	
			Auditory	
			Kinesthetic	
		Resolution		
	Consistency	Across all senses		
	Personal Perspective	First		
		Second		
		Third		

Fig. 2 A taxonomy for NVEs

as subjective and arbitrary as taxonomies themselves. It is argued that this taxonomy's best evaluation might be done through application or the use of the taxonomy. Therefore, this taxonomy is used to classify NVE systems and applications in the following sections. Further, this taxonomy is used to guide the design of an idealized NVE for mechanical engineering design.

4 Evaluation of NVE systems and applications

NVE systems provide a framework for building applications, through sets of libraries and tools for communication, visualization, and interaction, which reduce some of the programming effort required for the development of NVE applications. NVE systems include platforms, frameworks, and toolkits, which provide modules and libraries to mitigate the prototyping of new applications. They hide the complexities of detailed application development behind more easily understood interfaces and abstractions, leading to more rapid application development [70]. Applications are implementations of an NVE and are not especially designed for adaptation to new applications. Section 0 describes example NVE systems in terms of the taxonomy. Section 0 lists sample NVE applications considered. Finally, the evaluation of the taxonomy is presented in Section 0.

4.1 NVE systems

The DIVE system for the development of multi-user distributed applications was designed to scale to large numbers, while maintaining a maximum level of interaction at each participating site. [41,66,71,72]. DIVE environments are divided into virtual worlds, which are then organized into sub-hierarchies. Sections of the hierarchy are replicated and accessed when processes express an interest in them. Using partial, active replication, DIVE achieves low latency at local interaction. At the same time, consistency is enhanced with dead-reckoning techniques. The active replication model used in DIVE has proven its strength for applications with high amounts of user interaction, yet it has faired less favorably for applications with large amounts of data, because of the high network load. Unreliable multicast is used for streaming data (video and audio) while reliable multicast is used for making state changes to the environment entities. DIVE has a peer-to-peer architecture for communication among connected sites, but uses a name server for connecting new sites. DIVE users are represented as "body icons", with separated heads that are mapped to a user's movement in an HMD. Gaze direction is supported with two eyes on the head of the body icon representing the viewpoint from which the user sees the virtual world. Video and audio communication between participants is supported.

Type	Name	Description	Lab
CVE (general)	DDRIVE	Distributed Design Review in Virtual Environment	HRL Laboratories
	VR Spatial	Design and simulation of 4C spatial mechanisms	VRAC (Iowa State University)
	Virtual Mail	Synchronous and asynchronous system including the delivery of virtual mail messages	EVL (University of Illinois)
Virtual Conferencing	DIVE Rooms	Mimics rooms forming a virtual office	Swedish Institute of CS
	VIVID	Video stream textured onto the embodiment's face at each workstation	University of Nottingham
Tele-Operation	EVIPRO	Enables cooperation between multiple users and an autonomous robot	University of Toulouse
	MAESTRO	Distributed augmented reality system for training on installation and maintenance	University of Pisa
Tele-Immersive	CIBRVIEW	Collaborative Image Based Rendering Viewer	EVL (University of Illinois)
	TIDE	Tele-Immersive Data Exploration Environment	EVL (University of Illinois)
Large Scale	NPSNET	Military battlefield simulation environment	Naval Postgraduate School

Fig. 3 NVE applications considered

Distributed Open Inventor (DIV) [73] was developed to distribute applications developed with the Open Inventor [74] toolkit over a network. Open Inventor is a graphics toolkit, which reduces development effort for authoring graphical applications. The goal of DIV was to take the Open Inventor toolkit and extends it to accommodate the development of distributed graphical applications. The DIV architecture uses a client-server configuration, with partial replication and multicasting. DIV objects, which are stored in memory, are organized into a scene graph, or hierarchical data-structure representation of the scene. The server maintains a master copy of the scene graph and sends updates to all clients using multicasting. Reliability is ensured with the aid of a sequencing and retransmission algorithm. DIV makes no mention of having audio or video streaming capabilities.

VRJuggler is a virtual platform for the development of VR applications [70,75,76]. Its components for dynamic configuration and abstraction of input and output devices facilitate the implementation of heterogeneous systems. VRJuggler is so named from its ability to "juggle" any device, graphics API, and platform configuration. In addition to hardware extensibility, VRJuggler is portable to multiple platforms, including SGI, HP, and Windows. The VRJuggler network manager is designed to assist in the distribution of tasks to multiple processors in order to support downwards scalability. Although VRJuggler also contains modules for monitoring performance, it is not yet a stand-alone framework for the development of NVE applications. It can currently be run with other component frameworks, such as CAVERNsoft [18] and Bamboo [52], for distribution across a network, and its own distribution capability is imminent.

CAVERNsoft [46] is a toolkit for constructing NVEs focused on tele-immersive collaboration. In order to support these high-performance applications, which require the transfer of massive amounts of data between distributed sites, CAVERNsoft specializes in incorporating the use of supercomputing resources among distributed sites on high-speed networks. CAVERNsoft was built upon a foundation of experience combining client-server based networking using CAVEs and persistence in CALVIN (Collaborative Architectural Layout Via Immersive Navigation) [40] and the introduction of a message passing system (peer-to-peer) in NICE (Narrative Immersive Collaborative/Constructionist Environment) [77, 78]. As a result, CAVERNsoft is designed to support both client-server and peer-to-peer models with a choice of networking protocols while providing both low-level and high-level abstract access to networking tasks.

4.2 NVE applications

The previous section has given an overview of NVE systems and described them in terms of the taxonomy. This section will just list the NVE applications considered in our study (Fig. 3). A more comprehensive review of these applications is found in [69].

4.3 Comparisons

The research presents a taxonomy for describing NVEs. The taxonomy is constructed to give a full view of all characteristic components of a NVE system, including autonomy of objects, interaction and presence. Next the taxonomy is put to the test, by using it to describe current applications. This section discusses how well the taxonomy performs. The systems and applications described in the previous section are condensed into the taxonomy for comparison in Fig. 4 (autonomy), Fig. 5 (interaction), and Fig. 6 (presence). Each of the applications' characteristics are marked in the tables as they are described in the literature, however in some cases,

		DIVE	DIV	VRJuggler	CAVERNsoft		CIBRView	TIDE	VIVID	DIVEroom	DDRIVE	VRSpatial	Virtual Mail	EVIPRO	MAESTRO	NPSNET-IV
Autonomy	Collision detection						-	-				+		+		+
	Gravity						-	-								
	Kinematics						-	-				+				+
	Behaviors						+	+				+		+		+

yes and no options are indicated by '+' and '-' respectively, a choice between other options is indicated with an 'x'

Fig. 4 Evaluation of applications (autonomy)

					DIVE	DIV	VRJuggler	CAVERNsoft		CIBRView	TIDE	VIVID	DIVEroom	DDRIVE	VRSpatial	Virtual Mail	EVIPRO	MAESTRO	NPSNET-IV
Interaction	Environmental	Responsiveness	Band-width	high								x							
				medium							x			x					
				low															x
			Latency	high							x					x			
				low	x														
			Reli-ability	high															
				low															
			Con-sistency	high															
				low															
		Range		high															
				low															
		Mapping		Natural									x						
				Arbitrary							x								
		Scalability	Upwards	small							x	x		x	x	x	x	x	
				large	x														x
			Sideways								-	+		+			-		+
			Downwards				+	+		+	+								
			Hardware				+			+	+	-		+	-	-			
			Software				+					-		+	-	-	+	-	-
		Remote Manipulation			-	-	-	-		-	-	-	-	-	-	-	+	+	-
		Persistence						+			+		+		+	+	-	+	
	Social	Gaze direction			+					-		+		-		+			
		Facial Expression								-	+	+	-	-		-			-
		Gestures								-	+	-			+	+			

yes and no options are indicated by '+' and '-' respectively, a choice between other options is indicated with an 'x'

Fig. 5 Evaluation of applications (interaction)

where information is not clearly described but implied, a few assumptions are made.

The following assumption is made concerning the autonomy characteristics of the applications investigated:

– Teleimmersive applications do not have collision detection, gravity, or kinematics: The teleimmersive applications CIBRView and TIDE deal solely with visualized data, not with objects or other artifacts that exist in the real world. Therefore, we conclude that there might not be a need for implementing collision detection, gravity, or kinematics functions to portray real world object behavior.

The following assumptions are made concerning the interaction characteristics of the applications investigated:
Sideways scalability: Applications which support a maximum of two participants do not have sideways scalability as a group of two cannot be split into smaller groups.
Video implies facial expression: The use of video for the representation of participants implies the portrayal of facial expression.
No video implies no facial expression: Facial expression in a virtual environment can be supported either through the use of video or tracking points on the face and mapping movements a virtual representation. The first implementation is

					DIVE	DIV	VRJuggler	CAVERNsoft	CIBRView	TIDE	VIVID	DIVEroom	DDRIVE	VRSpatial	Virtual Mail	EVIPRO	MAESTRO	NPSNET-IV
Presence	Vividness	Representation	Visual	virtual only					**x**			**x**	x	**x**	**x**	**x**		x
				virtual in real						x	x							
				real in virtual														
				no														
			Auditory						+	+	+	+	+		+	+		-
			Kinesthetic						-	-	-		-	-	-	+	-	-
		Resolution		high														
				low														
	Consistency across all senses																	
	Personal Perspective		First							x		x		x	x			x
			Second						x				x					
			Third											x				

yes and no options are indicated by '+' and '-' respectively, a choice between other options is indicated with an 'x'

Fig. 6 Evaluation of applications (presence)

most often used, and has been used in a number of the NVE applications described here. The second approach necessitates serious development effort, which would likely be reported if it were implemented. Therefore, unless otherwise reported, it is assumed that applications that do not incorporate the use of video into user embodiment do not portray the participants' facial expression.

Differentiating high, medium, and low bandwidth: High bandwidth includes Ethernet (>10 Mbps). Medium bandwidth includes T-1 connections (1–10 Mbps). Low bandwidth includes modems across phone lines (<1 Mbps). In cases with heterogeneous networks (multiple bandwidths), the lowest bandwidth of the network determines system bandwidth.

High and low reliability: Highly reliable applications make use of either reliable multicast or unicast protocols. Low reliability applications do not employ reliable data transfer methods. This differentiation permits only a abstract discussion of the reliability of NVEs. Reliability is difficult to compare between two published applications because reliability details are sparsely reported.

Remote manipulation: Integrating remote manipulation into a virtual environment requires significant effort. Therefore, applications are assumed to not have remote manipulation unless specifically reported.

The following assumptions are made concerning the presence characteristics of the applications investigated:

No report on video implies virtual only: If an application reports the use of visual representation of the environment, but makes no distinction on the use of video, the application is assumed to have a virtual only visual representation.

No report on haptic implies no haptic: As force feedback would change the nature of the interaction with the environment, it is too defining a quality to omit from the publication if it was implemented.

5 An NVE for mechanical engineering design

The goal here is to use the derived taxonomy in designing a NVE to support collaborative engineering design. Previous work at Clemson University has resulted in the development of the single user modeling application, Virtual LEGO, which provided 3D blocks and simple CAD operations for the construction of 3D models [79]. Anticipating its future extension to a distributed multi-user application, realistic cues for social interaction were implemented. The single user environment allows the users to see their own hands constructing models in the environment, and in future collaborative scenarios, they would be able to see each other's hands interacting. The hand embodiment is accomplished with the integration of a video camera and projection-based display with the blue-box technique. The blue-box technique uses sequences of blue-flashes on the projection-based display in order to extract the user's hands from the real environment and incorporate it into the virtual world [80]. The next phase in the development of this application is the extension to a networked environment with multiple participants.

With its simple primitives and basic CAD functionality, the networked VirtualLEGO environment provides a collaborative space for modeling in the early stages of design. Product design review is a collaborative design task occurring throughout the product realization process, with many meetings in later stages focused on decision-making concerning the final details of the product. These meetings, populated with multiple stakeholders in the product development such as design, manufacturing, analysis, quality, maintenance departments and suppliers, have different requirements than those in the early stages, as the design has already been specified and detailed to a great extent. Tools supporting collaborative design review must be able to support social interaction and the visualization of complex models. Software and hardware scalability are essential, as diversity of interfaces and systems may increase with multiple involved organizations. Further, in mechanical engineering detailed stage review, collision detection is important, as the simple geometries used in the early stages are replaced by complex solid models in these later stages. Although a design review application should achieve high performance to support successful collaboration, some of the performance must be relinquished in order to handle interaction and visualization of large models. Latency may be impacted, as limits on bandwidth may cause lags and congestion in the transfer of huge amounts of data.

In later stages of the design process, visualization of complex objects for analysis among collaborators from different fields becomes more important. An environment supporting these requirements most closely resembles a teleimmersive application, as complex mechanical models require enormous amounts of memory and processing power to synchronize, construct, and view [81]. Therefore, complex virtual CAD systems must be downwardly scalable, and would probably benefit from the integration of a supercomputer near the center of a client-server model.

With these design requirements, an idealized mechanical engineering design review NVE application may be specified in terms of the proposed taxonomy. This provides developers with a good understanding of the specific functionality that must be incorporated into the system. Further, existing systems may be investigated to determine whether they might be extended to satisfy the desired requirements.

collaborative design review				
Autonomy			Collision detection	yes
			Gravity	yes
			Kinematics	yes
			Behaviors	yes
Interaction	Environmental Interaction	Respons-iveness	Bandwidth	high
			Latency	low
			Reliability	high
			Consistency	high
			Range	low
			Mapping	natural
		Scalability	Upwards	small
			Sideways	no
			Downwards	yes
			Hardware	yes
			Software	yes
			Remote manipulation	no
			Persistence	yes
	Social Interact.		Gaze direction	no
			Facial Expression	no
			Gestures	yes
Presence	Vividness	Repre-sentation	Visual	real in virtual
			Auditory	yes
			Kinesthetic	no
			Resolution	high
			Consistency across all senses	yes
			Personal Perspective	first

Fig. 7 Taxonomy specifications for a collaborative design review environment

6 Use and critique of the taxonomy through the development of an NVE for design review

The taxonomy specifications mentioned in Fig. 7 can be further applied to compare academic systems dedicated to design and/or more specifically for design review, some of which have already been compared in Figs. 4, 5, and 6. Analyzing and comparing these applications help to identify the gaps existing in the current technology therefore emphasizing the needs for future research. In addition, these specifications can be considered as being requirements for the development of an NVE for design reviews.

As mentioned previously, Virtual LEGO is a developed application that offers simple modeling capabilities. This work has been the foundation for further development of a collaborative environment dedicated to design review. In this perspective, the taxonomy has been used to identify tools to fill in the gaps that were identified as shown in Fig. 8. VRJuggler provides the libraries necessary to handle the different VR hardware. CAVERNsoft handles the networking functions necessary for building the collaboration. V-Collide provides the collision detection libraries. Finally, OpenGL Performer is used as a graphic engine. These different libraries provide the means necessary to build an application dedicated to modeling and design that encompasses networking collaboration capabilities while being portable to different operating systems such as Windows and Linux. While the application is still under development, the current results show that the application can be run on both platforms as a client as well as a server [82]. The taxonomy helps identify the need for research in the field of collaborative environments

collaborative design review					
Autonomy			Collision detection	yes	V-Collide
			Gravity	yes	-
			Kinematics	yes	-
			Behaviors	yes	-
Interaction	Environmental Interaction	Respons-iveness	Bandwidth	high	CAVERNsoft
			Latency	low	-
			Reliability	high	CAVERNsoft
			Consistency	high	-
		Range		low	-
		Mapping		natural	-
		Scalability	Upwards	small	-
			Sideways	no	-
			Downwards	yes	-
			Hardware	yes	VRJuggler
			Software	yes	All libraries
		Remote manipulation		no	-
		Persistence		yes	-
	Social Interact.	Gaze direction		no	-
		Facial Expression		no	-
		Gestures		yes	-
Presence	Vividness	Repre-sentation	Visual	real in virtual	Open GL Performer
			Auditory	yes	-
			Kinesthetic	no	-
		Resolution		high	-
	Consistency across all senses			yes	-
	Personal Perspective			first	VRJuggler

Fig. 8 Extensions to virtual LEGO with supporting technologies

dedicated to modeling/design review. As mentioned in Section 0, the nature of the taxonomy is arbitrary and subjective. Using the taxonomy to analyze existing systems and then to develop an application does highlight areas of the classification that either need to be clarified, moved, or removed.

For instance, in the Interaction → Environmental → Responsiveness category, the criteria bandwidth and latency can be interpreted as depending on each other since for any given-size packet, the higher the bandwidth available, the lower the latency. However, in real-time applications, latency only concerns the delay between the user's input and its effect in the virtual environment. Therefore, in this classification, the term latency can be specified as being solely computation latency. Further, in the perspective of modifying the taxonomy, the criterion mapping can be seen as restricting. Indeed, mapping of an object or any texture in a computed environment is achieved through position matrices, or vertices therefore a true mapping in a virtual environment can not realistically be natural. However, the definition and the precision of the mapping can be reduced to a point where a user would not feel restricted by the positioning if the detailing of the environment is fine enough. Thus, while this taxonomy supports analysis and comparison of existing collaborative environments dedicated to design review, it is still a living classification and should be re-examined as new technologies are introduced.

7 Conclusions

The emerging field of NVEs is large, diverse, and lacking in classification. Therefore, this research develops a taxonomy to facilitate the research and design of NVEs. The taxonomy is derived from definitions of VR and networked VR found in the literature. These characterizations are organized with overlap controlled by tests for perceptual orthogonality. The functionality of the taxonomy is demonstrated by comparing existing NVEs.

The taxonomy serves multiple purposes for the research and design of NVEs. First, it brings order and clarity to a large domain of information so that applications can be more easily compared and contrasted. Second, analysis on the use of the taxonomy determines that taxons could be categorized as distinguishing, optimizing, or quality characteristics. These characterizations determine the uniqueness of types of NVEs, their performance in use, and the usability of their implementation. It is suggested that published descriptions of NVEs could use the taxonomy to more clearly describe the capabilities. Finally, the taxonomy provides a starting point for the design of a NVE for collaborative design review. The taxonomy establishes a list of characteristics to be addressed, reducing the risk of not addressing key development issues.

The taxonomy developed in this research classifies NVEs in an organized manner, but it does not take credit for being the only solution to classification. As research in the field progresses, particularly in the future development of NVE projects, the taxonomy should be refined and expanded as new issues are discovered. It is not meant to be a final complete tool, but instead a living one, which will evolve with the field as it matures.

References

1. Sastry, L., Boyd, D.: Interactions in virtual environments. Eur Res Consortium Inform Math **41**(April) (2000)
2. Coomans, M., Timmermans, H.: Towards a taxonomy of virtual reality user interfaces. In: Proceedings of the International Conference on Information Visualization (IV97), London, UK, August 1997
3. Smith, P., Marsh, S., Wright, T., Duke, D.: A shared framework of virtual reality. In: 5th UK-VRSIG, Exeter, UK (1998)
4. de Candolle, A.: Theorie Elementaire de la Botanique. Paris, (1813)
5. Jeffrey, C.: An Introduction to Plant Taxonomy. Cambridge University Press, Cambridge (1982)
6. Derr, R.: A taxonomy of social purposes of public schools (1973)
7. Gershenson, J., Stauffer, L.: Taxonomy for design requirements from corporate customers. Res. Eng. Des. **11**, 103–105 (1999)

8. Slaughterbeck-Hyde, R.: A measure of design quality. Master's thesis, University of Idaho (1989)
9. Dunn, G., Everitt, B.: An Introduction to Mathematical Taxonomy. Cambridge University Press, New York (1982)
10. Saar, K.: Virtus. In: Symposium on the Virtual Reality Modeling Language (VRML '99). Paderbourn, Germany (1999)
11. Dixon, J., Duffey, M., Irani, R., Meunier, K., Orelup, M.: A proposed taxonomy of mechanical design problems. In: ASME Computers in Engineering Conference (CIE'88), San Fransisco, CA. ASME (1988)
12. Huang, M., Alessi, N.: Current limitations into the application of virtual reality to mental health research. Virt. Environ. Clin. Psychol. Neurosci. (1998)
13. Bowman, D., Gracey, M., Lucas, J.: Efficient, intuitive user interfaces for classroom-based immersive virtual environments. In: IEEE Virtual Reality 2004, Chicago, IL, 27–31 March 2004, pp. 219–220
14. Pike, K.: Commercial virtual reality applications enter military. National DEFENSE, Febuary 1996
15. Summers, J., Butler, A., Kuo, E.: Development of a feature based design system using virtual reality. In: ASME DETC-1999, Las Vegas, NV. DETC99/CIE-9034 (1999)
16. Singhal, S., Zyda, M.: Networked virtual environments: design and implementation (1999)
17. Frecon, E., Noeu, A.: Building distributed virtual environments to support collaborative work. In: ACM Symposium on Virtual Reality Software and Technology (VRST'98), Taipei, Taiwan (1998)
18. Capin, T., Magnenat-Thalmann, N., Thalmann, D., Pandzic, I.: Toward natural communication in networked collaborative virtual environments. Framework for Immersive Working Environments International (1996)
19. Ford-Morie, J.: Cgi taxonomy project (2001). www.siggraph.org/education/curriculum/projects/Taxonomy2001.ht
20. Pierra, G.: Towards a taxonomy for interactive graphics systems. In: Palanque, P., Bastide, R. (eds.) Proceedings of Design, Specification, Validation of Interactive Systems. Springer Computer Science Series, pp. 362–370. Springer, Heidelberg (1995)
21. Bakker, A., Kuz, I., Van Steen, M.: Towards a taxonomy of distributed-object models. In: Proceedings of the Third Annual ASCI Conference, Heijen, The Netherlands, June 1997, pp. 22–27
22. Macedonia, M., Zyda, M.: A taxonomy for networked virtual environments. IEEE Multimedia **4**(1), 48–56 (1997)
23. Greenbaum, P.: The lawnmower man. Film and Video **9**(3), 58–62 (1992)
24. Roberts, D., Warwick, K.: An overview of virtual reality. In: Warwick, K., Gray, J., Roberts, D. (eds.) Virtual Reality in Engineering, pp. 1–24 (1993)
25. Furness, T., Barfield, W.: Introduction to virtual environments and advanced interface design. Virt. Environ. Adv. Interf. Des. (1995)
26. Barfield, W., Zeltzer, D., Sheridan, T., Slater, M. (1995) Presence and performance within virtual environments. Virt. Environ. Adv. Interf. Des., pp. 473–513 (1995)
27. Steuer J.: Defining virtual reality: dimensions determining telepresence. J. Commun. **42**(4), 72–93 (1992)
28. Bryson, S.: Approaches to the successful design of virtual environments. Virt. Real. Appl., pp. 3–15 (1995)
29. Slater, M., Usoh, M., Steed, A.: Depth of presence in virtual environments. Presence: Teleoper. Virt. Environ. **3**(2), 130–144 (1994)
30. Kim, G., Seo, J.: Design for presence: a structured approach to virtual reality system design. Presence: Teleoper. Virt. Environ. **11**(4), 378–403 (2002)
31. Heeter C.: The subjective experience of presence. Presence: Teleoper. Virt. Environ. **1**(2), 262–271 (1992)
32. Slater, M., Usoh, M.: Presence in immersive virtual environments. In: IEEE Virtual Reality Annual International Symposium (VRAIS'93) (1993)
33. Slater, M., Usoh, M.: Representation systems, perceptual positions, and presence in immersive virtual environments. Presence: Teleoper. Virt. Environ. **2**, 221–233 (1993)
34. Lombard, M., Ditton, T.: At the heart of it all: the concept of presence. J. Comput. Mediat. Commun. **3**(2) (1997)
35. Kalawsky, R.: The Science of Virtual Reality and Virtual Environments. Wokingham, England
36. Zeltzer, D.: Autonomy, interaction, and presence. Presence: Teleoper. Virt. Environ. **1**(1), 127–132 (1992)
37. Ohya, J., Kitamura, Y., Takemura, H., Kishino, F., Terashima, N. (1993) Real time reproduction of 3d human images in virtual space teleconferencing. Teleconferencing, VR 408–414
38. Kindratenko, V., Kirsch, B.: Sharing virtual environments over a transatlantic atm network in support of distant collaboration in vehicle design. In: Virtual Environments Conference, Eurographics Workshop Proceedings Series (1998)
39. Daily, M., Howard, M. et al.: Distributed design review in virtual environments. In: Collaborative Virtual Environments (CVE'00), San Francisco, CA. ACM, New York (2000)
40. Leigh, J., Johnson, A.E., Vasilakis, C.A., DeFanti, T.A.: Multi-perspective collaborative design in persistent networked virtual environments. In: IEEE Virtual Reality Annual International Symposium (VRAIS'96) (1996)
41. Frecon E. Stenius M.: Dive: a scalable network architecture for distributed virtual environments. Distributed Syst. Eng. J. **5**(3), 91–100 (1998)
42. Imai, T., Johnson, A.E., Leigh, J., Pape, D., DeFanti, T.: Supporting transoceanic collaborations in virtual environment. In: 5th Asia-Pacific Conference on Communications and 4th Optoelectronics Communications Conference (APCC/OECC), Beijing, China, pp. 1059–1062 (1999)
43. Yoon, W., Tsumaki, Y., Uchiyama, M.: An experimental system for dual-arm robot teleoperation in space with concepts of virtual grip and ball. In: International Conference on Advanced Robotics (ICAR'99), Tokyo, Japan
44. Sonnenwald, R., Bergquist, D., et al.: Designing to support collaborative scientific research across distances: the nanomanipulator environment. In: Munro, A., Churchill, E., Snowdon, D. (eds.) Collaborative Virtual Environments: Digital Places and Spaces for Interaction, pp. 202–224. Springer, London (2001)
45. Leigh, J., Johnson, A.E., Brown, M., Sandin, D.J., DeFanti, T.A.: Visualization in teleimmersive environments. IEEE Comput. **32**, 66–73 (1999)
46. Park, K., Cho, Y., et al.: Cavernsoft g2: A toolkit for high performance tele-immersive collaboration. In: ACM Symposium on Virtual Reality Software and Technology (VRST'00), Soeul, Korea (2000)
47. Calvin, J., Dickens, A., et al.: The simnet virtual world architecture. In: IEEE Virtual Reality Annual International Symposium (VRAIS'93). IEEE Press, Seattle (1993)
48. Macedonia, M.: Mbone provides audio and video across the internet. IEEE Comput. **27**, 30–36 (1994)
49. Snowdon, D., West, A.: Aviary: Design issues for future large scale virtual environments. Presence: Teleoper. Virt. Environ. **3**(4), 288–308 (1994)
50. Miller, D., Thorpe, J.: Simnet: The advent of simulator networking. IEEE (1995)
51. Schwartz, P., Bricker, L., et al.: Virtual playground: Architectures for a shared virtual world. In: ACM Symposium on Virtual Reality Software and Technology (VRST'98), Taipei, Taiwan (1998)
52. Watson, K., Zyda, M.: Bamboo: A portable system for dynamically extensible, real-time, networked virtual environments. In: IEEE

Virtual Reality Annual International Symposium (VRAIS'98), Atlanta, GA (1998)
53. Brand, S.: The medialab: inventing the future at mit, viking penguin (1987)
54. Reynard, G., Benford, S.: Vivid: A symbiosis between virtual reality and video conferencing. In: UKERNA Video Conferencing Workshop, Nottingham University (1996)
55. Kuijpers, N., Jense, H.: Collaborative engineering in distributed virtual environments. In: Spring Simulation Interoperability Workshop, Orlando, FL (1997)
56. Broll, W.: Interacting in distributed collaborative virtual environments. In: IEEE Virtual Reality Annual International Symposium (VRAIS'95), Research Triangle Park, NC. IEEE Computer Society Press (1995)
57. Grigore, B.: Force and touch feedback for virtual reality. John Wiley and Sons, (1996)
58. Arai, F., Tanimoto, M., and al.: Distributed virtual environment for intravascular tele-surgery using multimedia communication. IEEE Virtual Reality Annual International Symposium (VRAIS'96) (1996)
59. Leigh, J., Johnson, A., DeFant, T., Bailey, S., Grossman, R.: A methodology for supporting collaborative exploratory analysis of massive data sets in tele-immersive environments. In: High Performance Distributed Computing. IEEE Press, Redundo Beach (1999)
60. Vaghi, I., Greenhalgh, C., et al.: Coping with inconsistency due to network delays in collaborative virtual environments. In: Virtual Reality Science and Technology. London, UK (1999)
61. Lamport, L.: How to make a multiprocessor computer that correctly executes multiprocessor programs. IEEE Trans. Comput. **C-28**(9), 690–691 (1979)
62. Goodman, J.: Cache consistency and sequential consistency. SCI Committee (1989)
63. Dubois, M., Wang, J., et al.: Delayed consistency and its effects on the miss rate of parallel programs. In: IEEE Supercomputing Conference (SC'91) (1991)
64. Gharachorloo, K., Lenoski, D., et al.: Memory consistency and event ordering in scalable shared-memory multiprocessors. In: International Symposium of Computer Architectures (ISCA'90) (1990)
65. Greenberg, S., Marwood, D.: Real-time groupware as a distributed system: Concurrency control and its effect on the interface. In: ACM Conference on Computer Supported Cooperative Work (CSCW'94) (1994)
66. Carlsson, C., Hagsand, O.: Dive-a platform for multi-user virtual environments. Comput. Graph. **17**, 663–669 (1993)
67. Birman, K., Schiper, A., Stephenson, P.: Lightweight causal and atomic group multicast. ACM Trans. Comput. Syst. **9**(3), 272–314 (1991)
68. Robinson, M., Pekkola, S., et al.: Extending the limits of collaborative virtual environments. Collaborative Virtual Environments: Digital Places and Spaces for Interaction, pp. 21–42 (2001)
69. Baladi, M.: A taxonomy for the design and evaluation of networked virtual environments. Master's thesis, Mechanical Engineering Department, Clemson University, Clemson, SC (2003)
70. Just, C., Bierbaum, A., Baker, A., Cruz-Neira, C.: Vr juggler: A framework for virtual reality development. In: 2nd Immersive Projection Technology Workshop (IPT'98), Ames, IA (1998)
71. Hagsand O.: Interactive multi-user ves in the dive system. IEEE Multimedia **3**, 30–39 (1996)
72. Norman V., Babski C., et al.: The coven project: Exploring applicative, technical, and usage dimensions of collaborative virtual environments. Presence: Teleoper. Virt. Environ. **8**(2), 218–236 (1999)
73. Hesina, G., Schmalstieg, D., Reitmayr, G.: Distributed open inventor: a practical approach to distributed 3d graphics. In: ACM Symposium on Virtual Reality Software and Technology (VRST'99), University College, London (1999)
74. Strauss, P., Carey, R.: An object-oriented 3d graphics toolkit. In: Proceedings of SIGGRAPH Conference on Computer Graphics (1992)
75. Bierbaum, A.: Vr juggler: A virtual platform for virtual reality application development. In: Computer Engineering. AMES, Iowa State University (2000)
76. Bierbaum, A., Just, C.: Software tools for application development. In: ACM SIGGRAPH, Orlando, FL, pp. 3.1–3.45 (1998)
77. Johnson, A., Roussos, M., Leigh, J., Barnes, C., Vasilakis, C., Moher, T.: The nice project: learning together in a virtual world. In: IEEE Virtual Reality Annual International Symposium (VRAIS'98), Atlanta, GA. IEEE Press, Seattle (1998)
78. Roussos, M., Johnson, A., Moher, T., Leigh, M., Vasilakis, C., Barnes, C.: Learning and building together in an immersive world. Presence: Teleoper. Virt. Environ. **8**(3), 247–263 (1999)
79. Klar, T.: An interactive virtual environment for 3d model and assembly with video immersion. Master's thesis, Department of Mechanical Engineering, Clemson University, Clemson, SC (2001)
80. Van Pernis, A.: Surface construction from within a virtual environment. Master's thesis, Department of Computer Science, Clemson University, Clemson, SC (1998)
81. Bidarra, R., van den Berg, E., Broonswort, W.F.: Collaborative modeling with features. In: ASME Design Engineering Technical Conferences (DETC'01), Pittsburg, PA. ASME (2001)
82. Vitali, H.: Framework for building a hardware and operating system independent collaborative virtual environment for design review. Master's thesis, Mechanical Engineering Department, Clemson University, Clemson, SC (2005)
83. DuPont, P.: Applied Virtual Reality. In: Warick, K., Gray, J., Roberts, D. (eds.) Virtual Reality in Engineering. The Institution of Electrical Engineers, Exeter, pp. 153–168 (1993)
84. Bershad, B., Zekauskas, M., Sawdon, W.: The Midway Distributed Shared System, IEEE International Computer Conference (COMPON '93) (1993)