Collocated Multi-Device Virtual Worlds

Bill Tomlinson, Man Lok Yau, Eric Baumer, Andrew Correa, Gang Ji, Joel Ross
Contact: wmt@uci.edu
UC Irvine

Abstract

Many human activities now take place in settings that include several computational devices – such as mobile phones, desktop computers, laptops, and GPS navigation systems – in the same physical space. However, there is a lack of interaction paradigms that support a coherent experience across these collocated technologies and enable them to work effectively as systems. One possible paradigm is the creation of virtual worlds that span across multiple collocated computing devices. This paper presents a framework for the creation of collocated multi-device virtual worlds, and describes two examples of systems implemented using this framework. Aspects of this framework include: communication between devices, cross-device graphics and sound, embodied mobile agents that inhabit the multi-device world, and real world integration. The effectiveness of this framework was evaluated by analyzing both the development process and the end products of the two implemented systems. To respond to several implementation challenges encountered during the development of these systems, this paper presents a prototype of a suite of software engineering tools to help develop and test future systems. The core contribution of this paper is a novel framework for collocated multi-device virtual worlds; by presenting this framework, this paper lays the groundwork for a wide range of potential applications.

1 Introduction

Over the past several decades, computational devices have spread rapidly among many human societies. Because of the rapid growth in the usage of these devices, they are often located in physical proximity to each other. However, while devices often have the capability to network with each other and with the Internet, these collocated devices rarely take full advantage of their physical proximity to each other to help them interact or to facilitate their interactions with people. The single-device interaction paradigms that humans use when engaging separately with these various heterogeneous devices rarely facilitate a coherent experience across several devices. As a result, human users miss out on the significant potential of collocated devices, and may feel overwhelmed rather than supported by the multiplicity of devices present. In order for these devices to integrate smoothly and for people to understand their collective operation, new interaction paradigms are needed.

Large collocated groups of people have a wide range of remarkable capabilities—as companies, as orchestras, as armies, as sports teams, as social clubs, as universities. While computational devices will not form these specific kinds of institutions, they may be able to work together to enable a similar kind of synergy. Just as people can work closely together when physically near each other, devices should be able to do so as well. The goal of the research presented here is to develop a framework for an interaction paradigm suited to collocated multi-device interactive systems, in order to allow people to benefit from having so many devices in the same physical space. By enabling the creation of this one form of multi-device HCI, we hope to lay the groundwork for a wide range of other forms of multi-device interaction.

To describe this framework, we begin with an assessment of existing research and commercial products that relate to the project. Thereafter we introduce two implemented prototypes of interactive exhibits that were built while developing this framework. We then present a model for human interaction with collocated multi-device virtual worlds, including an interaction metaphor that lends itself to these virtual worlds. Next, we present the core elements of the collocated device framework: collocated communication between devices, collocated graphics and virtual lighting, collocated sound, collocated interaction with people through embodied mobile agents, and real world sensing and integration. We then use the prototypes mentioned above to evaluate the viability of the framework as a whole, from the perspective of both end-users and developers. Finally, we present our ongoing and future work in the area of collocated devices, including a prototype for a multi-device software engineering toolkit that we developed to facilitate future efforts to build similar systems. Through this paper, we seek to demonstrate the viability of collocated multi-device systems in general, and specifically the usefulness of the framework and models presented here. Collocated multi-device systems are underutilized to date and have significant potential for improving the ways in which people interact with the devices that surround them by streamlining interactions across devices.

2 Related Work

This research draws on previous efforts to create an effective user experience for multi-device environments. The Pick-and-Drop system, for example, enables a user to transfer files by picking up files from one computer and dropping them into another computer with a pen (Rekimoto, 1997). It hides the underlying technical details from the users, simplifying the procedure of transferring files and data between multiple devices. In the framework described here, we sought to preserve the simplicity and elegance of Pick-and-Drop while enabling interactions with complex autonomous entities.

There have been several systems that involved multi-device interactions with agents. The AgentSalon project is a system with desktop computers and mobile devices to facilitate face-to-face communication (Sumi & Mase, 2001). A large display is shared with multiple users, and each user has a mobile device, such as a PDA, which holds an animated agent. The agents can be transferred to the large display, where they engage in automated conversation. These automated conversations are intended to facilitate conversations between the users. Agent Chameleons is a system with agents that can transfer between robots and virtual environment (O'Hare & Duffy, 2002). The project explores the embodiment of agents in robotic platforms as well as in virtual environments. Each platform has different associated behaviors and capabilities; the agent understands the platforms that it can migrate to and evaluates which one is more appropriate for the current situation. The agent will then migrate to that platform and continue to function. The PEACH system – Personal Experience with Active Cultural Heritage (Stock & Zancanaro, 2002) – augments museum exhibits near which the user is located with additional information, such as supplemental text and pictures, through the use of a PDA. The extra information can enhance the experience of the visitors. All of these projects represent the type of interaction that would be enabled by the spread of collocated multi-device systems. However, the research described in this article is different from these prior works in several ways: it presents a broader theoretical framework for this type of system, it has a significant focus on the heterogeneity of the component devices, and it highlights the moment of transfer between devices as a critical component for the realism and believability of multi-device virtual worlds.

Many different commercial products also involve close coordination of collocated devices. For example, mobile phones and Bluetooth headsets are designed to work closely together, as are video game consoles and wireless game controllers. The HotSync functionality that allows Palm devices to synchronize their data with a desktop computer is also example of two devices coordinating their action. However, most of these solutions are specifically tailored for connecting a small, fixed number of devices. The research described in this project seeks to understand the principles that underlie the development of systems that span larger groups of heterogeneous collocated devices.

Various factors influenced the choice of virtual worlds as the core interaction paradigm for this framework. Virtual worlds provide a number of benefits over other styles of interaction, such as windowing systems or text-based interfaces. They promote social engagement and activity (Barab, Thomas, Dodge, Carteaux, & Tuzun, 2005). They allow for an increased sense of embodiment and presence (Biocca, 1997). Finally, virtual worlds are able to augment and enhance understanding and education (Roussos et al., 1999). While virtual worlds are only one possible interaction paradigm for collocated multi-device systems, they provide a useful starting point for exploration of this growing area.

3 The Virtual Raft platform

Despite the innovative work already done towards developing multi-device systems, there does not yet exist a design framework to guide users or implementers in conceptualizing these systems. This paper offers such a framework, using the creation of a collocated Multi Device Virtual World (MDVW) as the core example. Two projects have been created using this framework: the Virtual Raft project and the EcoRaft project. Although the two projects have different applications, the technical details are very similar.

3.1 The Virtual Raft Project

The Virtual Raft project (Tomlinson, Yau, O'Connell, K. Williams, & Yamaoka, 2005) is a multi-device game that teaches color theory. It consists of three desktop computers and three tablet PCs. The desktop computers display “virtual islands” and the tablet PCs display “virtual rafts." Each island contains a central bonfire and several virtual characters that hold torches of the same color as the central bonfire. The bonfires are colored red, green, and blue—three additive primary colors. When a tablet PC is brought close to a desktop computer, a virtual character "jumps" from the virtual island to the virtual raft, moving from the desktop to the tablet PC. As the tablet PC is moved close to a different tablet desktop, the character jumps from the current raft to this new raft or island. When a user transfers a virtual character to new island, the torch color of the virtual character will mix with the central bonfire, changing its color. For example, a character with a red torch arriving at an island with a blue flame will create a violet fire. Mixing all three colors together creates a white flame. The goal of the game is to create white fires on all three islands, which requires each island to have at least one character from each of the other islands. Users interacting with the system can discover additive color mixing, such as how combining blue and red creates violet while combining red and green creates yellow.

3.2 The EcoRaft Project

The EcoRaft project (Tomlinson et al., 2006) is built using the same platform. It is a multi-device museum exhibit that helps children learn about restoration ecology. This project was developed in collaboration with an ecology professor and her students who study restoration of Costa Rican ecosystems. Like the Virtual Raft

Figure 1: Several children interact with a multi-device virtual world in the EcoRaft exhibit.

project, the EcoRaft project consists of three desktop computers and three tablet PCs. Each desktop computer contains a virtual ecosystem, modeled after real ecosystems in Costa Rica. The ecosystems are made up of Coral trees, Heliconia flowers, and different types of hummingbirds. One of the desktop computers represents a National Park, which always thrives with all of the species. The other two desktop computers represent more delicate ecosystems. These computers are each connected to a silver button, which when pressed removes all of the plants and animals from that island. Pressing the button represents the ability to over farm an island and devastate the ecosystem. Users can help restore the ecosystems by transferring plants and animal species from the National Park to the devastated islands. These species are carried using the tablet PCs, which represent virtual collection boxes and can be used to physically carry the virtual species between the virtual islands. Each box can only carry a single species, so users must work together to repopulate an island. This activity teaches users that the destruction of ecosystems is very easy, and restoration, while difficult, is still possible.

4 Multi-Device Systems

Most dominant interaction paradigms in HCI revolve around interacting with a single device at a time (Baecker, Grudin, Buxton, & Greenberg, 1995), (Card, Moran, & Newell, 1983), (Norman, 1988). Even when people are in groups, the emphasis with mobile devices is often on interacting with a single device, e.g., (Barkhuus et al., 2005). Many CSCW systems also incorporate group activity, e.g., (Grudin, 1988), (Grudin & Palen, 1995), (Heath, Luff, Lehn, Hindmarsh, & Cleverly, 2002), (Hindmarsh, Heath, Lehn, & Cleverly, 2005), but the interaction paradigm still revolves around users interacting with a single, isolated device, application, or system. This paper explores ways that the heterogeneity and complexity of multiple devices in the real world can be wrangled to create unified interactive experiences.

Both the Virtual Raft project and the EcoRaft project are examples of installations that attempt to take advantage of the unique interaction style afforded by multi-device systems. One way to conceptualize these multi-device systems is as *human-mediated networking*. Just as computer mediated communication (CMC) deals with the ways in which computers facilitate interactions between people, human mediated networking addresses ways in which people help coordinate interactions among devices. In this figure-ground inversion (a concept from art in which the viewer switches to perceive the background as the foreground and vice versa), the focus is on the computational devices, the interactions between those devices, and the role that humans play in those interactions. Consider the situation an individual encounters when moving a file between two devices via a network connection. From the human perspective, the user can sit at one machine, log into another machine, browse the file system, and transfer the required data. However, from the devices' perspectives, there is an entire complex of activities occurring, including the negotiation of network protocols, rapid exchange of data, and possible translations of file system architectures. In contrast, consider the process involved when moving data between two collocated devices that are not directly connected via a network or similar method. One common approach is using a USB drive to transfer the file, which involves inserting the drive in the first machine, mounting the drive, browsing the file system to locate the desired file, copying this file to the drive, unmounting the drive, physically disconnecting the drive from the first machine, physically connecting it to the second machine, mounting the drive, locating the desired data on the drive, copying it to the local file system, unmounting the drive, and physically disconnecting it from the second machine. During this process, the various devices involved (the two computers and the USB drive) carry out a series of file system tasks, the most involved being mounting an external device and copying data to or from it. However, the human user now serves as part of a file transfer protocol, one that involves not just digital negotiation between multiple devices but physical manipulation of devices, as well. Just as there is an important human role played in any cyberinfrastructure (C. P. Lee, Dourish, & Mark, 2006), it is also useful to consider the role played by human users in any multi-device networking protocol.

This is not an argument that humans and non-human devices be given a uniform, symmetric treatment. A number of arguments have been made in favor of treating humans and devices as homogeneous components of a distributed cognitive system socio-technical network (Law, 1986), (Latour, 1992), (Hutchins, 1996), as well as against it (Nardi, 1996), (Kaptelinin & Nardi, 2006). The approach being advocated here is that, by performing a figure-ground inversion on the usual computer-mediated communication paradigm, we are given a new analytic lens through which to consider the design of multi-device systems—one that focuses not only on the way that computational devices connect human users, but also on the ways that human users configure computational devices to deal with the heterogeneity (Bell & Dourish, 2007) and "seamfulness" (Chalmers, Dieberger, Höök, & Rudström, 2004) of multi-device systems.

4.1 The island metaphor

One way of framing the heterogeneous development and interaction inherent in multi-device virtual worlds is through the island metaphor (Tomlinson, Baumer, & Yau, 2006). In this metaphor, stationary devices are like islands of virtual space, separated from one another by seas of physical space. Mobile devices are like virtual rafts, allowing virtual entities to move between different islands of virtual space. The island metaphor is more appropriate than other interaction metaphors for multi-device virtual words for a number of reasons. First, by comparing the land/water distinction to the virtual-space/real-space distinction, the metaphor offers an intuitive model for how real space and virtual space relate to each other. Users can take advantage of previous notions of the process of traveling, such as having a specific method for getting into and out of a raft. People understand that they cannot simply climb into a raft when it is in the middle of the ocean—it must be physically close to do so, so the physical proximity requirements of the multi-device virtual world make sense.

Second, the island metaphor provides a clear role for autonomous software agents in computation—as inhabitants of a heterogeneous network of computational devices. The desktop computing metaphor does not lend itself to computational autonomy. Real desktops do not feature autonomous entities—paper documents do not sort themselves, pens and pencils do not automatically complete one's words as one is writing. The idea of autonomous systems on desktop computers is at odds with the metaphor on which the interactions with those computers are based, and therefore it is understandable if autonomous computational agents are viewed with ambivalence. Since people may not be accustomed to interacting with autonomous entities in the context of a real desktop, it could be harder for them to draw on their previous experience to know how to behave when confronted with autonomous agents in the context of a virtual desktop. Choosing an interaction metaphor for MDVWs that explicitly includes a place for computational autonomy and therefore lends itself to cross-enhancement with interactions with autonomous entities in the real world could be more effective than a metaphor where there is no clear connection.

The island metaphor also offers a more social and natural interaction paradigm than that created by the desktop metaphor. Multi-device systems lend themselves to social interaction—just as multiple devices work together to form a virtual world, multiple people may work together to experience and interact with that world. The island metaphor leads to interactions that involve multiple people—even if everyone carries his/her own virtual raft, they must interact at the same virtual islands, thereby requiring them to be in physical proximity to each other in order to transfer characters between islands. Through this process, systems based on the island metaphor may encourage a more social style of interaction with the virtual world, and possibly promote social interactions between users.

5 Framework

Creating multi-device virtual worlds involves several key elements. In this section, we describe the framework used for developing the Virtual Raft platform in greater detail. First we discuss networking and communication between collocated devices. We then discuss the framework's use of collocated graphics and virtual lighting, followed by the use of collocated sound. Finally, we discuss collocated interaction with people through the use of embodied mobile agents. Our goal in developing this framework is to allow for the easy creation of multi-device virtual worlds, while supporting a natural and intuitive interaction paradigm with these virtual worlds.

5.1 Multi-Device Networking

A collocated MDVW requires networking technologies to connect multiple devices. Communication among collocated devices factors into a number of other areas of research that have helped to inform the implementation of these projects. The Meme Tags project enabled the transportation and display of text fragments among small electronic badges (Borovoy et al., 1998), thereby creating a collaborative system using wearable devices. Brumitt et al. created technologies that support intelligent environments in which different devices communicate with each other to provide services for the users (Brumitt, Meyers, Krumm, Kern, & Shafer, 2000).

For example, pressure sensors and cameras are used to track users in a room, enabling the working session of that person to be transferred to the computer that is next to him or her. Tangible interfaces (e.g., (Ishii & Ullmer, 1997)) also often involve multiple collocated computational devices.

Some programs may require constant connections while others only need to transfer information at specific moments. There are many kinds of networking technologies that can be used in this process, and indeed some MDVWs may use more than one kind of networking protocol to achieve their requirements. In both the Virtual Raft and EcoRaft projects, the desktop computers are equipped with Ethernet connections and Infrared (IrDA) adapters. The mobile computers have built-in IrDA capabilities. The projects use IrDA for proximity detection because it operates over relatively short distances (compared to Bluetooth or wireless Ethernet) and can detect another device within approximately a 30-degree angle, which are suitable characteristics for the desktop computers to determine the proximity and orientation of the tablet PCs. However, the line of sight requirement and slow transfer speed of IrDA makes it difficult to transfer character information over that connection. In both projects, users frequently move around the space. It becomes difficult to have users hold up the tablet PC and maintain a line of sight connection while the virtual characters graphically transfer between computers. As such, we choose to only use IrDA to detect the proximity between the devices and to fetch the unique network name of the nearby device. The system then uses a WiFi connection to send the actual data for the virtual character. WiFi is faster and does not require line of sight, so the character can still transfer even if the user breaks the IrDA connection by moving the tablet PC. This combination of IrDA for proximity detection and WiFi for data transfer is well suited to the dynamic and mobile nature of the projects.

5.2 Multi-Device Graphics

Computer graphics, computational cinematography, and virtual lighting play a significant part in this framework for collocated systems. Previous research in several areas has informed the project on this topic. MR MOUT (Mixed Reality for Military Operations in Urban Terrain) focuses on the construction of algorithms that allowed for color transferring and shadowing between physical and virtual entities (Hughes, Reinhard, Konttinen, & Pattanaik, 2004). "Virtual Light" uses a virtual flashlight to emit an image that represents the same image a user would experience with a physical light source (Naemura, Nitta, Mimura, & Harashima). It focuses on the correlation between virtual and physical entities, and uses a virtual image to emulate a physical object under lighting. Previous work in virtual cinematography (e.g., (He, Cohen, & Salesin, 1996), (Tomlinson, Blumberg, & Nain, 2000), (Drucker, He, Cohen, Wong, & Gupta, 2003)) has suggested various ways to control the movement of the camera in virtual worlds. In addition, research in the growing area of lighting design in interactive environments (e.g., (El-Nasr, Zupko, & Miron, 2005)) has sought to create dynamic lighting design automatically in gaming environments to create a more engaging gaming experience.

Our framework uses multi-device computer graphics to create the illusion that a virtual character maintains its identity when transferring from one device to another. This helps to support the establishment of a MDVW, in that a single virtual entity appears to be moving between devices. To create this illusion, we use carefully coordinated timing, as well as graphical animations that support the idea of virtual characters who are actively mobile, rather than just passively transferred.

In the Virtual Raft and EcoRaft projects, when one device detects another using IrDA, they exchange connection information and determine if a character transfer should occur. Once a decision has been made to transfer, the inhabited computer sends data about the character to the targeted computer and initiates the transfer animation for the virtual character. For example, consider when a virtual character present in one of the tablet PCs will be transferred to a desktop computer. The tablet PC sends data about the character to the desktop computer, then begins the animation of the virtual character moving off the tablet. Once the desktop computer receives the information about the transferring character, it creates a graphical clone of the character based on that information—a virtual character who looks and acts exactly the same as the original. However, the new character is created off-screen at first, waiting for a specified time period before it begins its own transfer animation and appears on the desktop's display. This time delay insures that the character has already disappeared from the tablet PC, creating a fluid transfer and the illusion that the same virtual character was transferred directly from the tablet PC to the desktop computer. The timing of the transfer plays an important role in preserving the integrity of that character's identity, since showing the character in both the original location and the destination will break the illusion of it being the same entity on both devices.

When two devices have different graphical capabilities, it may not be possible for both devices to represent an entity in the same way; when the entity is transferred between such devices, it undergoes a stylistic transformation that may cause a visual discontinuity. One way to improve graphics and animation across heterogeneous devices involves separating the moment of cross-device transfer from the moment of stylistic transformation. The technique involves implementing an explicit stylistic transformation on the device with superior graphical capabilities, shortly before or after the cross-device transfer. Making the stylistic transformation explicit, and separating it temporally from the transfer, also helps to preserve the identity of the graphical entity as it moves between devices.

5.2.1 Multi-Device Virtual Lighting

The use of multi-device virtual lighting can also help create the illusion of a unified virtual space across the multi-device system. A problem that occurs in virtual worlds displayed on mobile devices is that the shadows do not move according to the physical motion of the device, thereby breaking the illusion of a unified virtual world. The system described here uses physical sensing technologies to supplement the virtual lighting system, thereby allowing for the creation of multi-device virtual lighting.

Consider, for example, if there is a virtual character on a tablet PC with virtual shadows cast from a virtual light source in direction of physical North. When the user turns the tablet PC clockwise, the virtual light source does not change location. So from the user's perspective, the shadow has also turned clockwise, breaking the laws of real-world physics. But if the table PC can sense the physical movement, it can adjust the virtual light source to match the user's expectations. So now when the user turns the tablet PC clockwise, the device will detect the amount of clockwise rotation and rotate the virtual light source a corresponding amount *counterclockwise*. This process insures that the virtual shadows cast by the virtual character will continue to be cast in the direction of physical North in the user's perspective, and the illusion of a consistent virtual world is maintained.

This multi-device virtual lighting system has been implemented in a project based on the Virtual Raft platform. This project consists of three computers: two stationary desktops PCs and one mobile tablet PC. As in the Virtual Raft project, the desktop computers each represent a virtual island and the tablet PC represents a virtual raft. Humanoid and plant characters inhabit the virtual islands. Each character has a shadow that appears as if cast from a light source at an infinite distance. This virtual light source is specific to each virtual island. One of the virtual islands is connected to a knob that supports turning. Users can use this knob to change the direction of the shadows on that particular virtual island.

In order to determine the physical orientation of the mobile tablet PC, we used a dual-axis gyroscope. The gyroscope measures movement along two axes and outputs a rate of rotation. However, with the system we are using, the angle of rotation drifts significantly if the user turns at different speeds. A more effective sensing device for this project could be a digital compass, so the result would not be speed-dependent. Also, with only the gyroscope, we cannot obtain the necessary information about the physical world (namely, physical location) to emulate a point light source. With the help of future versions of GPS, Place Lab (LaMarca et al., 2005), or other positioning systems, it may be possible to implement point light in future iterations of the project. However, such locative technologies would need to be quite precise to be used effectively for virtual point light positioning.

In addition to affecting the placement of the shadow on the mobile devices, the computed rotation data is transmitted over a local network link to the tablet PC, which then adjusts the shadows according to the received rotation data. When the MDVW is first launched, the shadows' directions shown on the various devices are different and do not align with real world shadows. When the virtual characters are transferred between devices, shadow orientation information is copied along with the character data. Thus transferring a character propagates the shadow direction to the receiving computer. So although only the tablet PC is connected to sensing hardware, the shadows on all three virtual spaces will be calibrated after two character transfers—from the first desktop to the tablet PC, and from the tablet PC to the second desktop.

5.3 Multi-Device Sound

Audio is another important aspect to creating an engaging, immersive experience for users.

For most people sound is a pervasive aspect of daily experience, and the unique psycho-acoustic aspect of our hearing helps us to internalize our surroundings through our aural experience of them (Ong, 1982). Creating a well-crafted soundscape for any virtual environment is an important part of bringing that environment to life and making it believable.

There has been much research on using sound in conjunction with mobile devices. Nomadic Radio (Sawhney & Schmandt, 2000) uses an audio interface to deliver notifications to a mobile user. Audio also plays an important part in supporting the creation of virtual environments. For example, Drettakis uses 3D sound to help evaluate urban planning in a virtual environment (Drettakis, Roussou, Reche, & Tsingos, 2007). Multi-device systems have taken advantage of audio design to create a virtual soundscape, such as Audio Aura (Mynatt, Back, Want, Baer, & Ellis, 1998) and 'A New Sense of Place?' (M. Williams, Jones, Fleuriot, & Wood, 2005). In these projects, sounds are linked to specific spatial locations, and users experience the sounds by moving through physical space using location-aware devices and a pair of headphones. In contrast, this platform for multi-device virtual worlds surrounds the user with different devices that each produce audio output, so that the soundscape corresponds to the location in physical space, rather than a location in virtual space.

One major challenge in dealing with audio for multi-device virtual worlds is the synchronization of audio and graphics across different devices. Although synchronization may not be an important issue for background or environmental sound effects, when certain events within the virtual world take place across different devices, coordination and timing of the audio output is crucial. For example, in the Virtual Raft project, when a character jumps from an island to a raft, a sound clip of a voice saying "Whee!" is played on the island, followed by a splashing water sound played on the raft device when the character lands. This aural continuity between devices helps to support the visual continuity achieved by properly timing animations that occur during the transfer, as well as to cue the user as to where he or she should look to follow the action. Improper audio synchronization between devices can confuse users and interfere with their suspension of disbelief. For example, when the sound effect for the cross-device transfer occurs too late, users think that a second transfer event is happening, and may not understand when no transfer actually occurs.

Another difficulty in creating audio for MDVWs involves potential differences between different devices' audio capabilities. Just as different computational devices may have different graphics cards or different resolution displays, different devices may also have different sound cards with different hardware capabilities, as well as different speaker setups ranging from monaural or stereo to 7.1 surround sound. Adapting audio output across devices with different capabilities can be done using different sets of sounds samples, or may involve more complex audio manipulation. This issue of differing audio capabilities also extends to different devices having different speaker configurations, which need to be considered when synchronizing audio.

This framework for multi-device virtual worlds provides some unique opportunities for sound designers to create an immersive audio environment. On the one hand, this environment gives the sound designer slightly less control than a surround sound system in which the locations of each speaker are known and fixed. On the other hand, this combination of stationary and moving sound sources can be used to create a more immersive experience. For example, rather than simulating a hummingbird whizzing around by emitting the sound from various speakers, the hummingbird's wing beats emit from the tablet while it is physically carried around the space. In the EcoRaft project, the ambient sound emitted by each stationary computer also serves as in implicit indication of that island's ecological status. A completely deforested island is almost totally silent, with only a faint, haunting wind blowing in the background. As restoration progresses, the background audio begins to be filled with the sounds of rustling leaves, birds chirping in the distance, and other rainforest noises. Each island has a characteristic set of background sounds, such that the full audio aesthetic of the space can only be appreciated when all the islands are fully restored. Furthermore, each individual species of hummingbird has a slightly different call, so that as different species of hummingbirds are brought to different islands, participants are immersed in a fuller, more complex soundscape. With these multiple aural cues, a participant can simply stand in the middle of the installation and very quickly get a rough impression of both the ecological state of the three islands and the activities occurring in the space. Although this approach might be less accurately informative than having the voiceover say, for example, "Cocos Island is at 87% of full restoration," it fits much better with the aesthetic of the project and helps to make the experience more engaging by immersing participants in a diverse, complex, and informative aural environment.

5.4 Multi-Device Agents

A central element of the multi-device virtual world framework is the use of autonomous and semi-autonomous computational agents that inhabit the world. These *embodied mobile agents* (EMAs) (Tomlinson, Yau, & Baumer, 2006) are designed to be able to operate on any of the devices in the multi-device virtual world, and to transfer seamlessly between devices. EMAs can take a range of forms, including animated characters (personified agents with the appearance of sociality), animated creatures (agents modeled after non-human species), and animated objects (based on inanimate objects in the real world).

EMAs are an important part of a virtual world for a number of reasons. First, they provide a mechanism for connecting interactions on the various devices in a way that is comprehensible to users. Seeing the same agent performing similar tasks across different devices can make a system's operation more transparent. Also, animated characters provide an important means by which computational systems engage users even in single-device virtual worlds—creating characters that live in these multi-device systems can help bring this engagement to bear on the systems being built. Mobile agents in general, whether embodied in an animated form or not, can also be an effective means of transferring data among devices.

One of the challenges with EMAs is enabling the characters to transfer believably across devices. There are a number of factors that contribute to this believability, including the agent's animation on both sides of the transfer, the timing of the movement between devices, graphical effects that can support the animations, and careful integration with multi-device sound. When moving across devices that have significantly different graphical styles, these agents may be enhanced via heterogeneous animation techniques (Tomlinson, Yau, & Gray, 2005) that help smooth over differences (e.g., screen size, resolution) that might otherwise compromise the continuity of the animated transfer between devices.

5.5 Real World Integration

A great deal of research has been done on novel and engaging ways to blur the boundaries between physical and virtual space (e.g., (Ishii, Mazalek, & J. Lee, 2001), (Khoo et al., 2006)). Many of these approaches connect the manipulation of a physical object to the manipulation of some digital entity. For example, the phicons in metaDESK allow the user to physically manipulate the location of digitally displayed structures (Ishii & Ullmer, 1997), and the rattling of Live Wire provides a physical and audible clue as to the current state of digital network traffic (Weiser & J. S. Brown, 1996). In contrast, this work seeks to emphasize the ways in which physical and virtual worlds partially overlap and coexist. The goal here is not to enable users to manipulate digital data physically, but rather to allow physical users and autonomous virtual entities to inhabit the same, hybrid space.

The problem of creating a connection between a virtual world and the physical world, and doing so in a fashion that is familiar to the average human being, has been an ongoing subject of research. “Mixed Reality” projects blend between a virtual world and the physical world. Human Pacman (Cheok et al., 2003) and Age Invaders (Khoo et al., 2006), for example, are played in both the physical and the virtual worlds. Our work, like these systems, seeks to create human interactions across both physical and virtual space.

In both the Virtual Raft and EcoRaft projects, the tablet PCs are equipped with accelerometers, which are able to detect orientation about two axes. When the tablet is tipped front-to-back or side-to-side, the virtual entities contained therein react to the device’s physical orientation. The humanoid characters in the Virtual Raft project try to balance on the raft; if the participant tilts the raft too much, the character falls in the water and his or her torch is extinguished. In EcoRaft, hummingbirds react to tilting by trying to fly to the highest point in their virtual cage, and seeds react by rolling around inside the virtual box the tablet represents. In this way, virtual entities can react to the physical orientation of the device on which they are located, helping to blur the boundary between physical and virtual spaces.

Another example of such blurring involves the use of webcams in these projects. Other researchers have used closed circuit cameras in interactive contexts, for example to situate museum-goers within an interactive art piece (Heath et al., 2002), (Hindmarsh et al., 2005). In the projects presented here, webcams are not used to place images of the participants within the installation, but rather to serve as the virtual characters’ “eyes” out into the physical world. In the Virtual Raft, when a user approaches an island, the webcam mounted atop the large display detects the motion. In response, the characters stand up and approach the front of the screen, giving the appearance that they are walking toward the users who are walking toward them. If there is no motion for a period of time, the characters turn around and return to the central fire. Similarly, in EcoRaft, if the webcam detects motion, hummingbirds will fly up to the front of the screen and hover for a moment directly in front of users. Similar to the way that virtual characters on the tablet PC react to the tablet's physical orientation, the webcams allow characters to react to aspects of their physical surroundings, thus helping to further blur the physical/virtual distinction.

An important aspect in the design of multi-device virtual worlds with respect to blurring the boundaries between the physical and the virtual worlds is the physical placement of devices within an interaction. In both the Virtual Raft and the EcoRaft installations, the three large displays are situated roughly in a circle facing one another, so as to give the impression of three distinct virtual spaces that are not directly connected to one another. Also, these large displays are placed at the edge of the interaction space, so that participants cannot pass behind them. This placement helps to give the impression that, rather than the virtual world being contained within the display, the display is a window to a virtual space on the other side of the screen.

The overall goal is to give the impression that human users and virtual characters both inhabit one continuous interaction space. The methods described above for blurring the boundary between physical and virtual space are some examples for how this goal is achieved in the projects described in this paper.

6 Evaluation

The projects implemented using this framework have been shown to a wide variety of audiences, with over 3000 participants at a number of different conference venues including ACM SIGGRAPH (Tomlinson et al., 2005a), ACM CHI (Tomlinson et al., 2005b), CSCL (Tomlinson, 2005), and AIIDE (Baumer, Tomlinson, Yau, & Alspaugh, 2006). In addition, these projects have been demonstrated to several hundred participants in a university lab space, as well as through temporary installations at the Discovery Science Center (DSC) in Santa Ana, CA. During these exhibitions, the project teams observed and took notes about users' interactions with and reactions to the installations. A number of different specific evaluations were also performed, including a series of open-ended, semi-structured interviews with participants at SIGGRAPH and DSC (Tomlinson, Baumer, Yau, & Black, in submission), as well as a video analysis of participant interactions at AIIDE. The following section reports on users' general reactions to and experiences with the various systems, as well as the development teams' experiences designing and implementing virtual worlds using multi-device systems.

6.1 User Experiences

During the various deployments, users on the whole enjoyed interacting with these multi-device virtual worlds. When asked about preferences between these multi-device interaction paradigms and single-device ones, most preferred the multi-device interactions, responding, "Yes, this is much better," "I actually like the fact that you can move things around," or "it destroys the normalcy of what the society thinks is like normal computer interfacing." Participants also appreciated the physical aspects of the project: "I like it ... 'cause you are walking and moving.... You feel like you are carrying the hummingbird ... instead of just clicking and dragging." Some participants commented that they "liked the physicality of it, ... the fact that you walked around." While such reactions are not uncommon for systems with tangible interfaces, the CMDVW paradigm goes farther, allowing participants to share the same space with virtual entities that led to a sense of connection. One felt that "when you hold something with two hands you know that it's very important," and thus the need to carry the tablet with two hands made "each element ... sacred and special." Another said that the way the virtual entities on the tablet reacted to its orientation made them into "quasi-physical objects." These comments indicate that our efforts at blurring the boundaries between the physical and virtual spaces are successful, and that they lead to more engaging user experiences.

During the evaluation of EcoRaft as a tool to facilitate children's learning about restoration ecology (Tomlinson et al., in submission), several important themes emerged. Evaluations consisted primarily of interviews with children and adults who had interacted with the EcoRaft exhibit, supplemented by observations of participants interacting with the system. The transcripts of those interviews and observation notes were analyzed using qualitative methods (J. Lofland & L. Lofland, 1995). The findings highlight the effectiveness of MDVWs, and the EcoRaft system in particular, in creating an environment conducive to collaborative, discovery based learning. A portion of those findings are summarized here.

In this evaluation process, each of the fourteen students interviewed mentioned the importance of collaboration. During observations, students were noted telling one another about the ecosystem's dynamics, e.g., why a seed would not grow or why a hummingbird flew away. The way in which collaboration permeated interview and observation data indicate that the MDVW design was likely effective in encouraging social interaction.

Another important aspect supported by the design was different roles for participants to play. MDVWs can provide a wealth of interaction modalities; EcoRaft allows participants to carry three different species of organisms on three different tablets, to directly affect those organisms through the physical orientation of the tablet, to protect the islands by preventing others from pressing the silver buttons, to affect hummingbird behavior through the webcams, and to interact in a variety of other ways. Furthermore, because of its physical configuration, EcoRaft also provides participants indirect ways of interacting with the installation, through observing, questioning, or making suggestions to those interacting with it. In the learning context for which EcoRaft was designed, these different interaction capacities support a broad range of learning styles.

These evaluations also led to a set of unanticipated findings about unintended results of certain physical aspects of the system. For example, one adult noted that, due to the weight and heft of the tables, the objects contained therein acquired a certain preciousness. “When you hold something with two hands” as one must do with the tablets, “you know that it’s very important.” On the other hand, “mov[ing] it with a mouse drag and drop, you kind of lose that interactivity that makes something sacred.” While the incorporation of this theme was an unintended effect of using the tablets, the importance and sacredness of the objects being carried on the tablets helps participants assume the role of restoration ecologists that is central to EcoRaft.

6.2 Developer Experiences

Designing virtual worlds for multi-device systems presents a number of difficulties with respect to the development process, including keeping track of which code runs on each device, handling code distribution across different devices, code versioning among multiple devices, and debugging interactions between different devices. This section describes various approaches used in managing this development process, along with the strengths and weaknesses of each method. The authors hope that sharing their experiences in developing these systems will aid others in the development of similar systems by helping to avoid possible pitfalls and pointing out useful techniques.

Our initial approach for developing systems for multi-device virtual worlds was to load a separate, full copy of the code base, stored in a central CVS repository, onto each device used in an installation. This method had the advantage that launching the installation was relatively fast, because all the code was being executed locally. However, there were often occasions when the code on two different machines was not identical, so when Java tried to serialize an instance of a class for transmission between devices, the class definition on the two devices did not match and the transmission failed. This meant that whenever developers wanted to test the installation, they either needed to check their code into the repository or manually copy the code between devices. Using CVS had the advantage of automatically checking for conflicts between files, but the code in question was often being tested, and development practices dictated that only properly functioning code be committed to the repository. Copying the code directly between devices had the advantage of ensuring that improperly functioning code was not committed into the repository, but it required extra time and care to make sure that no changes were inadvertently overwritten. This approach also made it easy to fix a problem on one or two devices, but failed to insure that the change was committed to the repository so that all devices were up to date. Both of these methods for developing multi-device software were time consuming and error prone, so a better solution was pursued.

The next approach was to use one central machine as a code server and let all the other devices in the installation mount a network drive to connect to the code server, ensuring that any time the installation was run, all machines were running the same code. This solved the problem of different devices having different versions of the code when attempting to run the entire system, but it did not alleviate the difficulties regarding version control, and at times made the development process even more difficult. Despite all devices running code from a single location, the code on that device still needed to be updated with the latest version from the repository before launching the installation. Furthermore, a developer would often make edits on his or her local workstation, but neglect to either to commit the changes to the repository or to check out from the repository onto the code server. As such, fixed bugs would seem to mysteriously reappear, sometimes on multiple occasions. Furthermore, though it was not significant enough to make the installation not engaging, there was a slight but noticeable degradation in performance due to running the code over mounted network drives.

Developing any sort of multi-device system poses difficulties similar to those listed above, as well as other challenges. In the future work section we describe some ongoing work to address these problems in a consistent, effective manner by developing a set of software engineering tools.

7 Future Work

This research is continuing to progress in a number of areas, including the design of software engineering tools to enhance development of these kinds of systems, the development of reputation modeling to coordinate interactions among devices, and the creation of a series of exhibits based on different regional ecosystems.

7.1 Multi-Device Software Engineering

Developing software for multiple platforms has proved to be an arduous task, as current widely available programming tools have no functionality to aid development in which code is written on one computer and run on another. Even though the Virtual Raft platform was written in Eclipse, one of the most popular integrated development environments for Java programming, we had no automated way to write code for two or more separate interacting programs on one of our desktop PCs, compile the programs, transfer the appropriate programs to various tablets and desktops, and run all the programs on their respective devices. While Eclipse does provide some remote debugging tools, including the ability to connect to a remotely running Java virtual machine, it lacks the sort of all-inclusive solution required. Instead, our software development team had to coordinate the transfer of files manually—a tedious process which was exacerbated by the frequency of human error that arises when performing mundane repetitive tasks. Furthermore, this requirement often obstructed the flow of creative thought required to develop a system the size of the Virtual Raft platform, which consists of many tens of thousand lines of code. On top of these difficulties, many of the features coders have become accustomed to (such as the ability to perform an automatic stack trace in Eclipse) were not available to our team when writing code for a remote machine, as remote debugging tools were not available at that time. This long debugging cycle also discouraged our programmers from experimenting with their code, as making even a simple change would take a significant amount of effort to disseminate across multiple devices, arguably causing a worse end product than could have been achieved had less time and labor been necessitated.

To begin to patch this hole in tool functionality, we are developing a plug-in to Eclipse that will facilitate the development of multi-device systems by automating the debugging cycle across multiple platforms. This plug-in will be supplied with the IP addresses of remote devices, and the remote devices will run a server program to accept connections from the instance of Eclipse running on the local device. Files will then be labeled as remote, local, or some combination of these two (so any one file can be a used by the local device as well as any number of remote devices simultaneously). When the program is told to run, all files marked as remote will be transferred to the appropriate remote devices, and the all the programs (both local and remote) will be run.

During execution on the remote device, the standard error and standard out streams will be piped from the remote device to the local device. We hope this will create the illusion that the remote programs are running on the local device, decreasing perceived complexity. Also, by forwarding all programming data to the locally running instance of Eclipse, all of the powerful functionality that comes with Eclipse can be used. We hope this plug-in will greatly increase productivity in the development of multi-device systems, decrease the present difficulties that come with creating such systems, and give developers the freedom to experiment with their multi-device systems during development, thereby creating better end products.

7.2 Network of Exhibits

These multi-device software engineering tools will be particularly useful in the next stage of this research project. The EcoRaft exhibit described above focuses on a Costa Rican rain forest ecosystem. The future stages of this project will develop a network of six interactive museum exhibits based on common themes in restoration ecology. Each of the six exhibits will address an ecological issue that is relevant to the geographical region in which it is displayed. For example, an exhibit in Florida might feature the snakehead fish and one or more native species of fish, while an exhibit in Minnesota might feature wolves and rabbits. Each participating museum will be able to run the regional content that was developed for the other museums as well, thereby encouraging repeat visitation and helping visitors learn about ecological principles that stretch across different ecosystems. Development of these exhibits will build upon and enhance the core infrastructure of the multi-device virtual world system as well.

7.3 Reputation Modeling Across Devices

We are also working on creating a reputation-management system for modeling trust on multi-device systems. This system will allow a device to receive information (such as location, air quality, or any other sensed values) from other collocated devices, and then make informed decisions based on the information gathered. For example, a PDA could determine its location based on the information shared by its GPS-enabled neighbors, even without having an embedded GPS device itself. Such a system would allow the PDA to properly value received information based on how trustworthy it believes its neighbors are. Thus we will be able to propagate information throughout a heterogeneous multi-device system, even systems with unknown or distrusted devices.

Such modeling is necessary for multi-device systems that do not know in advance what devices will be involved. A reputation modeling system would be able to accommodate the introduction of "strange" devices into a system while decreasing the risk of malicious devices corrupting the system (Srinivasan, Teitelbaum, & Wu, 2006). We will be able to create multi-device systems which do not require custom hardware —for example, we could create a MDVW in which a user's personal PDA or cell phone could act as a virtual island or virtual raft. Also, by using this reputation system to disseminate information, we will be able to generate contextual information about the physical world with a smaller number of sensors. Thus we will be able to create an MDVW with even greater blurring between the physical and virtual worlds.

8 Conclusion

This paper has presented a framework for the design and implementation of interactive systems across networks of collocated devices. In particular, it has focused on the use of multi-device virtual worlds as an interaction paradigm for creating coherent experiences across multiple heterogeneous devices. This framework incorporates multi-device networking, graphics, sound, embodied mobile agents and real world sensing. The framework was used in the production of two interactive projects: the Virtual Raft project and the EcoRaft project. An evaluation of the framework was also offered, describing the ways in which this interaction paradigm both created an engaging experience for users and provided designers with a conceptual grounding for implementing the two projects described. While multi-device virtual worlds are not a solution to the entire broad problem of enabling people and devices to work together more effectively, these worlds do provide a possible means for creating a coherent interaction paradigm across multiple collocated devices.

As computational devices become more common across human societies, the potential usefulness of groups of these devices that are physically proximate to each other increases significantly. The growing frequency with which people find themselves in the presence of several different devices necessitates more effective ways for people to engage with those devices as systems, rather than in isolation. Just as people can achieve greater functionality when they work together, so too can devices become more useful and effective when they are enabled to operate smoothly together. Multi-device virtual worlds are just one example of this potential future application domain.

9 Acknowledgments

This research was funded the Nicholas Foundation, the Emulex Foundation and NSF grant IIS-0644415. It was supported by the California Institute for Telecommunications and Information Technology (Calit2) and the Donald Bren School of Information and Computer

Sciences. The authors thank the other members of the Social Code Group, who contributed greatly to the implementations and ideas described here.

10 References

Baecker, R. M., Grudin, J., Buxton, W. A. S., & Greenberg, S. (Eds.). (1995). *Reading in human-computer interaction*. San Francisco: Morgan Kaufmann.

Barab, S., Thomas, M., Dodge, T., Carteaux, R., & Tuzun, H. (2005). Making learning fun: quest atlantis, a game without guns, *Educational Technology Research and Development*, 53(1), 86-107. doi: 10.1007/BF02504859.

Barkhuus, L., Chalmers, M., Hall, M., Tennent, P., Bell, M., Sherwood, S., et al. (2005). Picking pockets on the lawn: the development of tactics and strategies in a mobile game In LNCS 3660 (pp. 358-374). Tokyo: Springer.

Baumer, E., Tomlinson, B., Yau, M. L., & Alspaugh, T. A. (2006). Normative echoes: use and manipulation of player generated content by communities of npcs, *Artificial Intelligence and Interactive Digital Entertainment (AIIDE-06).*

Bell, & Dourish. (2007). Yesterday's tomorrows: notes on ubiquitous computing's dominant vision, *Personal and Ubiquitous Computing*, 11(2), 133-143. doi: 10.1007/s00779-006-0071-x.

Biocca, F. (1997). The cyborg's dilemma: embodiment in virtual environments In , *Cognitive Technology, 1997. 'Humanizing the Information Age'. Proceedings., Second International Conference on* (pp. 12-26).

Borovoy, R., Martin, F., Vemuri, S., Resnick, M., Silverman, B., & Hancock, C. (1998). Meme tags and community mirrors: moving from conferences to collaboration In *Proceedings of the 1998 ACM conference on Computer supported cooperative work* (pp. 159-168). Seattle, Washington, United States: ACM.

Brumitt, B., Meyers, B., Krumm, J., Kern, A., & Shafer, S. A. (2000). Easyliving: technologies for intelligent environments In *Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing* (pp. 12-29). Bristol, UK: Springer-Verlag.

Card, S., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Chalmers, M., Dieberger, A., Höök, K., & Rudström, Å. (2004). Social navigation and seamful design, *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society*, 11(3), 171-181.

Cheok, A. D., Fong, S. W., Goh, K. H., Yang, X., Liu, W., & Farzbiz, F. (2003). Human pacman: a sensing-based mobile entertainment system with ubiquitous computing and tangible interaction In *Proceedings of the 2nd workshop on Network and system support for games* (pp. 106-117). Redwood City, California: ACM.

Drettakis, G., Roussou, M., Reche, A., & Tsingos, N. (2007). Design and evaluation of a real-world virtual environment for architecture and urban planning, *Presence: Teleoper. Virtual Environ*, 16(3), 318-332.

Drucker, S. M., He, L., Cohen, M., Wong, C., & Gupta, A. (2003). Spectator games: a new entertainment modality for networked multiplayer games. Available at: research.microsoft.com/~sdrucker/papers/spectator.pdf

El-Nasr, M. S., Zupko, J., & Miron, K. (2005). Intelligent lighting for a better gaming experience In *CHI '05 extended abstracts on Human factors in computing systems* (pp. 1140-1141). Portland, OR, USA: ACM.

Grudin, J. (1988). Why groupware applications fail: problems in the design of organizational interfaces In *Proc. ACM Conf. Computer-Supported Cooperative Work CSCW'88* (pp. 85-93). Portland, OR.

Grudin, J., & Palen, L. (1995). Why groupware applications succeed: discretion or mandate: In *Proc. European Conf. Computer-Supported Cooperative Work ECSCW'95*. Stockholm, Sweden.

He, L., Cohen, M. F., & Salesin, D. H. (1996). The virtual cinematographer, *Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques*, 217-224.

Heath, C., Luff, P., Lehn, D. V., Hindmarsh, J., & Cleverly, J. (2002). Crafting participation: designing ecologies, configuring experience, *Visual Communication*, 1(1), 9-33.

Hindmarsh, Heath, Lehn, D. V., & Cleverly. (2005). Creating assemblies in public environments: social interaction, interactive exhibits and cscw, *Computer Supported Cooperative Work (CSCW)*, 14(1), 1-41. doi: 10.1007/s10606-004-1814-8.

Hughes, C. E., Reinhard, E., Konttinen, J., & Pattanaik, S. (2004). Achieving interactive-time realistic illumination in mixed reality, *Army Science Conference*.

Hutchins, E. (1996). *Cognition in the wild*. Cambridge, MA: The MIT Press.

Ishii, H., Mazalek, A., & Lee, J. (2001). Bottles as a minimal interface to access digital information In *CHI '01 extended abstracts on Human factors in computing systems* (pp. 187-188). Seattle, Washington: ACM.

Ishii, H., & Ullmer, B. (1997). Tangible bits: towards seamless interfaces between people, bits and atoms In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 234-241). Atlanta, Georgia, United States: ACM.

Kaptelinin, V., & Nardi, B. A. (2006). Artifacts, agency, and (a)symmetry In *Acting With Technology* (pp. 237-252). Cambridge, MA: MIT Press.

Khoo, E. T., Lee, S. P., Cheok, A. D., Kodagoda, S., Zhou, Y., & Toh, G. S. (2006). Age invaders: social and physical inter-generational family entertainment In *CHI '06 extended abstracts on Human factors in computing systems* (pp. 243-246). Montreal, Quebec, Canada: ACM.

LaMarca, Chawathe, Consolvo, Hightower, Smith, Scott, et al. (2005). Place lab: device positioning using radio beacons in the wild, *Pervasive Computing*.

Latour, B. (1992). Where are the missing masses? the sociology of a few mundane artifacts In *Shaping Technology/Building Society* (pp. 225-258). Cambridge, MA: MIT Press.

Law, J. (1986). On the methods of long distance control: vessels, navigation, and the portuguese route to india In J. Law (Ed.), Sociological Review Monograph. (pp. 234-263). Henley: Routledge. Retrieved from http://www.lancs.ac.uk/fss/sociology/papers/law-methods-of-long-distance-control.pdf.

Lee, C. P., Dourish, P., & Mark, G. (2006). The human infrastructure of cyberinfrastructure In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work* (pp. 483-492). Banff, Alberta, Canada: ACM.

Lofland, J., & Lofland, L. (1995). *Analyzing social settings: a guide to qualitative observation and analysis*. Belmont, CA: Wadsworth.

Mynatt, E. D., Back, M., Want, R., Baer, M., & Ellis, J. B. (1998). Designing audio aura In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 566-573). Los Angeles, California, United States: ACM Press/Addison-Wesley Publishing Co.

Naemura, T., Nitta, T., Mimura, A., & Harashima, H. Virtual shadows in mixed reality environment using flashlight-like devices, *Transactions of the Virtual Reality Society of Japan*, 7.

Nardi, B. A. (1996). Studying context: a comparison of activity theory, situated action models, and distributed cognition In *Context and Consciousness: Activity Theory and Human-Computer Interaction*. Cambridge, MA: MIT Press.

Norman, D. (1988). *The psychology of everyday things*. New York: Basic Books.

O'Hare, G. M. P., & Duffy, B. R. (2002). Agent chameleons: migration and mutation within and between real and virtual spaces, *The Society for the Study of Artificial Intelligence and* *the Simulation of Behaviour AISB*, 3-5.

Ong, W. (1982). *Orality and literacy* (2003). New York: Routledge.

Rekimoto, J. (1997). Pick-and-drop: a direct manipulation technique for multiple computer environments In , *Proceedings of the 10th annual ACM symposium on User interface software and technology* (pp. 31-39). Banff, Alberta, Canada: ACM.

Roussos, M., Johnson, A., Moher, T., Leigh, J., Vasilakis, C., & Barnes, C. (1999). Learning and building together in an immersive virtual world, *Presence: Teleoper. Virtual Environ*, 8(3), 247-263.

Sawhney, N., & Schmandt, C. (2000). Nomadic radio: speech and audio interaction for contextual messaging in nomadic environments, *ACM Trans. Comput.-Hum. Interact*, 7(3), 353-383.

Srinivasan, A., Teitelbaum, J., & Wu, J. (2006). Drbts: distributed reputation-based beacon trust system In *Dependable, Autonomic and Secure Computing, 2nd IEEE International Symposium on* (pp. 277-283). Retrieved December 11, 2007, from http://ieeexplore.ieee.org/iel4/4030851/4030852/04030893.pdf.

Stock, O., & Zancanaro, M. (2002). Intelligent interactive information presentation for cultural tourism, *Proc. of the International CLASS Workshop on Natural Intelligent and Effective Interaction in Multimodal Dialogue Systems*, 28-29.

Sumi, Y., & Mase, K. (2001). Agentsalon: facilitating face-to-face knowledge exchange through conversations among personal agents In *Proceedings of the fifth international conference on Autonomous agents* (pp. 393-400). Montreal, Quebec, Canada: ACM.

Tomlinson, B. (2005). A heterogeneous animated platform for educational participatory simulations In *Proceedings of th 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years* (pp. 677-681). Taipei, Taiwan: International Society of the Learning Sciences.

Tomlinson, B., Baumer, E., & Yau, M. L. (2006). The island metaphor In *ACM SIGGRAPH 2006 Research posters* (p. 84). Boston, Massachusetts: ACM.

Tomlinson, B., Baumer, E., Yau, M. L., & Black, R. (in submission). A participatory simulation for informal education in restoration ecology, *Submitted to: the Journal of Science Education and Technology*.

Tomlinson, B., Blumberg, B., & Nain, D. (2000). Expressive autonomous cinematography for interactive virtual environments In *Proceedings of the fourth international conference on Autonomous agents* (pp. 317-324). Barcelona, Spain: ACM.

Tomlinson, B., Yau, M. L., & Baumer, E. (2006). Embodied mobile agents In *Proceedings of the* *fifth international joint conference on Autonomous agents and multiagent systems* (pp. 969-976). Hakodate, Japan: ACM.

Tomlinson, B., Yau, M. L., Baumer, E., Goetz, S., Carpenter, L., Pratt, R., et al. (2006). The ecoraft project: a multi-device interactive graphical exhibit for learning about restoration ecology In *CHI '06 extended abstracts on Human factors in computing systems* (pp. 1445-1450). Montreal, Quebec, Canada: ACM.

Tomlinson, B., Yau, M. L., & Gray, J. (2005). Heterogeneous character animation: how to make an interactive character jump between stationary and mobile graphical computing platforms In *ACM SIGGRAPH 2005 Sketches* (p. 18). Los Angeles, California: ACM.

Tomlinson, B., Yau, M. L., Gray, J., Baumer, E., O'Connell, J., Williams, K., et al. (2005). The virtual raft project: a network of mobile and stationary computer systems inhabited by communities of interactive animated agents In *ACM SIGGRAPH 2005 Emerging technologies* (p. 30). Los Angeles, California: ACM.

Tomlinson, B., Yau, M. L., O'Connell, J., Williams, K., & Yamaoka, S. (2005). The virtual raft project: a mobile interface for interacting with communities of autonomous characters In *CHI '05 extended abstracts on Human factors in computing systems* (pp. 1150-1151). Portland, OR, USA: ACM.

Weiser, M., & Brown, J. S. (1996). Designing calm technology, *PowerGrid Journal*, 1(1), 75-85.

Williams, M., Jones, O., Fleuriot, C., & Wood, L. (2005). Children and emerging wireless technologies: investigating the potential for spatial practice In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 819-828). Portland, Oregon, USA: ACM.

Figure Captions

Figure 1: Several children interact with a multi-device virtual world in the EcoRaft exhibit.