Visualising Sound Perception in a Submarine: A Museum Installation

Emmanuel Frécon, Olov Ståhl, Jonas Söderberg, Anders Wallberg
Swedish Institute of Computer Science
Box 1263
SE-164 29 Kista
Sweden
{emmanuel, olovs, jas, andersw}@sics.se

Abstract

We describe a museum installation that explains the technical and mental process that sonar operators undergo when identifying underwater sounds in the surroundings of a submarine. The installation places the public in a cramped space composed of several coupled interactive stations offering different perspectives onto a virtual environment representing a part of the Baltic Sea. The virtual environment and its presentation within the installation are implemented as an application of the DIVE research toolkit. The installation has been on display in several museums on a daily basis for over a year. We describe the technical solutions that we have employed to realise the installation and some of our learning.

1. Introduction

This paper describes the design and implementation of a museum installation that illustrates how a submarine uses sound to gather information about its surroundings. A submarine under water is more or less blind and relies heavily on its sonar system to detect other vessels. Although the periscope can be used to spot surface vessels, its range is limited to a couple of miles. A sonar on the other hand can pick up sounds from objects hundreds of miles away. To make this ambivalence clear to visitors, the installation relies heavily on sound, using a soundscape consisting of authentic underwater recordings of a range of different types of vessels and animals.

The installation is a collaboration between SICS and the Swedish national maritime museums and is part of a long-term touring exhibition to celebrate the centennial of the Swedish submarine force. The installation described here has been on display for over a year in so far four museums across Sweden. Part of the work has been conducted within the SSF DAPHNE project Technically, the installation is an application based on the Distributed Interactive Virtual Environment (DIVE) toolkit, a research prototype for the development of virtual environments, user interfaces and applications based on shared 3D synthetic environment (see [1] and [2]).

Logically, the installation is based on a shared virtual environment that represents a part of the Baltic Sea south of Stockholm. Within this environment navigate a number of different objects: the submarine used for the purpose of the explanation and a number of vessels and animals. The installation shows this environment from within three very different perspectives at a number of publicly accessible and interactive stations. One station takes the form of an interactive nautical chart. A second station is a digital periscope. The third station alternatively visualises the direction of incoming sonar waves onto the submarine and an identification process of audible nautical objects. These three stations are complemented by a surround sound system that renders the sound of all objects in the vicinity of the submarine.

In the following sections we describe a number of more detailed requirements that the installation have had. We also present the technical solutions that we have employed to minimise risks and maximise user experience at each station while still utilising the different perspectives on a single shared environment and providing cues to cross-reference these.

2. Motivation

The submarine installation has been on constant display to the public for over a year. It is routinely started during the mornings and switched off at museum closing time. The challenge of taking a research prototype such as DIVE into such a stringent environment is the major motivation for our involvement in the project.

In practice, there were two different actors involved in ordering the installation. Once we had recognised

Figure 1. A view of the installation in Malmö. There the space also allowed for two back projection screens, one on each side of the room, showing a looped recording of the forward and aft view of the control room of a Swedish submarine.

Figure 2. In Gothenburg, the installation was built on the lower deck of a lighthouse boat, in a floating ship museum. This added a lot to the atmosphere but forced us to skip the back projections due to the confined space.

this, we looked into using participatory design strategies when designing the installation and the components that it would include. This resulted in a more concerted exhibit where the requirements of all parties were taken into account. We have previously used such strategies in a number of projects and these have shown the importance of the dialogue that occurs.

Previous experience with an installation called the Pond [3] had shown us the importance of shoulder-to-shoulder collaboration. When designing the submarine installation, we wished to explore this concept further. The Pond offered a single perspective on the environment. For this installation, we chose a strategy that would be based on a shared virtual environment, without any representation of participants, without any navigation and with perspectives so different that the notion of "sharing" almost disappears. The technical challenge has been for us to see whether our system, which is highly tuned for navigable shared environments, could successfully be used in such a different context.

3. Background and Related Work

The ICE laboratory has a long tradition in building VR-based applications that uses shared virtual worlds for information presentation and exploration, for example [9] and [10]. Typically, collaboration in shared virtual environments assumes that each participant sees the same content, still from a different perspective. However, earlier experiences in 2D interfaces have shown that this is not totally adequate and this has led to the introduction of "subjective views" ([4] and [5]). These are usually put in the context of fully navigable environments and result in applications where the view onto the environment is tweaked depending on the participants roles. In this paper we show how the subjective view concept can be used as the basis for a VR application consisting of several processes, each presenting a completely different view of a common environment to the museum visitors.

VR or 3D systems have seen more widespread use within museum settings in recent years, especially in the area of archaeology and cultural heritage. However, most applications are either single-user [14] where the system is mainly used to build a graphical GUI, or focus on collaboration or communication between human users [13]. In the submarine installation on the other hand, we use the shared aspect of a VR system as a foundation for building a multi-process application, which is able to present several GUIs in multiple stations.

Periscope-like devices for information presentation have been used in several projects. In [12], an application is described where users are able to experience a historical world (including sounds) using a periscope. Another project [11] uses a periscope in a woodland setting to present QuickTime movies to children that illustrates various aspects of how the wood changes character over a season and how this affects some of its inhabitants.

4. Requirements

The strongest requirement for the technological implementation was the high degree of reliability that the system would have to achieve. Bringing a research prototype in the picture to solve the task is not a

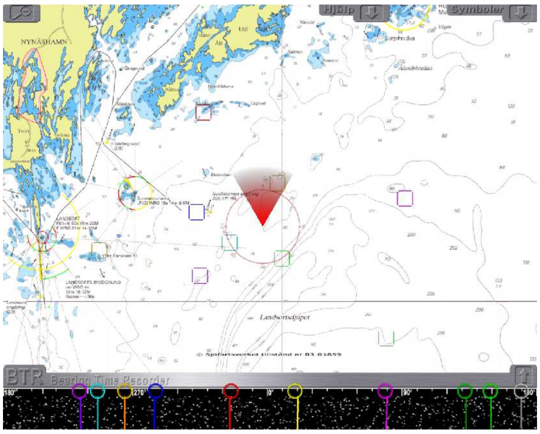

Figure 3. On the interactive chart, the submarine is at the middle of the circle. Unidentified nautical objects are represented by squares and colour codes allow to match them on the BTR at the bottom of the screen.

Figure 4. An example screen shot of the view seen in the digital periscope. The direction and extent of the current periscope view is represented by the disc arc at the centrer of Figure 3.

straightforward solution. However, years of development have made us confident in the increasing stability of the DIVE system and its viability as a more industrial and robust platform. Actually, the successful realisation of this very installation has given us even more confidence in the readiness of the system for more demanding applications.

This requirement on reliability was seconded by the nature of the project as a *public* installation. Consequently, the different stations have to explain enough without any need to actually be explained. We attempted to design the stations, the interaction and the application behind them as robustly as possible. While the different stations are designed for a few individuals gathering around them, we wished that others passing by would be able to understand what was going on and to learn from this experience.

Also, during the design process we had the requirement to provide an installation that would explain how it was to be in a submarine and how a submarine perceived its environment. Therefore the experience should be as real as possible, while still being explanatory enough. An example of this thin balance between realism and didacticism is the nautical chart. Rather than the regular map used in submarines, the station combines this map with the result of the object recognition process itself through icons for unidentified vessels. Thus it acts as a central gluing and explaining artifact in the installation.

5. The Installation

The installation room is intended to represent a submarine, submerged somewhere in the Baltic Sea. The physical setup of the installation has been slightly altered for each of the museums visited so far to make best use of the available space. However, the content and user experience have remained more or less similar. The basic idea has been to create a somewhat cramped space reminding visitors of the restricted space aboard a submarine.

The large-scale visual experience of the exhibition is driven by four wall mounted 40-inch plasma displays, one on each wall. In the room are also two "hands-on" interactive stations. The first is a podium containing an almost horizontal 21-inch touch screen display showing an interactive chart. The second interactive station is a digital periscope. These can be seen in Figure 1 and Figure 2.

Two of the plasma displays mirror what is seen in the periscope and in the nautical chart podium to allow groups of people to experience what is on display. The remaining two plasma displays shows an underwater scene of the submarine with its sonar sensors and the sound environment it is in. Incoming sound waves are displayed as colour-coded circles approaching the submarine, as shown in Figure 5. During a user triggered sound analysis these displays will also show frequency spectrum, sonar operator video and eventually the identified vessel, as shown in Figure 6.

An important part of the installation is the computer controlled eight-channel sound system responsible for rendering the soundscape. Eight loudspeakers are evenly spread out around the room making it possible to simulate sounds coming from all directions.

5.1. User Interaction

Via the eight channel sound system, visitors are presented with a cacophony of sounds coming from all

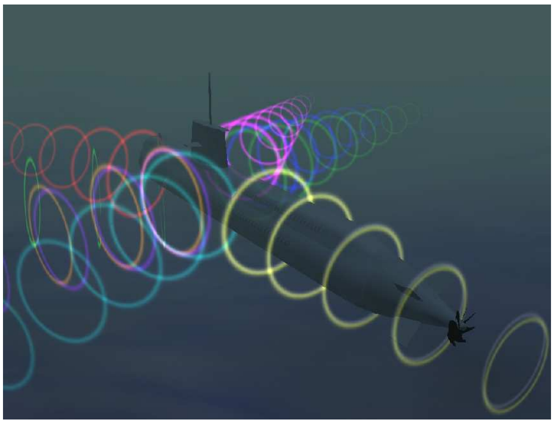

Figure 5. A screen shot showing the submarine and sound waves coming from acoustic objects. Waves are implemented using particle systems.

Figure 6. When a sound analysis phase is initiated, video windows are placed over the underwater scene shown in Figure 5 to illustrate the work of a sonar operator.

directions, representing a multitude of objects surrounding the submarine. Via the touch sensitive nautical chart (see Figure 3), visitors are able to click on symbols representing these sound-emitting objects. When such a symbol is selected via a click, all other sounds will fade away and the selected sound will be played in isolation. During this time the visitors are able to experience the individual characteristics of this sound, be it a seal, a small boat with an outboard motor, or a large tanker. By selecting different symbols and thus sounds, visitors are able to discover that different types of vessels and animals produce quite different sounds. When a selected sound has been played for a short while, all object sounds will start again, making the previously selected sound indistinguishable.

When the installation is started, all sound objects visible in the nautical chart are displayed using a symbol that represents unidentified objects. This means that visitors do not initially know the identity of the objects producing the sounds they are hearing. When a symbol is selected, an info panel is shown on the chart display, showing an image of the vessel or ship generating the corresponding sound. If the object has not been identified, the image consists of a large question mark. By clicking on the info panel of an unidentified object, a visitor may start a simulated sound analysis phase, intended to illustrate the work of an sonar operator. The phase consists of three video sequences displayed in the underwater scene, showing a frequency spectrum (a so called LOFARgram) of the selected sound, a sonar operator at work and eventually the identified vessel or animal, as shown in Figure 6. When the videos have ended to indicate that the sound has been identified, the object's symbol in the chart is changed to represent the identified vessel or animal. The object's info panel image is also updated.

Visitors will also encounter the BTR (Bearing Time Recorder) tool which is used aboard modern submarines. The BTR shows the bearing of the sound objects relative to the submarine over time as plots in a scrolling display. Selecting the plot of an unidentified object in the BTR starts the same sound analysis phase as described above.

The periscope presents a graphical view of the ocean surface at the position of the submarine, and can be rotated to look for ships in all directions. Ships that are within a couple of miles distance from the submarine will be visible as 3D objects moving slowly through the oceans waves (see Figure 4). By rotating the right periscope handle, visitors may change the zoom factor between 1.5 and 6. The left handle controls the tilt-angle, allowing the periscope to look upwards or downwards in the virtual environment. These controls are similar to the ones available on most modern submarines. An electrical motor in the periscope allows visitors to adjust its height via a button. Moving the periscope head up or down doesn't have any affect on the virtual view, it's simply a way for visitors to find a comfortable viewing position.

5.2. Design Process and History

It is through a demonstration of the Pond in a trade fair that we got in contact with the first curator of the Swedish national maritime museums. This developed into a series of meetings with his team to discuss what the installation would have to mediate. The global understanding that emerged from these meetings was that visualisation would form the essential part of the installation. Acoustic information from the sonar is shown aboard in two different ways: the bearing of the acoustical object (BTR) and graphical frequency curves (LOFARgram). Using this information the operator builds an internal representation of the surroundings. We were commissioned to visualise this technological and mental process to the public.

Our proposal was to use a combination of projections surfaces and of a video projector mounted on articulated telescopic arm to let visitors progressively discover a virtual environment through interaction with a console in the middle of the room. The projector would "paint" the acoustic objects composing the environment with more and more accuracy in concert with interaction, thus being similar to active sonar pulses. This environment would be seconded by an acoustic landscape that would use similar ideas and would be refined with interaction.

Using several mock-up videos, we presented our concept to our contact group and later on to a reference group mainly formed of retired navy officers. The reference group reacted very negatively to the proposal. The objective part of the criticism was geared towards two problems. Submarines have no windows. Therefore, showing something outside of the submarine is inappropriate and can be misleading. Also, submarines almost never use active sonar pulses, using them means exposing the submarine.

We started all over again and decided to use a method based on participatory design ideas [6]. Together with the first curator, we created our own reference group. This group mixed people with experience in underwater sound recording and acoustics and an active submarine officer. In a series of meetings, we alternated requests for information needed for the installation and sessions where they could freely tell us how *they* would design the installation. Slowly we aggregated all information and designed a new concept with the first curator. We had finally understood that the real challenge of this project consisted in finding the right balance between, on one side, the requirements for realism and authenticity that the navy had and, on the other side, the requirements for experience and pedagogical clarity that the museum had.

6. Implementing the Installation in DIVE

The submarine installation is implemented using the DIVE toolkit, running on three Windows XP 2.4GHz Pentium 4 computers equipped with GeForce4 class graphics cards. These computers are networked together without any connection to the outside world. The following sections will describe the hardware and the software environments; describe some technical issues regarding the design and implementation as well as experiences learned as the work progressed.

6.1. The DIVE Toolkit

DIVE is a long-established system for CVE research prototyping. It is especially tuned to support multi-participant virtual environments over the Internet. The toolkit has been in development at SICS for more than a decade and binaries are available for a number of platforms.

At the networking level, DIVE is based on a peer-to-peer approach. Peers communicate by reliable and non-reliable IP multicast, following an approach pioneered by SRM (scalable reliable multicast). Conceptually, all peer processes partially share a hierarchical database populated by objects termed *entities* over the Internet. There is one such database per shared virtual environment and processes interact by concurrently accessing and modifying it. The implementation partially replicates the database at all connected peers.

In a typical DIVE world, a number of actors (the conceptual representations of human users) leave and enter dynamically. Additionally, several applications exist within a world. Such applications typically build their user interfaces by creating and introducing graphical objects. Thereafter, they "listen" to events in the world so that when an event occurs, the application reacts according to control logic. Events can be user interaction signals, timers, object collisions, and so on.

DIVE offers a wide variety of programming interfaces and advocates a development strategy that mixes interfaces so as to benefit from the advantages of each programming interface and languages [2]. The scripting interface based on Tcl plays a prominent role by gluing together application components written in different languages such as C/C++ applications and plugins, or Java applications. Component integration is facilitated by a number of application-oriented facilities such as the ability to associate and distribute data with the environment or signalling objects using application semantic.

6.2. Software Design and Implementation

The chart, periscope and underwater scenes are generated by three DIVE processes that each runs on a separate PC. The processes share a virtual world that represents the ocean space where the submarine is located, and which is populated by a number of sound emitter objects representing surface and underwater vessels as well as some animals.

6.2.1. Application logic. The DIVE code that implements and drives the installation can be divided into two categories, Tcl scripts and C++ plugins. Scripts are mainly used to handle initialisation and application logic. For example, they handle user interactions, animations or the starting and stopping of sounds. Plugins implement more "heavy-weight services" that are used and controlled by the scripts. For instance, there is a Midi plugin that exports script commands to open a Midi device and for starting and stopping Midi notes. This plugin is loaded by one of the Tcl scripts that are executed by the chart process and is then used to control the playback of underwater sounds as the visitors interact with the GUI. Other plugins are used for rendering of the sea surface in the periscope, for communicating with the periscope hardware and updating the view as the periscope is turned, as well as the generation and display of the BTR tool in the chart GUI.

It is our experience that combining scripts with plugins is an efficient way to develop applications. By using plugins, new features and services can be introduced into DIVE in a flexible way, without the need to modify the core system. Each such plugin can be given an application neutral interface and a script interface. Application specific code is then moved to scripts, which are easier to maintain and modify.

6.2.2. Process Synchronisation. The scripts that are executed by each process are mainly devoted to tasks concerning the local station, e.g., updating the GUI, starting sounds, etc. However, they also contain code to synchronise the three processes at various points in time. When the processes are started by the museum staff each day, they wait for one another to start up and initialise before continuing to execute their main run-time logic. Synchronisation is achieved by each process creating a property (a DIVE object attribute that may be created during run-time) in a shared object and setting the value of this property to indicate its startup status. By examining these properties, a process is able to determine if others have initialised correctly. Whenever a property is created, updated or deleted, DIVE generates distributed events which may be caught and examined by any of the three processes. Scripts will therefore set up event subscriptions for property creation and update, which allows them to know exactly when the other processes are ready to start. A similar procedure is used when the installation is shut down each night. Properties are also used to synchronise the chart and underwater processes during the identification procedure.

We have found that using properties in this way makes it fairly easy to program various process synchronisation tasks. Although DIVE supports a message passing API which could have been used to send synchronisation messages between the processes, using properties is much simpler since the script does not need to handle message construction, error situations, remembering that a message with a certain content has been received, etc. Properties are persistent and distributed automatically by DIVE, which means that if process *A* creates a property before process *B* is started, process *B* will still see the property and its value once started since the property is part of the shared state. Also, since properties may be added to objects during run-time, it is easy to add application specific object attributes since no C or C++ code in the DIVE system needs to be changed and thus recompiled.

6.2.3. Subjective views. Since each scene is quite different graphically, the shared environment serves mainly as a common coordinate system where sound objects are positioned. The world itself has no common graphical representation since each process needs to present it in a different way. For instance, the chart station will display a sound emitter as a 2D icon while the periscope will present the same object using a 3D model. Graphical distinction is achieved by using DIVE's *holder* mechanism. A holder is a DIVE scene graph object that may contain the URL of a file that, once loaded, creates a sub-hierarchy of objects that are placed under the holder in the graph. By making the holder a local object, DIVE will distribute the holder, but not objects that are placed beneath it in the scene graph. Consequently each process will load the file contained in the holder locally, but the objects that are created as a result will be local to that process. By placing files with different content but identical names on the three PCs, and setting the holder URL to the name of these files, each process will load and create a different sub-hierarchy of objects under the holder. In the chart process, a sound emitter object will have a sub-hierarchy of objects that define a 2D icon while in the periscope the same object will have a sub-hierarchy that looks like a 3D ship.

Thus, all three processes share the non-graphical objects that define the position and orientation of the sound emitters, while each process decides individually how these objects look. If a shared object is moved or rotated, all processes will see the same change in the position or orientation attribute, but visitors will experience such a change differently depending on which graphical station they are using.

6.2.4. Sound Rendering. The sounds that are used in the installation are underwater recordings of various vessels and animals, provided by the Swedish navy. Although DIVE includes an audio module that supports 3D sound, it was not able to produce sound for the eight different channels that were necessary to drive the speakers in the installation. Instead, we chose to use an external audio software called Cubase together with a software sampler plugin. The reason for not going with a conventional sound card surround system was that we needed perfect real time control over eight independent but equal channels, and that we preferred uncompressed Hi-Fi sound. We also had experience in using Midi for generating sound via DIVE.

Cubase associates the received Midi notes to different sounds, which allows DIVE to start and stop sounds by producing a stream of Midi events. Cubase is started automatically by the chart initialisation scripts and terminates when the chart process is shut down in the evenings.

By using a different Midi channel for each speaker, the chart process is able to select the direction from where each sound emitter will be heard in the installation room. By determining the bearing of the sound emitter relative to the submarine in the virtual world, the chart process selects the most appropriate speaker and thus Midi channel for that particular sound. The Midi note corresponding to a particular sound is determined via a lookup in a hard coded table in the chart script that maps vessel types to Midi notes.

Even though the eight Midi channels allow us to produce sounds in different directions, we also wanted to change the sound volume to match the distance from the submarine to the sound emitter. Although the Midi standard includes a volume controller, it was of no use to us since it changes the volume on all sounds for a particular channel. That is, if several sounds where mapped to the same speaker, they would all be given the same volume which was unacceptable. Instead, we use the velocity parameter part of the Midi *noteOn* and *noteOff* events, and mapped velocity to volume in the software sampler used by Cubase. This allowed us to set different volumes for different sounds even though they may be placed in the same speaker.

Each speaker corresponds to a particular sector which has its origin in the submarine position in the virtual world. A fairly smooth cross-over of a emitter sound from one speaker to another is achieved by playing the sound in both speakers, and letting the volume in each speaker correspond to how close the vessel is to the line dividing the two sectors.

6.2.5. Video rendering. As described in Section 5, video is used to illustrate the work of a sonar operator during the simulated sound analysis. We chose not to use DIVE's built in video support since it mainly focuses on distribution of video images, which is not necessary in this case. Also, since the display of video in DIVE is based on 3D texturing, performance is rather limited, especially with high resolution video. The implementation uses an external video player which is started by DIVE when needed. Whenever the identification process is initiated, DIVE starts three different instances of the video player, each responsible for displaying one of the sequences. The video windows are positioned over the DIVE rendering window at certain pre-defined screen coordinates, and are displayed without decorations so that the video looks integrated in the 3D underwater scene. Knowing the length of each video sequence and using a timer, DIVE will terminate the players when the video has finished, causing the video windows to disappear.

6.2.6. Periscope. The custom made periscope (see Figure 1 and Figure 2) showing the ocean around the submarine is rigged with two Leine & Linde 10-bit optical angle encoders for measuring bearing and angle of elevation. Along with a number of additional switches these encoders are connected to a USB-based data acquisition system called Minilab 1008 (made by Measurement Computing), providing for a robust, low cost way of getting the periscope state into the managing DIVE plugin. The ocean graphics is displayed on an off-the-shelf 12 inches LCD display located in the back of the periscope. The algorithm for generating the wave simulation is similar to [8]. A patch of waves is generated and updated through the use of a Phillips spectrum and inverse FFT. This patch is then tiled to create the water surface within the frustum. This algorithm is implemented in a DIVE plugin that uses a number of hooks into the DIVE renderer for setup, adjustments, and rendering of the ocean surface. This plugin is also responsible for simulating curvature of the ocean surface. DIVE uses Cartesian coordinates and curvature is simulated by adding an offset to the position of surface vessels during rendering based on the distance to the vessel and the earth’s radius. A third plugin is responsible for managing the graphics on the periscope hud, e.g. current bearing, zoom, and elevation information.

6.2.7. Maintenance. The fact that the installation was going to be on display in museums for several years without any possibility for us taking part in its daily handling made it necessary to focus on maintainability and smooth operation. We decided early in the implementation phase that the system would need to be restarted each day as opposed to running continuously for long periods of time. This relieved us from having to consider issues like power failures but also meant that someone other than ourselves would have to be able to start and stop the system without too much trouble. Assuming that the museum staff were no computer experts, we came up with a solution that allows the whole system to be started by clicking on three desktop shortcuts, and to be stopped by pressing the Escape key on one of the PC keyboards. Not having the computers on the Internet ruled out the possibility for remote maintenance. The main reason for this design choice was the fact that some of the museums visited have rules against connecting publicly available PCs to their network.

Even though the stability of DIVE has increased over the years, we still had some doubts whether the system would be able to run continuously without problems. Tests that we did prior to the museum “release” indicated no major problems. However, as we assumed that problems could still occur, logging became an important issue. All three processes creates log files where the application scripts and plugins continuously log various aspects of their internal state, user interactions, etc. The idea is that if something does indeed go wrong, the log files will give us an understanding of the problems and hopefully the history of events that caused the problem to occur. Uniquely named log files are created each time the installation is started. This prevents new log files from overwriting old ones in case the museum staff restarts the system after a crash. This also keeps the size of the log files small, which has proven to be important since the museum staff can use a floppy disk to retrieve and send us the latest files. Had we used one log file that would have been appended with each run, the size would eventually have been in the order tens of megabytes.

The maintenance issue is also one reason why we chose to implement the application logic using scripts and plugins. If problems are discovered, we can easily modify scripts and mail them to the museum to replace the old ones. With sufficient guidance, the museum staff can even make minor script changes themselves.

7. Conclusion and Future Work

This paper has presented a virtual submarine museum installation and discussed details concerning its design and implementation. The installation has been on display for over a year in four different museums, and have been well received by visitors and staff alike. Informal results, gathered during visits to the museums have been very positive. Visitors seem to understand the correlation between the different displays. They are also very curious, especially children, who have been seen to work together at the different interactive stations to maximise the experience. During a 10 day period of the exhibition in Stockholm the number of visitors were counted to 9500. A more formal user study of the installation is currently being conducted.

The observation that visitors seem to intuitively know how to operate the periscope has led us to conduct experiments in using a periscope as a generic input and output device in other applications as well. An example is a telepresence application where users are able to control remote video cameras, allowing them to look around in distant locations by turning the periscope in various directions.

8. References

[1] Frécon E., Smith G., Steed A., Stenius M., and Ståhl O., “An Overview of the COVEN Platform”, *Presence: Teleoperators and Virtual Environments*, Vol. 10, No. 1, pp. 109-127, 2001.

[2] Frécon E., “DIVE: A generic tool for the deployment of shared virtual environments”, *Proceedings of IEEE ConTel’03*, Zagreb, Croatia, 2003.

[3] Ståhl O. and Wallberg A., “Using a Pond Metaphor for Information Visualisation and Exploration”, In Churchill, E., Snowdon, D. and Frécon, E. (Eds), *Inhabited Information Spaces*, Springer, pp. 51-68, 2004.

[4] Snowdon D., Greenhalgh C., and Benford S., “What You See is Not What I See: Subjectivity in Virtual Environments”, *Proceedings of Framework for Immersive Virtual Environments (FIVE'95)*, QMW University of London, UK, pp. 53-69, 1995.

[5] Smith G., “Cooperative Virtual Environments: lessons from 2D multi user interfaces”, *Proceedings of the Conference on Computer Supported Collaborative Work'96*, Boston, MA, USA, pp. 390-398, 1996.

[6] Taxén G., “Introducing Participatory Design in Museums”, *Proceedings of the eighth conference on Participatory design*, Toronto, Ontario, Canada, pp. 204-213, 2004.

[7] Frécon E. and Smith G., “Semantic Behaviours in Collaborative Virtual Environments”, *Proceedings of Virtual Environments'99 (EGVE'99)*, Vienna, Austria, pp. 95-104, 1999.

[8] Tessendorf, J., “Simulating Ocean Water”, In *Simulating Nature: Realistic and Interactive Techniques*. SIGGRAPH 2001, Course Notes 47, 2001.

[9] Mariani, J. and Rodden, T. (eds) “The Library Abstract eSCAPE Demonstrator”, *eSCAPE Deliverable 4.1*., Esprit Long Term Research Project 25377, ISBN 1-86220-079-3, pp. 75-145, 1999.

[10] Schwabe D., and Stenius M., “The Web Planetarium and Other Applications in the Extended Virtual Environment EVE”, *Spring Conference on Computer Graphics*, Budmerice, 2000.

[11] Wilde, D., Harris, E., Rogers, Y. And Randell. C., “The Periscope: supporting a computer enhanced field trip for children”, *Personal and Ubiquitous Computing*, Vol. 7, No. 3-4, pp. 227-233, 2003.

[12] Benford, S., Bowers, J., Chandler, P., Ciolfi, L., Fraser, M., Greenhalgh, C., Hall, T., Hellström, S.O., Izadi, S., Rodden, T., Schnädelbach, H. and I. Taylor, “Unearthing virtual history: using diverse interfaces to reveal hidden virtual worlds”, *Proceeding of Ubicomp*, Atlanta, Georgia, USA, 2001.

[13] Galani, A. and Chalmers, M. “Can you see me? Exploring co-visiting between physical and virtual visitors.” in *Proceedings of museums and the Web 2002*, Archives and museum informatics, Boston, Mass., pp. 31-40, 2002.

[14] Corcoran, F**.**, Demaine, J., Picard, M. Dicaire, L-G., and Taylor, J., “Inuit 3D – An Interactive Virtual 3D Web Exhibition”, *Proceedings of the Conference on Museums and the Web 2002*, Boston, MA. April 17-20, 2002.