Dr. Sabine Bergler

CINEMATIC PRESENCE SYSTEM

Mohamed Hachem

A thesis
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montréal, Québec, Canada

April 2007

Your file Votre référence
ISBN: 978-0-494-52317-9
Our file Notre référence
ISBN: 978-0-494-52317-9

Abstract

Cinematic Presence System

Mohamed Hachem

Over the last ten years, the computer has extended its functionalities from a business tool used by computer literates to a multimedia display that provides information, entertainment and communication to all people. This evolution calls for new interaction technologies. Indeed, the stationary interaction of the keyboard and the mouse limits the usability of the computer. Both the mouse and the keyboard were introduced to write documents or to execute commands, but what tools were created to permit a dynamic navigational system through a movie or real-time editing of a song ? Such an enhanced manipulation of media content requires another human-computer interactive approach. Few of the current methodologies of interaction have considered the possibilities of real time editing of moving image content or musical scores. Our contribution to more user-centric interactive systems is the Cinematic Presence System (CPS). The CPS is a non-tangible user interface that stresses the notion of Free Body Hypermedia Interaction (FBHI- Mohamed Hachem and Jean-Claude Bustros), which frees the user from following certain steps, wearing certain tools, or enduring any control over his/her senses while interacting with a multimedia display. CPS consists of a multimedia display that reacts to the user's movements. A user's interaction with the display allows the media content to be navigated and extracted in a non-linear fashion. Furthermore, this user-system interaction will promote a renewed, expanded cinema experience, where the user can interact dynamically with the image, the environment, and the screen.

Acknowledgments

I would like to forward special thank you to my dad and mom, Mr. Fouad Hachem Mrs. Atiqa Kassi Lahlou, for their love and support without forgetting that of my brother, Issam, and my lovely sister, Youssra. Also, I would like to present my gratitude toward Dr. Sabine Bergler for her supervision and orientation, Professor Jean-Claude Bustros for his guidance and friendship and Dr. Mudur for his vital directions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Professor Jean-Claude Bustros of the Melhoppenheim School of Cinema at Concordia University investigates strategies in Human/ screen media interactions for the creation of Augmented cinematic experiences some of his ideas have been implemented through the work reported in the present thesis and practically reflected in RAFFI's show . It comprises an interactive media set-up composed of multiple screens, ultrasonic sensors, and real-time control systems that permit an instantaneous response to the users motion by browsing and extracting precise segments of the audio-visual content. First,we describe the show. Second, we provide a description of the media content used in this project. The third section, we dedicate to the project physical composition and spatial display. Finally, we provide a description of the artistic and scientific drive.

1.1 The CPS Show

1.1.1 CPS, a New Expanded Cinema Alternative

The Cinematic Presence System (CPS) is a form of expanded cinema display that senses the users movement and changes its content accordingly. The idea requires bidirectional interactivity between the spectator and the show. In fact, while the user is affected by the media content, her/his movement simultaneously affects the display. Thus, CPS establishes a new form of cinema display called augmented cinematic experience (Jean-Claude Bustros), where the spectator directs the show. In the traditional cinematic experience, spectators are required to remain silent, stay in their seats, have little or no interaction with one another as their focus should be on show and the images flashing on the screen and need not to process the information themselves as it is presented in a fully linear show imposed by the film-maker. In concordance to the framework of the research conducted by Prof. Jean-Claude Bustros, CPS allows a break with this tradition of the passive cinematic spectator. CPS brings an alternative to the passive media show by providing an active expanded cinema display. With this display, the user is free to move, speak, and interact with the show content and the other spectators. In addition, the user decides the logical flow of the cinematic events through his/her movements, which definitively ends the linear characteristic of traditional shows and calls for customized multi-directional time-line shows. It also highlights the possibility of having an active spectator, who is ready to put effort into understanding and managing the display content for themselves, rather than becoming mere receivers of the film-makers vision.

1.1.2 CPS: Basic Interaction

The CPS display is a screen that is empowered with ultrasonic sensors and a software program that ensures real-time interactivity between the spectator and the show. The CPS display interacts with users at distances of up to twelve feet, having an angle of 15 degrees, and could cover an area of about 38 square feet. The area of interactivity, called the CPS field, could easily be expanded by using more powerful sensors. Within the CPS field, the user can affect the content of the display by moving forward, backward, to the left, or to the right. Forward and backward movements will generate commands that scan the material on display forward or backward. Scanning speed varies from image freeze and different degrees of slow motion to multiple levels of fast motion. Left and right moves result in jumping from one video or audio track to another. While these movements do not affect the content, they do affect the quality and characteristics of the display. Indeed, the DVD player allows us to store different audio and/or video streams of the same movie scene or song sequence. For example, a movie could be shot from multiple angles. As each stream contains a different viewing angle of the same scene, jumping from one stream to the other will result in changing the displays angle. In a similar example, one could jump from one soundtrack the real audio content to another containing a modified version of the actors voices etc. Most of the movements made by the spectator are accompanied by animations and text messages, which inform the spectator of the changes that have been occurring in the show or reflect a response to the spectators requests.

1.1.3 CPS: Advanced Interaction

CPS does not only respond to the users movement, but it is also capable of recording these changes while simultaneously allowing the user to enjoy continuous interactivity. For example, if the user has already watched part of the movie and left for a short period of time, he/she would be able to continue watching the remaining parts of the movie after returning. The program registers the movie frame number when the user leaves and recalls it when the user comes back. Another feature of the system is its empathy with the user. In fact, if the user spends more time with the system, she/he can enjoy privileged access to some of the movies content. The CPS calculates the amount of time spent by the user, and accordingly, uses this information later to open up special parts of the media material. This advanced interactivity is not straightforward. We know that the CPS counts four zones of interaction, each having its own show, spatial area, and level of detail. For instance, if the spectator stands close to the display, this action shows more interest in the content, which will be reflected in the program by producing more detailed displays and vice versa. The four shows contained in the four zones are part of one big show, but their content is independent and the only way to get to another zone's display material without moving is by watching the show. Ones continued interest, over time, permits the possibility of crossing to another zones content, while standing in place. If the spectator leaves one show zone for another or steps outside the interactive area of one zone, this action will be recorded. Registering the movement allows the user to continue watching what was remaining from the zone that she/he left; this continuity through the zones shows makes the user more comfortable dealing with a system that understands and empathizes with her/his actions. CPS is aware of the quantity of interaction with which its system should deal. In the case where no interaction is perceived for a long period of time, the system enters sleep mode, where sensor detection is still working, but the other functionalities are turned off and the show is limited to a simple default display that loops repeatedly. Once a user enters into the interaction field, the CPS system will detect the user, automatically turn on all its functionalities and interact with the spectator. On the other hand, if the level of interaction is too high and the system is

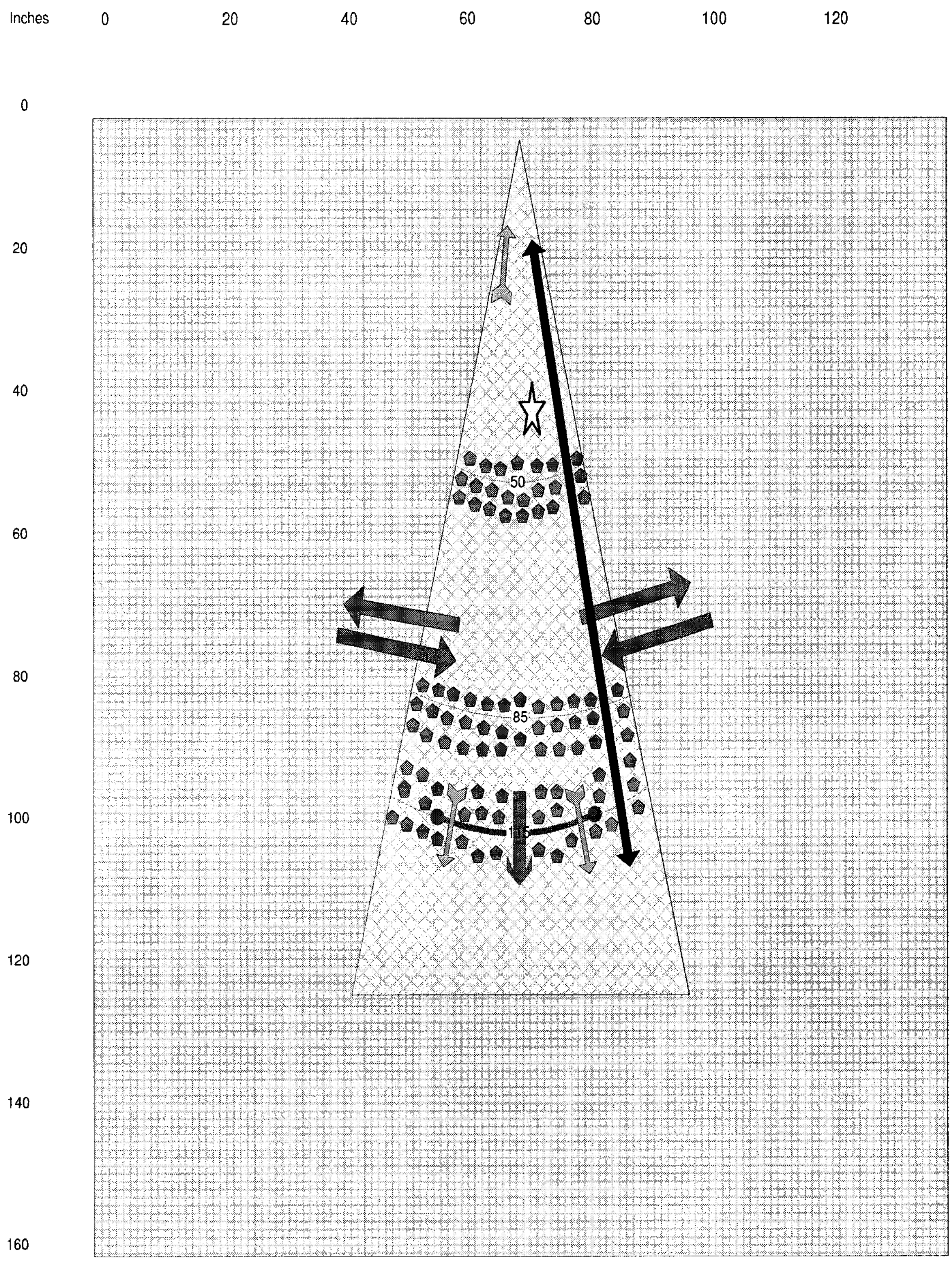
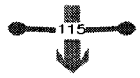


Figure 1: The user interaction field legend

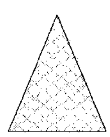
Key symbols description:



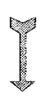
This symbol means that the spectator is using the door. Once the door exit is used everything is set to the default hence the default clip will show up right way.

This symbol means that the spectator is leaving the show from the side. It is referred to as walk way. A walk way always generates change on the played sequence of the show Hence the spectator will notice a change in the following : the view angle , the sound track, the motion speed, the graphical animation, and the text messages.

This symbol means that the spectator is coming back to the show from the side, in time less or equal to a variable "T" that is usually set to 30 seconds . It is referred to as walk back. A walk back always generates change on the played sequence of the show most of the time this change take the form of recovery from a "walk way" situation. Therefore, the spectator will notice a change in the following : the view angle , the sound track, the motion speed, the graphical animation, and the text messages.



This symbol is described as the field which is the area where the sensor is able to detect the spectator presence. It is about 130 inches long and 40 inches wide in the extremities



This symbol means that the user is going out of the field from the front or the back. This action is called an Open Exit. This means that the exit is legal and that no change will happen so far on the program but if no action is performed within a time "TR" the program will be set to default and the default clip will show up right way.

This symbol means that the user is moving to the front or to the back of the field. This Action is called a time fly. This means that movement of the user will allow a fly over the time line of the show backward or forward. The show could be divided into fixed portions of time zones called titles, chapters, or clips in this case we talk about static time fly otherwise if the show is set free without any divisions we talk about dynamic time fly hence the user can decide when to cut and when to concatenate her/his self made parts of the show.

This symbol means that the user is standing within space zone which is an area corresponding to a static division of the show a title, a chapter or a clip (static collection of frames) in the case of static fly time movements or to a dynamic division of the show in the case of dynamic time fly movements. It is called 4D flag, 4D refers to 4 dimensions the 3 formal space dimensions (X,Y,Z) and the time dimension. In fact, the field count seven main space/time zones four of them are stable zones which means that the are made to represent dynamic or static portion of the show instantaneously, the remaining three zones are perturbed which means that they are between the zones and hence the show is hesitating (flashing, jumping from sequence to the other, or looping the same sequence each few seconds, we see just brief scenes of the show sometimes just some static images) . Here is the summary of the space/time zones: 0 to 25 inches (stable), 25 to 35 inches (perturbed), 35 to 60 inches (stable) , 60 to 70 inches (perturbed), 70 to 100 inches (stable), 100 to 115 inches (perturbed), 115 > inches (stable, going out from the door exit).

This symbol means that the user is standing within the perturbed space/time zone which means that the user is standing between two stable zones and hence the show is hesitating we see just brief scenes of the show sometimes just some static images,

This symbol means that the user is standing on the borders of the perturbed space/time zone which means that the user is standing between a stable and a perturbed zone hence the show is extremely hesitating we see just brief scenes of the show sometimes just some static images,

Figure 2: The user interaction field legend

not able to respond to all the requests in a timely manner, it will prompt an image freeze, delete all the commands that are waiting to be processed and begin processing the newly issued commands. This behaviour prevents the system from crashing in the event that there are too many spectators moving quickly in all directions. The interaction process may seem complex, but it is easy to learn. Nevertheless, it calls for an active user, who will make an effort to get the intended results from the show. For more details about the user-system interaction, please refer to Figure 1 and 2.

1.1.4 CPS : Social Interaction

CPS is meant to be a social interactive system with more than one spectator at a time. It has no problem in dealing with multiple users. In fact, the ultrasonic sensors used by the system detect only the closest object; hence, the system will always deal with one user at a time. This default behaviour has opened the door for social interaction that moves beyond the intended bilateral interaction between the user and the system to create a trilateral interaction, Spectator-CPS-Spectator, called the SCPSS interaction. This trilateral interaction is caused when a spectator, who passes in front of another spectator, would be able to affect the show of the latter. Furthermore, the SCPSS interaction may prompt the second spectator to react by talking to the former user, by leaving the interaction field out of courtesy, or by advancing closer to gain control of the interaction process. Most important in all of these scenarios is the possibility of creating a customized show resulting from social interaction.

1.2 The Display Content

The material used in this show is a documentary about an HIV-positive AIDS Activist named Raffi, who struggles to increase the publics awareness about AIDS, about child abuse and its direct connection with the spread of AIDS in Africa and South Asia, and about the necessity of providing cheap medicines for infected people in developing nations. Raffi was filmed over a period of four years by Professor Jean-Claude Bustros. The documentary includes scenes from his daily life, including both private and public activities. Moreover, this documentary has all or nearly all of the information that a person would like to know about him. CPS makes it possible for a spectator to interact with Raffi, to follow him during his spare-time excursions, to attend his friendly gatherings, to enter the privacy of his home, or to listen to his intimate anecdotes. Simple movements allow the spectator to get into Raffi's world and browse moments over a four year period of his life.

1.3 The Projects Physical Composition and Spatial Display

The result of this project is contained in a dolly that holds five 15-inch LCD screens, five industrial Pioneer 7400 DVD players, one Sony projector, five Senix ultrasonic sensors, a customized computer, and eleven RS-232 serial ports. The number of screens and projections correspond to the number of displays needed for the show, which totals six displays, five screen displays and one main projection display, which allows the spectator to enjoy the content from a multitude of displays. The number of sensors corresponds to the number of LCD displays used in the set-up and the number of RS-232 cables correspond to the number of wires used to control all five sensors and six displays. The computer used in this project has been customized to respond to the shows requirements; hence, it includes eleven RS-232 input/output ports to control five sensors and six DVDs, a powerful processor of 4.2 GH to handle the data received by the five sensors, and 512 MB of RAM to have enough cash memory to operate in real-time mode. This packaging helped keep the project components easy to transport and display in different places and various locations. Indeed, the artist only has to place the screens and the sensors, then he/she can launch the application. The system is meant to work continuously for months due to the reliability of its components although most of the installations will only last for two or three weeks. The set-up would vary according to the place, the artist conception, the media content, and the expected public.

1.4 Artistic and Technical Drives

1.4.1 The Artistic Motivation

The artistic motivation was developed by Professor Jean-Claude Bustros as a reflection on the interaction of four primary elements: the image, the spectator, the environment, and the screen. These elements were separated in the past but are deeply overlapping in the current world. Society (the spectator) is affected by the movies (the image) and the movies (the image) are affected by society (the spectator). The image is within

Table 1: Comparing FBAR to Other Artistic Interactive Environments

The interaction	R*	FBAR	Augmented R*	Mediated R*	Virtual R*	Alternative R*
is in the real world	Yes	Yes	Yes	Yes	No	No
is in the virtual world	No	No	Yes	Yes	Yes	Yes
is under control	No	No	Yes	Yes	Yes	Yes
requires tools	No	No	Yes	Yes	Yes	Yes
requires training	No	No	Yes	Yes	Yes	Yes
is within an image	Yes	No	No	Yes	Yes	Yes
modifies cause/effect	No	No	No	No	No	Yes

R* refers to Reality

the screen (watching a movie on TV) and outside the screen (witnessing an event that would be reported in a movie). It is part of the environment (going out to watch a movie in theater), outside the environment (watching an imaginary story), and sometimes it is the environment (the world where we live will be represented in images to the future generations). The same thing could be said for the spectator who could represent the image (a movie reflecting society) or the reflection of an image (the society reflects the movies). This cinematographic thought was put into the spotlight by giving the user the chance to experiment direct interaction with the image, the environment, and the screen. This experience resulted in the creation of FBAR (Free Body Augmented Reality - Mohamed Hachem and Jean-Cluade Bustros) which represents a new milieu of artistic interaction that stands a part from most of the major artistic interactive environments, Table 1.

1.4.2 The Technical Drive

The computer has made dramatic incursions into our daily life from a business and industrial tool to the main medium of interacting with the digital world which engender multimedia, communication, information, and entertainment. The computer becomes a multimedia display through which we can achieve our work, watch our favorite musical clips, and talk with a friend on line. This evolution in the role of computers was wan watched by advances of the interaction technologies between the user and the computers. The interaction techniques remained stationary and focused on the use of a keyboard and a mouse. Tools that were invented decades ago to answer the need of writing a text on the screen or running a given program. We think that dealing with multimedia stations requires new technologies of interaction. We have opted for mobile interactions [Kallio *et al.*2003] and recognition based user interfaces. Our search has resulted in the implementation of the CPS, a user centric interactive media system. A system that calls for the fulfillment of the four points : First, the user is free to behave as he/she feels while dealing with the system, there should be no behavioral obligation from the user end; second, the user needs no tool and has to wear no gadget in order to communicate with the system; third, the user/system interaction is done in the real world in contrast to the immersive systems where the user is isolated from the real environment and the interaction is done within the virtual reality. Finally, the user's senses are not tricked by the system, no vision or audio illusion. These principles represent a reflection of our goal of promoting A Free Body Hypermedia Interaction (FBHI- Mohamed Hachem and Jean-Claude Bustros) where the user enjoys natural user centric interaction with the multimedia display.

Chapter 2

Background

2.1 Exploring Visual Languages for Real-Time Human Interaction

Visual language has become one of the main techniques used in recognition-based user interfaces following the advancement of image capturing and treatment technologies and the exponential improvement in computer processing power. Indeed, normal computers are currently able to treat more than 16-images-per-second, or work at the basic speed of a camera.

To highlight the application of visual languages in real-time human interactive interfaces, we reviewed two research projects completed at the University of El Pais Vaso, Spain. The aim of their research is exploring the power of visual languages, while using them for real-time systems. They focused on two main experiments, including: the control of remote robots and gesture-based music synthesis. We were interested in these projects because they touch on many subjects that are critical to our own project, such as their work with user interfaces, non-contact interaction, and the evaluation of an artistic expression tool.

2.1.1 First Experiment: The Remote Control of a Robot

This experiment consists of making a robot move according to the hand gestures made by a human controller. The control principle is very simple: a human controller holds two pieces of paper in his hands, each of a different colour. The computer calculates the colour centre of each paper at a rate of 25-times-per-second in order to achieve real-time interaction. This calculation creates a point moving from the first colour centre point to the second one, which is joined by a vector. It is the resulting vector that determines the robots direction and speed.

2.1.2 Second Experiment: Gesture and Movement-Based Music Synthesis

Here, we apply the same principle of interaction used in the remote robot control with little change. In fact, each colour now represents a musical instrument and the location of each colour centre determines the height and the duration of the musical note to be played.

2.1.3 First Observation: User Interface Requirements

Table 2: User Interface Requirements as Concluded by the University of El Pais Research (2003)

User Interface	Criteria
1	should work properly with any standard PC
2	should consume little CPU time
3	should be simple and intuitive to use

2.1.4 Second Observation: Advantages of Non-Contact Interaction

As you can see, this experiment and ours share the criterion of using a type of non-contact interaction, where the user is free from any physical contact with the machine and with any kind of effectors or sensors. This reduces the maintenance costs of the system by avoiding human interaction, and possible error, with mechanical devices[Ayala *et al.*2003].

2.1.5 Third Observation: Evaluating an Artistic Expression Tool

An artistic expression application is hard to evaluate as it can only be measured through the users level of satisfaction. Due to the difficulty of evaluating this tool, we have dedicated an entire section to public observations and feedback from our thesis project[Ayala *et al.*2003].

2.2 Interactive Displays

Interactivity with a display is one of the corner stones of the CPS system. It is also an immense area of research that applies different recognition technologies such as follows : (eye tracking[Boening *et al.*2006], pen-gestures[Mohamed *et al.*2006], tabletop and novel inputs / outputs [Morris2006], facial recognition [B.Reilly1998] and [Shinjo *et al.*2006] , touch screen[Wu *et al.*2006], vision detection and hand talk technology[Jun *et al.*2006] , speech[Shinjo *et al.*2006], head tracking[Liu *et al.*2003], etc.) and serves various sectors of the industry. In this section, we present a paper written by Dr. Gerald D. Morrisson. The purpose of the paper is to introduce the Digital Vision Touch system (DViT), which uses an innovative methodology of building interactive touchable displays. Our analysis of this paper focuses on the use of smart technologies and a comparison of the user-interaction characteristics of both DViT and CPS.

2.2.1 DViT

The Digital Vision Touch system (DViT) employs a touch screen that is able to deal with more than one user and more than one command at a time. It requires less processing power and less bandwidth, when compared to the previous generation of vision touch systems, due to the introduction of smart technologies and the use of a Digital Signal Processor (DSP). DSP is a local processor that is integrated within the sensor unit and allows for local treatment of the sensory data. The DSP transmits a higher level of data to the computer processor, which plays more of a logical role by analysing the users actions and issuing the reactionary commands to the display.

2.2.2 The Usage of Smart Technologies

The major weakness of visual sensors was the huge processing power that they involved. The introduction of smart sensors reduced this problem as they integrate embedded computational units, which process the collected sensory information locally and output the processed data to the computer. This capability has dramatically decreased the computational processing power required by the central computer unit. Though the CPS system is not using vision sensors, it uses semi-smart ultrasonic sensors. Indeed, the CPS sensors do not provide logically manageable data to the computer processing unit, but a string of characters that must be treated, in order to conclude the distance between the sensors and the moving object.

2.2.3 Vision Detection

Vision detection is done through the use of camera sensing. DViT uses four cameras placed on the corners of the screen, each covering a given area of the screen and allowing the system to detect of any contact with the screen. The detection is based on the change of ambient light over the screen as there will be less light when there is a contact with screen and more light there is no contact.

2.2.4 Comparison of the User Interaction Characteristics

DViT focuses on having natural human-computer interaction. Pointing and touching are normal human activities, so the system is attractive. However, natural interaction is a common point between DViT and CPS, albeit of a different type. Movement is also a natural human behaviour, but it is one that permits easier interaction for the public than is possible with the DViT system.

DViT and CPS are both real-time interactive systems, so their response time would play a principle role in providing user satisfaction. The two systems record low interaction latency, about 70 ms for the CPS, which is within the range of human-computer interaction optimal latency that is considered to be between 25 ms and 75 ms[I.MacKenzie and C.Ware1993]. At 250 ms, it takes the DViT system three times longer to react to the users actions when compared to the speed of the CPS system. Nevertheless, this response time is still acceptable because the user does not observe a system lag. The CPS not only has a faster response time, but also it is more robust when facing changes in the environments ambient light. Indeed, changes in the environments ambient light may have a negative impact on the camera interpretation of sensory data when, as in the case of DViT, using this type of detection. The data gathered through a cameras sensor is nothing more than a variation of ambient light on the screen. Hence, if the lighting around the screen changes for any reason, other then the user interaction with it, they may be erroneously interpreted as an instance of human interaction. The changes caused, for example, by shadows, bulb light or projector reflection, may have the aforementioned effect. Fortunately, this erroneous deduction of the camera sensor could be avoided by continuously updating the background image according to the algorithm proposed by [V.Cheng and N.Kehtarnavaz2000].

Despite its drawbacks, DViT allows for object identification and for a multiple-sized touch interfaces. To begin with, object identification is a strong feature of the DViT. It allows for differentiation between fingertips, styluses, or pointers, among other instruments. This capability allows DViT to react in various ways, according to the object in use. For instance, it could drop ink when the stylus is used or show a red point when the pointer is in use. In addition, DViT supports multiple-sized touch interfaces. In fact, the touch interface represents a camera projection on a screen. Since there is always a relation between the camera resolution and the screen pixel, changing the display size could be done by either increasing or decreasing the camera resolution. Hence, resizing the touch interface would be possible as long as the camera resolution is sufficient for pointer identification on the screen [D.Morrison2005].

2.3 Tele-Presence Immersive Display (Virtual Reality)

Tele-presence is a word that refers to being virtually present in a place through an immersive environment, where the user is entirely surrounded by virtual reality. The CPS system has little to do with immersive computing; nevertheless, it shares the same notion of incorporating a dynamic motion interaction with a display. The novel representing immersive display was developed in the vision and media computing laboratory at Nara Institute of Technology in Japan. This section will introduce the Nara tele-presence system and compare it with the CPS system.

2.3.1 The Nara Tele-presence System

The Nara tele-presence system is an immersive computing environment that allows the user to make virtual visits to distant places, while walking on a treadmill and being surrounded by three slanted rear-projection screens, displaying the front, left, and right views of the site. The user can only walk forward and their motion is reported by two sensors fixed on their legs. The sensory information is used to make the display change at the same speed of the user. Since the system mimics the speed of the user, he/she has the perception of walking through the surrounding virtual environment[IKEDA *et al.*2003].

2.3.2 Comparing the Two Systems

The Nara tele-presence system is capable of immersing the user in a completely different environment. However, the provided interaction is unnatural because the user can move only in one direction and the sensors report only the movement of the lower parts of the body. CPS was not made to be an immersive installation. Hence, it is much lighter in terms of equipment than the Nara system, which requires 14 PCs, 12 projectors, a treadmill, two portable sensors etc. CPS is also more natural because it allows the user to move freely in all directions and takes into account the motion of the user‘s entire body.

2.4 LifeClipper (Augmented Reality)

LifeClipper is an outdoor art project conducted by a team of researchers from the University of Basel in Switzerland. The project aims at exploring the boundaries of augmented reality through the application of ambient computing. In fact, the project consists of creating an augmented-reality outdoor tour, where the users movements are tracked and the users senses are augmented by the presence of extra images, art graphics, sounds, texts, and special effects. This augmentation of the environment is achieved through the use of a transparent head mounted display. The coming subsections will give a brief description of the LifeClipper project and compare its characteristics to those of the CPS.

2.4.1 LifeClipper

LifeClipper is a system that allows for augmented-reality tours thorough which the virtual merges with the real and in which the user experiences various augmented effects stretched over a diverse range artistic disciplines. First, the visitor experiences cinematic immersion, when an old cemetery is transformed into a horror movie by adding frightening sounds, whispers and breathing, accompanied by blurs when the visitor‘s head is moved. Furthermore, the background turns blue-grey and photos of famous people buried in the cemetery appear according to the visitor‘s position. Second, poetry is read, while the user is walking by the riverside. The poetry is supported by visual effects, which overlay the immediate field of experience. Third, art graphics would be used to signal different orders and warnings such as: Stop, turn left, turn right, leaving the area, watch out, watch your step, etc. Fourth, the display characteristics are modified to the level of transparency, brightness, and contrast. Moreover, the vision speed varies between the minimum and maximum frame rate. Fifth, historical documentary video overlays the scene when the visitor reaches Basel‘s old city wall. Sixth, actors and objects are placed in both real and virtual worlds to irritate the visitors by making them unsure about the nature of these appearances (real or not) and dissolve the boundaries between the real and the virtual. Seventh, images are displayed as flashes in certain parts of the head-mounted display screen. All of these effects, and many others, make the virtual and the real combine into their mutual interplaying and complementary parts[Torpus and Buhlmann2005].

2.4.2 LifeClippers Technical Settings

The user wears a video head-mounted display. The video camera records at a resolution of 320*240 pixels and is blown up to 800*600 pixels, its viewing angle is 30and it is rendered, in addition to the audio sound, by MAX/MSP/Jitter, which is a real-time editing software. The visitor‘s movement is reported by a hybrid system using GPS technology to compute the users position and movement and a compass to determine the users direction of movement. The visitors walking behaviour is tracked by pressure sensors incorporated in the soles of his/her shoes[Torpus and Buhlmann2005].

2.4.3 LifeClipper Vs. CPS

It is clear that both the LifeClipper and CPS systems have a lot in common. They both augment the show by adding video and audio effects, art graphics, text, and signals to the original scene. However, they differ in the way in which they interact with the user and the users environment. In fact, the LifeClipper does not incorporate a strong notion of empathy towards the user. Spending ten or fifty minutes by the old city wall is not going to change the systems interactive behavior with the user. However, the time spent with the show matters when utilizing the CPS system, which will allow the user who spends more time within a particular zone to see privileged material. LifeClipper limits the users freedom by issuing signal orders to the user (stop, move, turn, etc.), while the CPS uses the signal as auxiliary informative data, which serves, rather than controls, the user. Most importantly, LifeClipper is location dependent because the content of the show could be applied only in Basel. There is no decoupling between the content and the environment, while the CPS is a location independent system that can be shipped and displayed anywhere around the globe. On one hand, LifeClipper offers the possibility of interacting with the system over large distances due to its use of GPS technology, and it has the capability of memorizing the interaction by allowing the user to take pictures of the scenes she/he sees by pressing a button located on the head-mounted display. On the other hand, CPS interaction is restricted to a dozen feet and can memorize multiple points at which a user was last interacting with the system.

2.5 Interactive Mediated Reality (Mediated Reality)

Interactive-mediated reality reflects the concept of modifying the appearance and geometry of real objects on the fly. This modification is achieved by adding virtual graphics to the real scene, the resulting mediated scene is visible through a head-mounted display [Bimber *et al.*2002]. The display system counts four main phases of scene construction: first, the system reconstructs the 3D vision of the real environment. Second, it adds virtual objects to the scene. Third, it registers and calibrates the changes within the real world environment. Finally, it displays the newly constructed scene to the user. This section will present an interactive-mediated reality project made through the collaboration of researchers from Europe and New Zealand. The project represents an attempt to create a mediated-reality studio, where the artist dynamically combines both real and virtual objects in order to obtain a real-time, mediated scenery of the real environment. Indeed, this technology is meant to allow a dynamic change of the real environments scenery by adding virtual artistic objects (drawings, colours, textures, etc.) to it. For instance, movie scenes are modified during the post-production process in which the artist adds special effects and chooses the scenes, which will be retained for the movie. However, giving the film director the possibility of watching the special effects while the movie is being shot would help create better effects and help choose the right scenarios. The following subsections will introduce the project and compare it to CPS.

2.5.1 Virtual Studio

The Virtual Studio project creates a mediated-reality studio, which contains the elements found in a real studio workspace, such as tools, a palette, and a scratch area. The workspace is used to hold the content, the tools are used to modify the content, the palette permits access to the virtual design elements (colours, textures, options, etc), and the scratch area is used to combine the virtual and real objects. This system allows restricted design operations (glue, paint, assemble, cut, etc.) to be supported by further editing operations (copy, paste, undo, multi-function tools, etc.). It also allows for the modification of the appearance of real elements by adding virtual elements on the fly to the scene. Wearing a head-mounted display and working with a tracked brush and tool palette is required for this system to work properly[Grasset *et al.*2005].

2.5.2 Hybrid Technology

The system was inspired by previous work. Indeed, it was developed on the top of the Studierstube framework [Schmalstieg *et al.*2002], the painting algorithm was based on the ArtNova approach [Foskey *et al.*2002], the lighting adjustment is done virtually or manually as mentioned by [Debevec1998], the texture orientations are fixed interactively by a plane dragging widget similar to[Agrawala *et al.*1995]. The texture patch could be real or it could be taken from an alternative source, such as a book or magazines as proposed by [Ryokai *et al.*2004], and labels could be associated with different objects similar to [Kalnins *et al.*2002].

2.5.3 CPS Vs. the Virtual Studio

CPS and the virtual studio project provide the dynamic addition of graphical objects and labels to the original scene. They also allow for an interactive change of color and lighting. If they share some characteristics, these two systems differs in many others. On one hand, the CPS allows an interactive modification of sound, video, and motion speed (fast or slow forward/backward). On the other hand, it is unable to provide some of the elementary options of the virtual studio, such as registering the created scenes or using some basic editing operations, like copy and paste.

2.6 Intelligent Virtual Environment (Alternative Reality)

An intelligent virtual environment represents artificial worlds that react dynamically and independently to any external physical intervention [Aylett and Cavazza2001]. The use of these environments is popular in virtual reality art productions [Moser1996] [Grau2003].This section relates to Ego.geo.Graphies, a virtual reality artistic work by Alok Nandi. Though it is an artistic work, Ego.geo.Graphies required the collaboration of teams of engineers and computer science researchers from four European and North American universities.

The work of Ego.geo.Graphies consists of creating an alternative virtual environment, where the user can interact with a fantasy world in which reasoning and causality are different than they are in the real world. In fact, while we normally explain the natural worlds phenomena in terms of certain logical causality, this system tries to alternate this causality by replacing the conventional physics laws with some alternative laws. The new laws are based on another gender of causality and which reflect the interactivity between the user and the system[Cavazza *et al.*2004]. For example, the movement of some virtual objects may increase their virtual physical density and may cause them to blow up. In fact, The system uses the users interactions with the worlds objects as an indicator of causality. It is the users interactions and experiences in the milieu that decides which physics laws would be applied in the environment. At the same time, it determines how the user affects the laws of physics and how they would be affected by them in a virtual reality artistic environment. The coming subsections introduce the Ego.geo.Graphies system, highlight its user-system interaction procedure, and compare Ego.geo.Graphies to CPS.

2.6.1 Ego.geo.Graphies

Ego.geo.Graphies is a cave environment, or a closed room, serving as the milieu of interaction between the user and virtual environment. It is created by four rear projection screens of 3*3 meters each, which are placed to the right, to the left, in front of the user, and on the floor. The environment could be explored by an audience of up to four people and reflects a virtual world of 17 000 meters square. The screens support stereographic display by using two computers per screen, one for each eye, and having a rendering rate of 60 frames per second. The system users must wear shutter glasses with lenses, which alternate at 120 frames per second. This speed equals twice the rendering rate required for normal vision. So each eye receives the view it is supposed to have at 60 frames per second. The users are constantly tracked by the system through a head tracking mechanism implemented within the user-mounted display. Tracking the head allows for the generation of a stable view of the virtual world by adjusting and calibrating the scenes according to the heads movement. Tracking other parts of the body determines the interaction of the users with the milieu. In regard to the software architecture, the system has adopted the notion of dynamic event model interaction, where the users actions are treated as events by the system[Jacobson and Hwang2002].

The Ego.geo.Graphies event system is built over the graphic engine and is responsible for handling the behaviours that take place in the virtual world. The event detection procedure does not allow for dynamic alteration of the cause/effect relations because they were hard-coded associations. In fact, the interaction process was coded as a succession of if-else statements, which restricted the combination of actions and the possibility that they could lead to a different outcome. To overcome this limitation, the system integrated a rule-based engine called the causal engine. This engine supports dynamic identification and modification of the actions in order to generate alternative effects. Since the behaviour of the object in the virtual environment is controlled by the event system, it is easy to allocate an arbitrary outcome to a given action. The causal engine receives low level events from the event system, freezes these actions and substitutes them for new outcomes. In other words, the system has a conventional static scenario that is followed in the case of a projected interaction and a rule-based engine, which stops this scenario and fires a combination of rules. In the case of unprojected interaction, this substitution is done by macro-operators (knowledge structures). These macro-operators are used to modify the effect part of the causal engine representation so as to generate an alternative outcome. The alteration of actions is done in concordance with the level of disruption, which is updated every 25 seconds. Moreover, the level of disruption reflects the user-creature proximity (the user proximity and usage of the artificial world objects) and the user- agitation (an application of the user movement). Furthermore, the level of disruption is responsible for choosing the type of macro-operator to be used by the causal system. Therefore, the user interaction determines the macro-operators to be used, which results in the alteration of the virtual environments cause/effect relations[Cavazza *et al.*2005].

2.6.2 User Interaction

Ego.geo.Graphies is a fantasy world of spheres. The user can push and move the spheres all around the environment. When the spheres are dense, they enter into collisions. The results of their collisions correspond to their level of density which is affected by the users historical interaction with the virtual world. For example, the spheres collision will lead to both a merger and the creation of one sphere, if the user has not spent much time in the environment, or its explosion, if the user has extensively displaced the spheres.

2.6.3 Ego.geo.Graphies Vs. CPS

The commonality between the Ego.geo.Graphies and the CPS system is the implementation of a user interaction factor that modifies the normal behavior of the system. For instance, CPS provides privileged content access to users, who invest a given amount of time with the show. It also keeps a short history of the users interaction, so the user would not have to watch scenes that he/she watched before, unless he/she wanted to see them again. The same behavior is observed in the Ego.geo.Graphies system, which reacts according to the time spent in proximity with the worlds creatures and the history of the users interaction with the environment.

CPS has the disadvantage of not having a causal engine. Indeed, the implementation of a rule-based engine has added more dynamism to the user-system interaction. A causal engine allows reactions to be modified on the fly, so the user can enjoy richer, more diversified, and more customized interactions. Nevertheless, the causal engine of Ego.geo.Graphies was built over a lower level event system, the same could be made for the CPS, which lacks the integration of a causal engine, but still has the advantage of being able to interact with more than four users at a time and requires much less processing power than does Ego.geo.Graphies.

Chapter 3

Project Development and Implementation

We wanted to build a real-time system that is able to detect spectator movement through the use of ultrasonic sensory data. This data determines, which part of the media content will be extracted and played on the display screen. Conceptually the system counts two main physical parts and two main intangible technological components. The physical elements are a DVD player and ultrasonic sensors, while the intangible components are the communication technology and the software, which manages the system. These four components are fully described in this chapter.

3.1 The CPS System

CPS is a form of expanded cinema that allows a movement interactive spectator-display relationship. Indeed, the spectators movements are detected by ultrasonic sensors, which send the collected sensory data as a collection of ASCII characters to a computer. Next, the computer analyzes the characters according to a predefined methodology of responses, where every movement corresponds to a particular DVD player command. The system could include more than one display, while the actual setting we used includes six displays, six DVD players, five sensors, and a central computer unit that manages the multi-display CPS system.

3.2 The CPS Functionality

CPS is able to provide real-time movement interactive shows, where the users movements dictate the part of the show to be put on display. The spectators moves also allow the addition of animations and text messages to the shows original material. Another characteristic of the CPS is its capability to change the shows volume level, scanning speed, etc. Moreover, CPS is conceived to work continuously as much as the managing computer unit is functional.

3.3 The CPS Distinctive Criteria

CPS is a simple system that includes a display, sensors, and a computer program. These three elements could be easily put together into one display and result in an CPS display that works with any display screen and it needs no expertise or user manual in order to be controlled. Indeed, all that is necessary is a few actions from the user. CPS is a highly portable, adaptable, and robust system. First, CPS is portable because the displays used could be LCD screens, projectors, computer screens, laptop screens, etc. In other words, a screen of any size could be used to present the content of a CPS show. Second, CPS technology is adaptable. In fact, it could use any type of future displays and sets of sensors. Moreover, at 46KB, its tiny object-oriented program could easily be extended or refurbished. Third, CPS is a robust system because it has the capability to recover and avoid crashes. Indeed, in case of a crash or non-response, the system relaunches its applications after 30 seconds. This recovery action is crucial to assuring the continuity of the show without any human intervention. Moreover, the real-time interactivity is assured due to the low processing time that the ultrasonic sensory data requires. In fact, due to its low processing time, CPS ranks as a better real-time application in comparison to other systems, which use image or sound processing, especially when deal with many users simultaneously.

3.4 The Pioneer 7400 Programmable DVD Player

To run CPS, we need powerful DVD players that will be able to play non-stop for long periods of time. Furthermore, we needed this media player to be programmable and to allow for a good level of control over the media content. The Pioneer 7400 industrial

DVD is robust, programmable and fires control over the media content.

3.4.1 Understanding the Functionalities

The Pioneer 7400 is complex as it contains an overwhelming number of functions. Our main focus was to understand the control possibilities provided by this DVD player. After careful study of over one hundred functions, we have resumed implementing 30 control applications, see the coming subsection, and the implementation was done in iterative phases, function by function.

3.4.2 The Control Class

```
Class Control
{
private:
static int cr;// cr stands for character return
void sendInt(RS X,int number);
//a recursive function that takes an int as an input, translates each
//digit of it into ASCII code, and transmits it to the serial line
int CharToint(char ch);//converts from char to ASCII code

  public:
void printcharacter (RS X,int line,char* message);// use to print message on a screen
void pause(RS X);//pause DVD player
void playtitle(RS X,int number);// play title number
void playchapter(RS X,int number);//play chapter number
void play(RS X);//play DVD player
void playframe(RS X,int t,int a,int b,int c,int d,int e,int f);//play frame number
void Angle(RS X,int number);// play different angle of the movie by changing the video track
void clearlines (RS X);//clear all lines on the display screen
void BringMenuRoot (RS X);//go to the main menu
void BringMenuTitle (RS X);// go to the menu title
void SpeedForward (RS X,int speed);//set the speed Forward
void SpeedBackward (RS X,int speed);//set the speed Backward
void AudioChannel (RS X,int Channel);// choose one audio Channel over eight, zero defines mute
```

```
void AudioOutput (RS X,int output);// set the audio output in a range from 0 to 7
void Subtitle (RS X,int language);// set subtitling language to any one of up to 32 languages
void VideoOn (RS X);// turn the video display on
void VideoOff (RS X);// turn the video display off
void SetStack (RS X,int group);// set the command stack group
void BlackBoardDisplay (RS X,int step);// set the blackboard display nature
void ClearBlackBoard(RS X);// clear the blackboard display
void BlackboardDownload (RS X);// download the blackboard content
void BlackboardUpload (RS X);// upload the blackboard content
void KeyLock(RS X,int number);// lock the DVD control buttons
void displayGroup (RS X,int group);// set the group to be displayed
void AutoDisplay(RS X,int group, int animation);// set the animation and the group to be displayed
void Time(RS X,int time);// prompt DVD to go to a certain time
void ScanForward (RS X);//scan Forward
void ScanBackward (RS X);//scan Backward
void ScanStop(RS X);//player resets to the normal playback, after a scan forward or backward
};

int Control::cr = 13; // 13 is the ASCII code for character return
```

3.4.3 Implementing the DVD Control Functions

There is little information about how to implement the DVD functions. In fact, The DVD programming manual was restricted to simple examples of creating simple serial commands. Hence, every control combination that included many commands was a real challenge. More importantly, there was no way of figuring out the best manner in which to implement these commands. To face this obstacle, we have decided to use an OO programming language, which made our implementation easy to both encapsulate and assemble. We have also conducted a series of experimental combinations of commands in order to determine those that are feasible to use. See the coming subsection, which includes the C++ implementation of some of the DVD control functions.

3.4.4 Examples of Functions Holding Combined Serial Commands

```
//************************************ Set playing time
void Control::Time(RS X,int time)
{
X.Transmit(CharToint('T'));// TM stands for search
X.Transmit(CharToint('M'));
sendInt(X,time);
// a recursive function that takes time as an input translates each
// digit into ASCII code, and transmits it to the serial line

    X.Transmit(CharToint('S'));// SE stands for search
X.Transmit(CharToint('E'));
X.Transmit(CharToint('P'));// PL stands for play
X.Transmit(CharToint('L'));
X.Transmit(cr);//cr represents the character return used to mark the end of the function
}

    //************************************ Set Display group
void Control::displayGroup (RS X,int group)
{
group=group+48;
// set the group number +48 is used to change number from int to ASCII
// There are 300 groups, and you could choose any number between 1 to 300.
X.Transmit(group);// send the number of the group to be displayed
X.Transmit(CharToint('G'));// GP stands for group
X.Transmit(CharToint('P'));
X.Transmit(CharToint('0'));
/*****
'0' stands for auto-display, specifying another number would lead to the the display of that character or drawing holding the specified number within the declared group. For example, writing 12 instead of 0 will lead to the display of the character or the drawing assigned with the number 12 in the chosen group. It is worth knowing that every group has up to 300 characters or drawings
*****/
```

```
X.Transmit(CharToint('V'));// VS stands for video display
X.Transmit(CharToint('S'));
X.Transmit(cr);// cr represents the character return used to mark the end of the function
}

//************************************** Speed Forward
void Control::SpeedForward (RS X,int speed)
{
/*************************************************
The default speed value is 30 and it counts for half of the maximum speed. Other possible values
are the following: 15 for half the default value
7 for a quarter of the default value
4 for 1/8 of the default value
1 to go frame by frame
*************************************************/

speed=speed+48;//set the speed number +48 is used to change the number from int to ASCII
X.Transmit(speed);// sends the speed to be used
X.Transmit(CharToint('S'));// SP stands for speed
X.Transmit(CharToint('P'));
X.Transmit(CharToint('M'));// M stands multi-speed in ASCII
X.Transmit(CharToint('F'));// F stands for forward
X.Transmit(cr);// cr represent the character return used to mark the end of the function
}
```

3.5 Serial Communication

The Pioneer 7400 DVD player can communicate with a computer through an RS-232 Interface. Hence, the serial commands can be exchanged between the DVD player and the computer in the form of ASCII characters. In fact, the DVD defines a code of communication, where each combination of letters leads to a given command. For example, sending the ASCII codes of the letter P followed by the ASCII codes of

the letter L is interpreted as the command play by the DVD as PL stands for the two first letters of the word play. At the beginning, we implemented a basic RS-232 function that sends and receives ASCII characters; this function was crucial in testing our combinations of serial commands. Once most of these combinations were proved to be functional, we decided to build a solid RS-232 communication structure, which will decouple the communication interface from the control system by building separate classes dedicated to the communication of commands. In other terms, we have thought about separating the information exchanged from the control by placing each of them on a different layer, which will allow better control and independent communication. Therefore, we have built a serial class RS dedicated to encapsulating all of the communication data transfers, see the coming subsection, and kept the controls within the control class.

3.5.1 The Serial Communication Class

```
class RS
{
private:
    // communication port variables
HANDLE handle;
BOOL opened;
    public:
    BOOL TwoDOpen(int Port, int Baud);
/*****************************************************
Goal: Opens a serial port with two digits
Parameters: Port stands for a serial port ranging from 10 to 99
Baud stands for the baudrate to be used 4800, 9600, 19200, etc ...
Returns: True, if the serial port is opened
False, if the serial port is not opened
***********************************************************/
    BOOL OneDOpen(int Port, int Baud);
/*****************************************************
Goal: Opens a serial port with one digit
Parameters: Port stands for serial port ranging from 1 to 9
```

Baud stands for the baudrate to be used 4800, 9600, 19200, etc ...

Returns: True, if the serial port is opened

False, if the serial port is not opened

```
*************************************************************/

BOOL Close(void);

/*****************************************************

Goal: Closes a serial port

Parameters: None

Returns: True, if the serial port is closed

False, if the serial port is not closed

*************************************************************/

BOOL Transmit(unsigned char ChByte);

/*****************************************************

Goal: Transmits a byte

Parameters: ChByte stands for the byte to be sent

Returns: True, if the serial port is sent

False, if the serial port is not sent

*************************************************************/

int GetSensorData();

/*****************************************************

Goal: Receives bytes from the sensors and interprets them as a distance in inches

Parameters: None

Returns: The distance in inches as it has been submitted by the sensors

*************************************************************/

};
```

3.6 The Senix Ultra-Sonic Sensors

3.6.1 Looking for a Suitable and Affordable Sensor

We had to look for an affordable sensor that would allow instant feedback from the users movements from a medium distance range of up to 10 feet. The search was huge because we had to choose among thousands of sensors. Our selection process

was based on two main phases. First, we had to decide on the type of sensor we were willing to use: a visual sensor, a light sensor, a heat sensor, etc. We had to decide which type would best serve our project. After a careful study of the particularities of each technology, we narrowed our set to two main sensor techniques: visual and ultrasonic, because we wanted to build a real-time system that delivers instantaneous responses to the user's actions. This would be difficult to achieve if we had to treat every image captured by a visual sensor using a normal computer. More importantly, we were not interested in having all the details that an image might provide. For example, a users physical aspects, such as their clothes or hair color, were not important for us. The only information we needed was the presence of a user at a given distance from the sensor. Our goal could be achieved through an ultrasonic sensor. Second, we had to decide which ultrasonic sensor would be the most suitable for us. Our decision was made clear after a detailed description of the characteristics of the sensor we were looking for and a definition of the amount of resources with which we had at our disposal. We were looking for small ultrasonic sensors of no more than five inches long with good timely responses of less than 0.1 second and the possibility of communication with computer programs. After a careful selection, we chose the Senix Ultra-Sonic Sensor number 151482.

3.6.2 The Senix Ultrasonic Sensor Number 151482

The 151482 Senix Ultrasonic Sensor measures the distance between the sensors and the target object by sending a pulse and measuring the time of its echo. This time is related to the speed of the pulse. Once calculated, the time is transmitted as binary data on an RS 232 output, where each count of the binary data corresponds to 0.13536 inches (it was decided, so by the manufacturer). The sensor provides data in a range that goes up to 10 feet, has a beam angle of 15 degrees (see figure below), and updates the measured distance at an interval of 50 milliseconds. The sensor performance could be affected by the position of the user. In fact, the sensor detects the users distance by calculating the time taken for the ultrasonic signal to reach the user and come back to the sensor. However, if the user stands with a certain angle to the sensor, the signals may get reflected away from the sensor, which will prevent the sensor from detecting the user (see figure below). Nevertheless, the case in which the user is literally undetected is unlikely to happen because the human body represents a

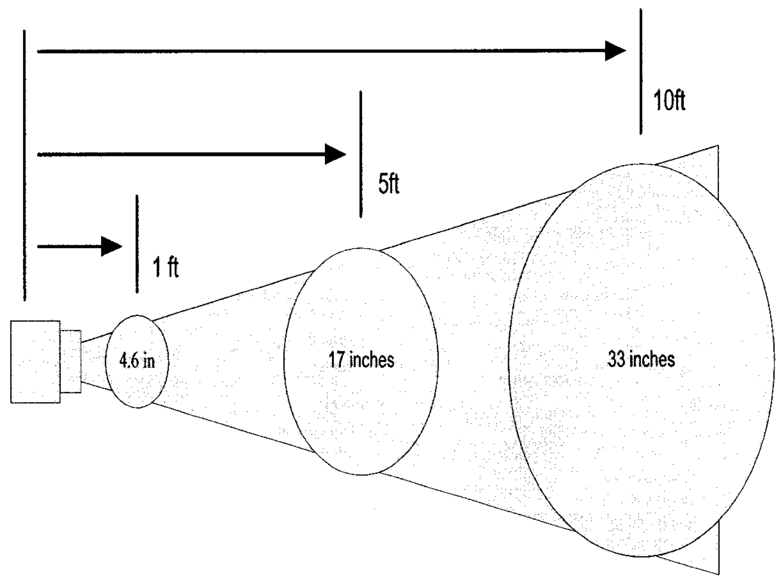


Figure 3: The Sensor Beams Range

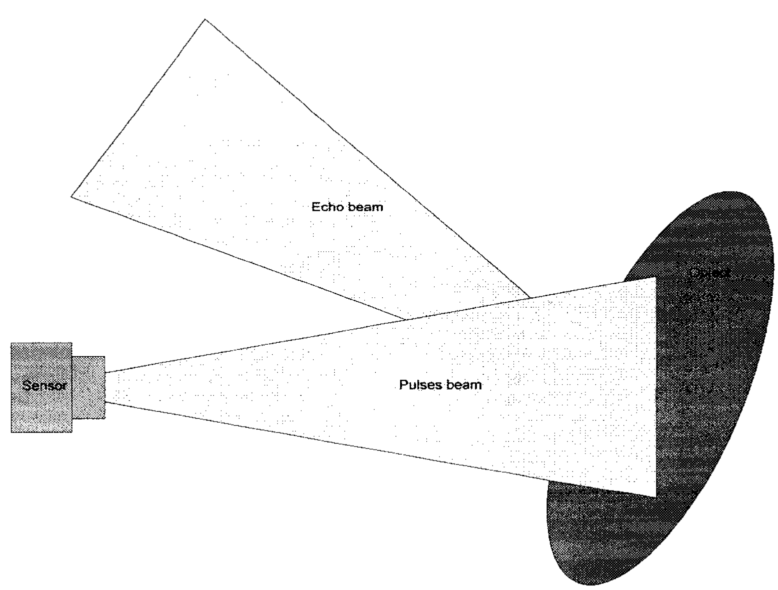


Figure 4: The Sensor Beams Reflection

bumped surface. In other words, it is a surface where the beam reflection takes multiple directions and results in sending some signals back to the sensor. Another weakness of the sensor is its vulnerability to the change in the environmental temperature due to the laws of physics. In fact, the beams speed of propagation changes by 1changes by 10 degrees. Hence, the time measurement taken by a signal over the same distance, but at different temperatures, would differ. Therefore, different distance measurements would be detected under the condition of different temperatures. Nevertheless, we are not concerned by this problem because we do not require detailed measurements during the interaction process between the sensor and the target user.

3.6.3 Implementation of the GetSensorData() function

```
int RS::GetSensorData()
{
BOOL bReadStat;
DWORD iBytesRead;
    double x=0;
double y=0;
double inches=0;
int intinches=0;
unsigned char a;
unsigned char b;
unsigned char c;
if ( !opened ||handel == NULL)return(FALSE);
    while(!ReadFile(handel,a, 1, iBytesRead, NULL)||((a! = 'U')&&(a! = 'u')));
    bReadStat = ReadFile(handel,b, 1, iBytesRead, NULL);
bReadStat = ReadFile(handel,c, 1, iBytesRead, NULL);
    x=static_cast(b);
y=static_cast(c);
    x=(y*256)+x;
    inches= x*0.013536;
    cout<<" The distance in inches is the following :"<< inches<<endl;
intinches=static_cast<int>(inches);
    // check for no bytes read or IO error
```

```
if ( !bReadStat ||GetLastError() == ERROR_IO_PENDING)return(FALSE);
if(iBytesRead == 0)return(FALSE);
    return (intinches);
}
```

3.7 Software Architecture

3.7.1 Static View

Class Diagram

The CPS system includes three main classes: Control, RS, and Manager. The control class is responsible for issuing serial commands to the DVD player, such as play, stop, scan, etc. The role of the RS class is opening and closing the serial ports of both the DVD player and the sensor and reading the sensory data updates. The manager class represents a logical link between the RS and the control classes as it issues commands to open the serial ports for both the DVD player and the sensor to the RS class, analyzes the sensory data issued by the RS class, performs timely, logical operations on it, and makes calls to the control class subroutines accordingly, see Figure 5.

3.7.2 Design View

Internal Structure

The CPS system is composed of three principal internal : the sensor, the manager unit, and the DVD player. The sensor receives ultrasonic-waves and translates them into RS-232 standard output. The DVD player receives RS-232 serial commands and issues audio and video output signals. The manager unit receives RS-232 serial commands input, analyzes them, and issues RS-232 serial outputs, see Figure 6.

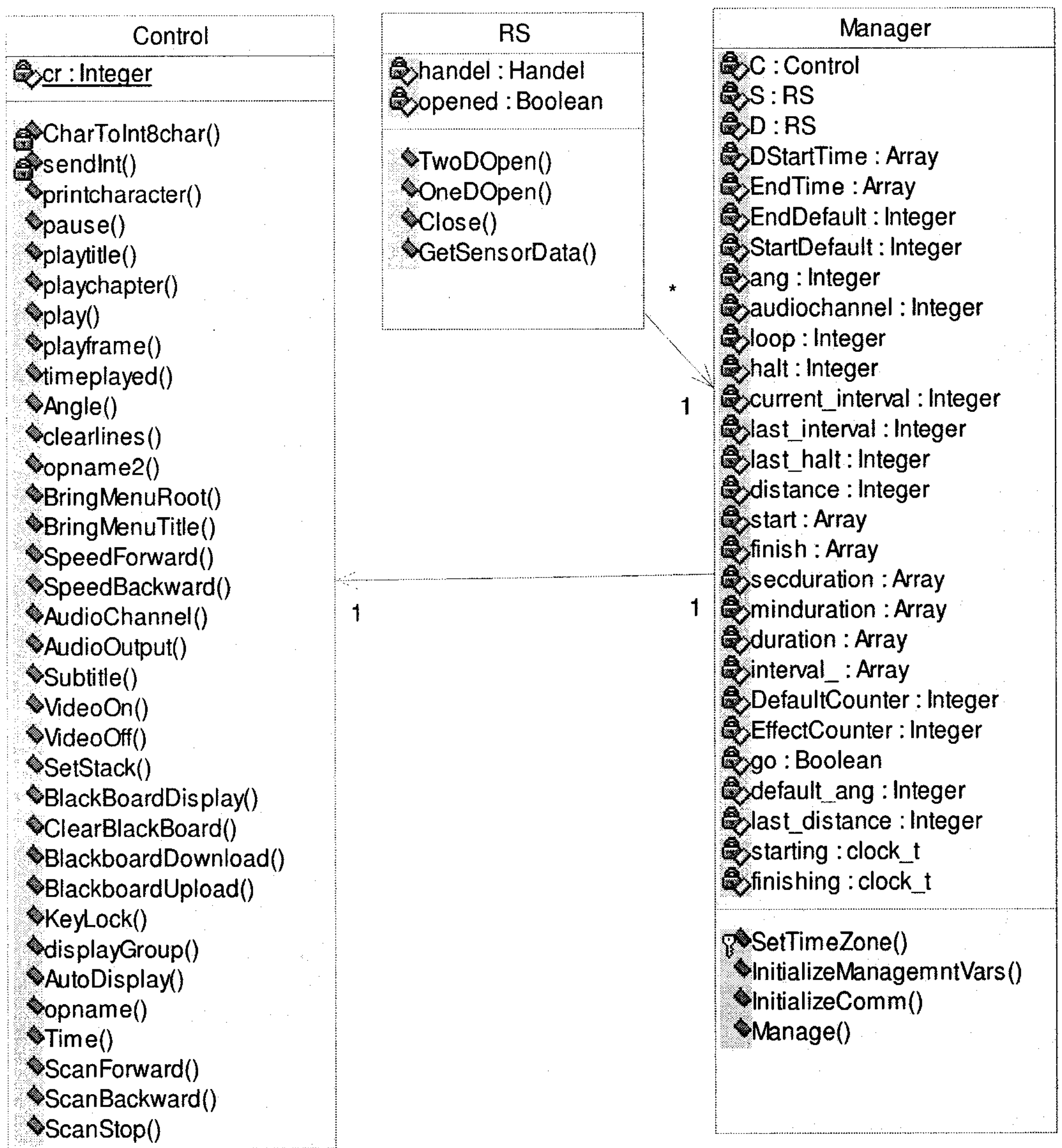


Figure 5: CPS Class Diagram

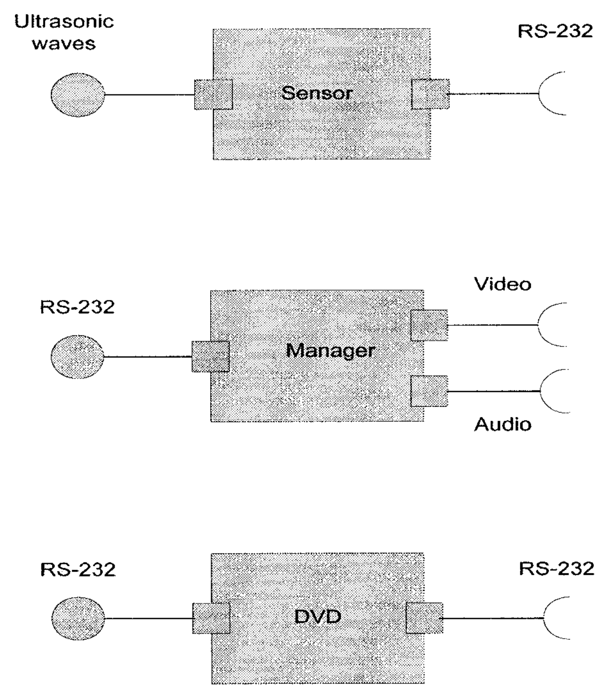


Figure 6: CPS internal structure diagram

Collaboration Diagram

The correct functioning of the CPS system is dependent upon the tight collaboration between six of its major components: the DVD controller, sensors, DVD player, operator, timer, and distance. The sensor gathers data about the users movements and sends it to the distance unit. Each distance is assigned a clock time before it is redirected

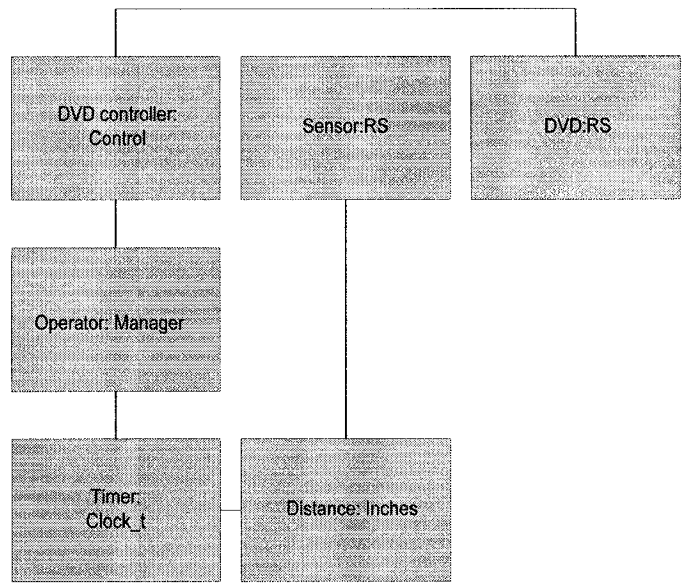


Figure 7: CPS Internal Collaboration Diagram

to the operator unit. Later, this operator performs logical operations, which results in the call of some DVD controller unit subroutines. There is also a dependency between the DVD controller unit and the DVD unit as the DVD controller subroutines use the serial port opened by the DVD unit, see Figure 7.

Pattern Diagram

The CPS software has been carefully designed to reflect the observer pattern. With a clear cut decoupling between the control and the view the CPS architecture constitutes a direct reflection of the Model/View/Controller principle. In fact, the MVC model fits the needs of the CPS system, which has relatively small code, uses sensory data for the control part and various displays DVD, projectors, LCD screens, computer screens... for the view part. Hence, a change in the sensory technology will require slight changes in the control part (RS class) of the code, but will have no effect on

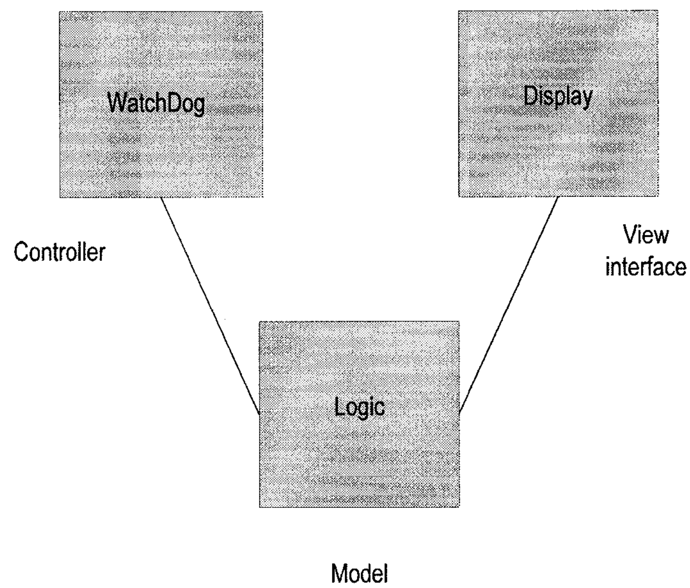


Figure 8: CPS Internal Pattern Diagram

the Logic (Manager class) or the view (DVD Control class). Similarly, any change in the display mediums will have limited impact on the system; it will require a given modification of the view part of the CPS system. Moreover, the isolation of the logical unit allows for a high level of robustness because any change in the technology would not affect the logic with which the CPS works. Any change in the systems behaviour and interaction could be integrated by changing the logical unit of the system only. The adoption of the MVC model has made the CPS more adaptable and flexible in dealing with any future upgrades or technological change, see Figure 8.

Component Diagram

The CPS is made of three principle components: watchdog (sensor, RS class), logic (manager class), and display (control class). The watchdog component receives ultrasonic weaves as a reflection of the users movements and interprets them into serial code that is sent to the logic component. This later analyzes the received sensory data and transmits serial commands to the display component, which changes the display content. This change may affect the user who reacts to it by making further movements. Hence, the interaction is a continuous action/reaction between the user and the CPS system, a loop that could end when the user leaves the interaction field for certain period of time, see Figure 9.

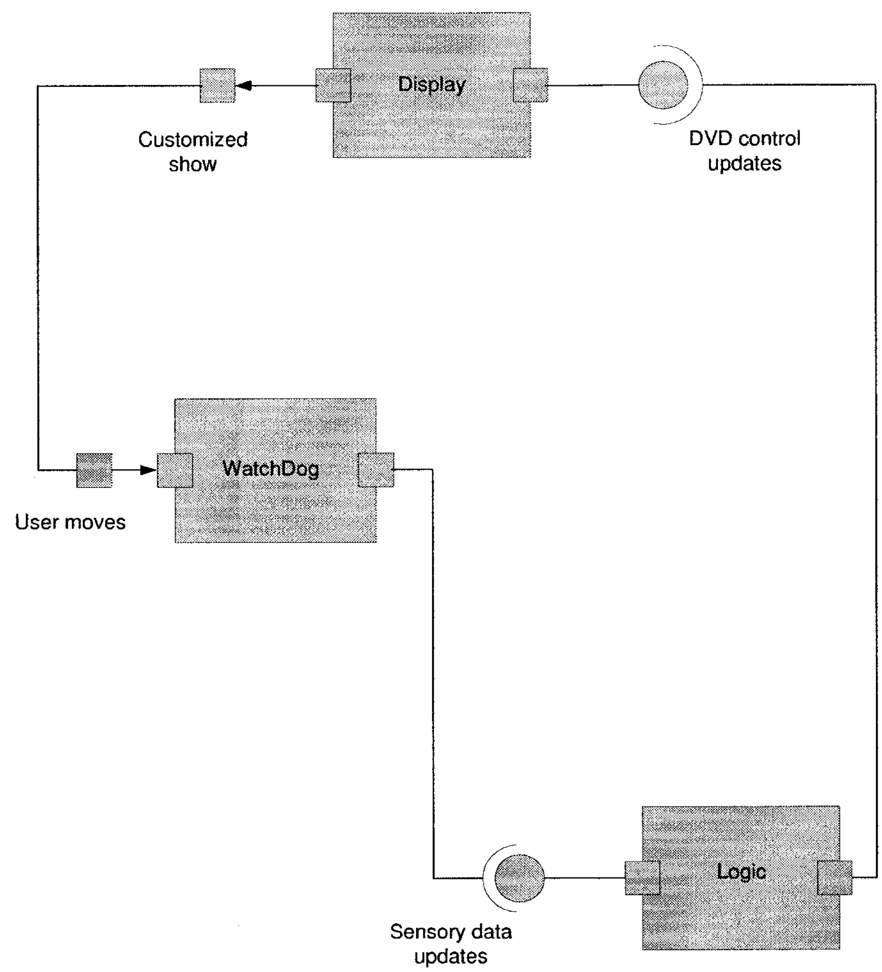


Figure 9: CPS Component Diagram

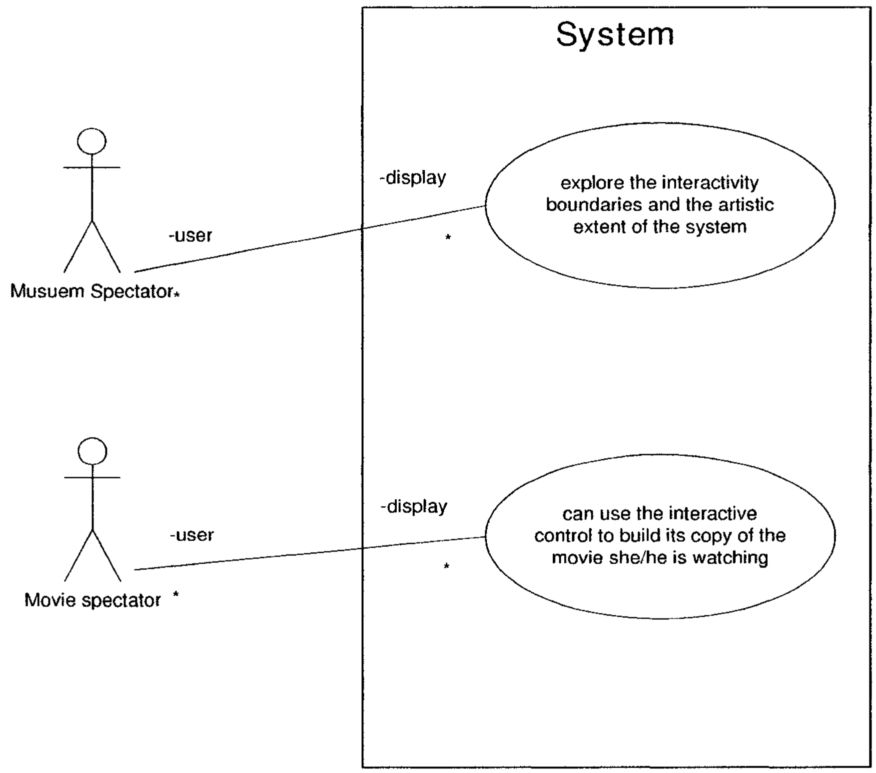


Figure 10: CPS Use Case Diagram

3.7.3 Use Case View

Use Case Diagram

Although the CPS system is capable of using different display content, integrating different sensory technologies, and having various audiences, this system was specifically designed to be a new form of expanded cinema, a medium that displays cinematographic content in an interactive way for museum or movie spectators. Hence, we have considered two type of actors while building the systems museum and movie spectators. The first actor is more interested in discovering the extent of the interaction and what this interaction would bring to the artistic content. The second actor is interested in dealing with the show, understanding it, manipulating it, and customizing the display. See Figure 10.

3.7.4 State Machine Diagram

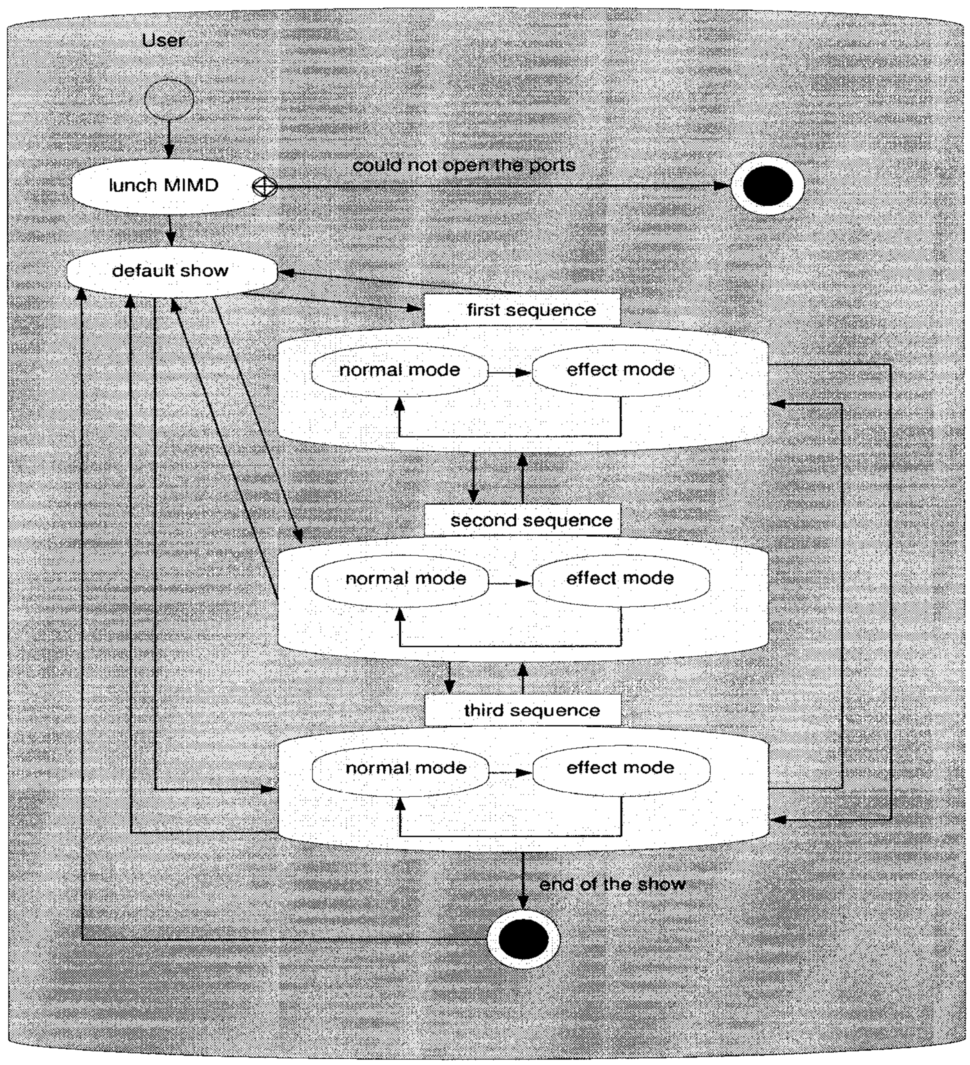


Figure 11: CPS State Diagram

The CPS works according to simple state machine sequencing. At the start, the system is launched by the artist or the artistic technician. Once launched the system displays the default show and waits for any movement from the users end. When the user enters the interaction field, the area in which the sensor is capable of detecting the users movements, the CPS jumps accordingly from the default show state to the sequence of the show that reflects the position of the user in respect to the display. The users moves affect the display criteria by adding special effects to it, mostly animation, view angles, scanning speed, slow motion, audio tracks, audio volume, text messaging, etc. These movements may also result in jumps from one sequence of the show to another or to the default show, if the user leaves the interaction field or has watched all the sequences, see Figure 11.

3.7.5 Activity View

The CPS activity is done in four main fields: user, sensor, logic, and display. The user starts the activity by taking an action or reacting to the display content. This interaction is resumed in two major states a move or no move (idle). In both cases, the distance of the user is recorded by the sensor and transmitted to the logical unit of the system. In this later state, the sensory data is treated and control commands are sent to the display, which executes the commands. The display modifies it causes the user to react, see Figure 12.

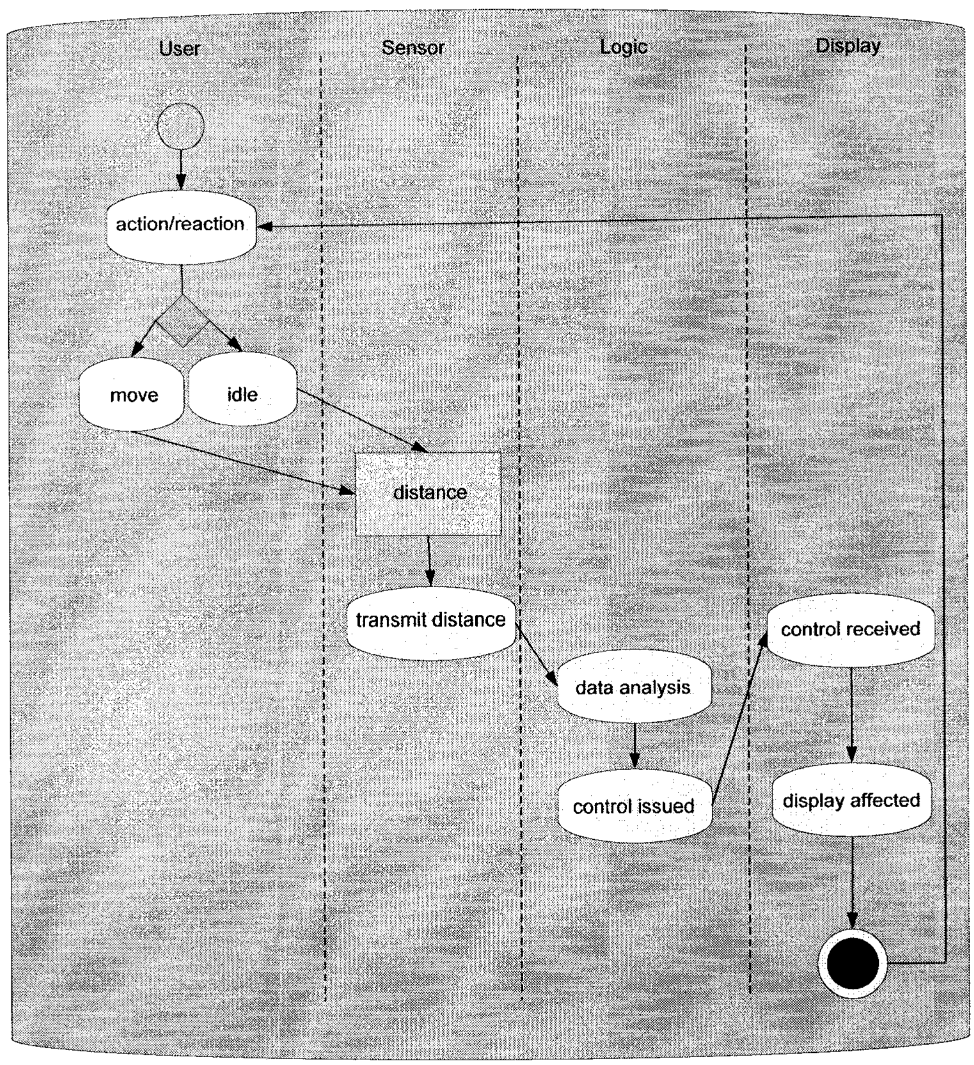


Figure 12: CPS Activity Diagram

3.7.6 Interaction View

Sequence Diagram

The user CPS interaction is done in five main steps: sensor, distance, timer, operator, and control. Sensor represents the first step, where the users movements are detect by the sensor and set as a distance in inches in the second step. During the third step, the system clock registers the time of the move and transmits both the distance and the time at which it was recorded to the logical operator, which issues a call to some of the control class subroutines. The last step resumes the execution of the called subroutines, see Figure 13.

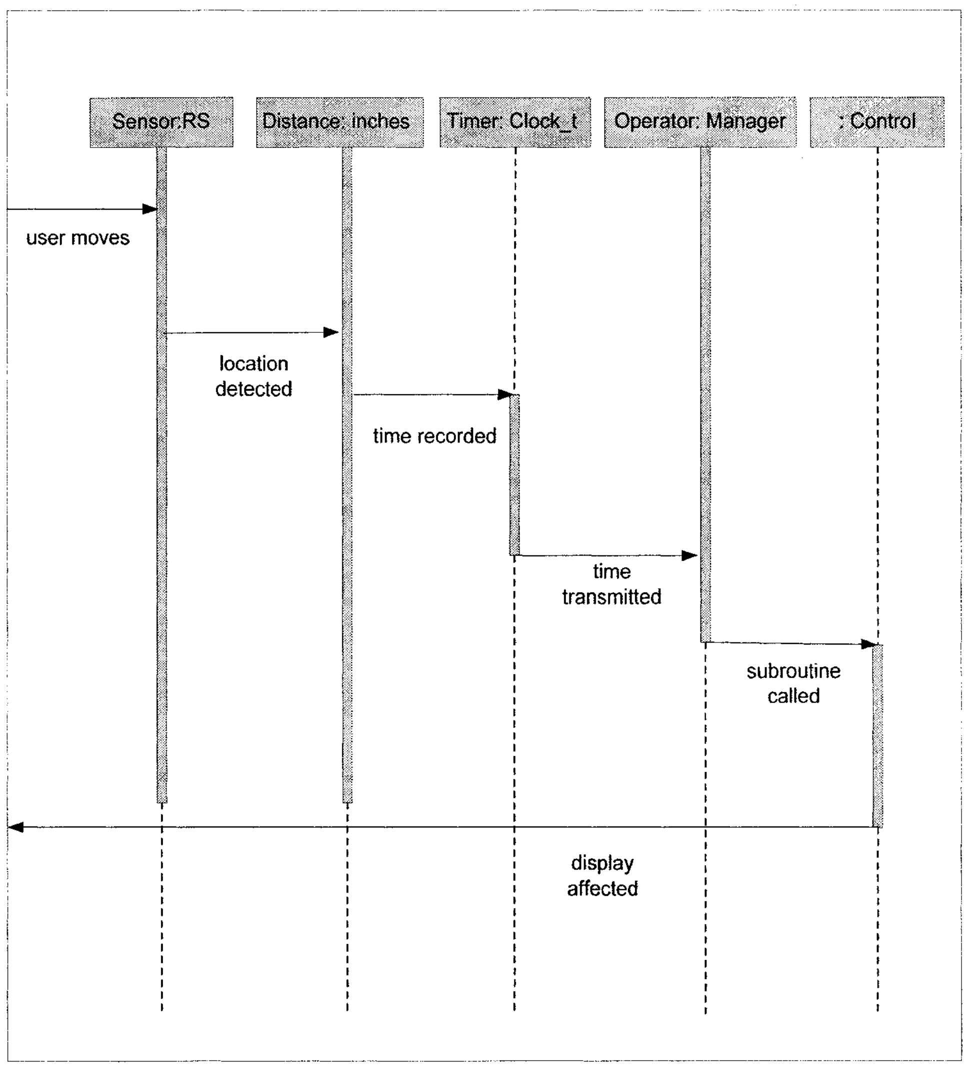


Figure 13: CPS Sequence Diagram

Chapter 4

Public Presentation, Feedback and Observations

Presenting the work was the most crucial part of the project because we were going to see if our assumptions and intentions were right. Moreover, it was the ultimate test to the practicality of our work and its potential involvement in bringing new methods of navigating movies and interacting with media content. Measuring the accuracy and efficiency of the system, mainly the software part of it, could not be truly achieved if the hardware suffered from any problems. Hence, we had to emphasize the organization of the material that we had on hand and our set-up design. Although we have taken all the protection measures, we were aware that such a test will reveal some misconceptions. Also, we were looking forward to hearing the users feedback, which would constitute a good measurement of our success in achieving the Movement Hypermedia Interaction (MHI) system and will constitute the basis for any future enhancement of the system. In the mean time, we were aware that our observations of the public are important. While hearing public opinion is good, watching the users reactions is better because the reactions are instantaneous and true. Indeed, we have gained a lot more from observing the users than from asking them about their view and remarks about the project. Finally, this presentation has not only allowed us to discover many drawbacks that we didnt foresee, but also enlightened us about many good aspects of the system. It also helped us to proceed into solving the most critical issues and to plan for the next version of the system.

4.1 Presenting the First Version of the Complete Work to the Public

The project had taken more than one year of implementation, but it had never been shown to the public before. Our main concern was about the inconsistency of the reactions or any crash of the system during the exhibition. Regardless of the fact that we have undertaken solid testing, we were conscious that it is a very delicate matter to anticipate all the users actions, especially in a system where no user manual and no instructions are set as a formal modality of communication between the user and the system.

4.1.1 Organization of the Material

Given that we dispose of heavy set of materials, including five 7400 DVDs, two Sanyo projectors, five Senix Ultra-Sonic sensors, five LCD screens, one projection screen, eleven RS-232 serial wires, and one computer unit, we had to make sure that everything was properly linked and that each unit was functioning correctly. Indeed, any deficiency of the hardware may affect the performance of the system, and hence, cause us to deduce wrong assumptions about the software implementation. Therefore, proper labeling, grouping, and unit testing was the eventual solution for proper assessment of the system. We gave a corresponding ID to each piece of equipment, put all related units together, and made sure that each and every unit is working with accuracy.

4.1.2 Organization of the Material

The set-up design was very critical because we had to be careful with the interference between each piece of equipment, the use cases and scenarios, the specificity of the place, and the placement of each unit. In truth, the aesthetic aspect of the presentation may influence the audiences appreciation and feedback, especially when most of the shows visitor had an artistic background. Therefore, we have paid close attention to this point by making the set-up as neutral as possible, so transparent in fact that it would not affect the visitors judgments and remarks.

4.1.3 The Users Feedback

We had the chance to have a mixed audience, including people from all ages and backgrounds, which helped enormously in receiving diversified feedback. In reality, we have received loads of comments concerning various aspects of the show, too many in fact, that we had to filter out the most relevant ones and place them within their user model category.

Kids

Most of the kids were fascinated with the fact that they could change the image without using a remote control. They were not concerned with the complexity of the system. Rather, they wanted just to trigger as many actions as possible. They were very happy and extremely motivated while changing the shows content. For them, it was a collective game in which your moves, your interaction with your screen and the one of your neighbour was accounted. They would run, walk, and jump all over the room and some were even looking to disturb their friends shows. Others enjoyed the signals on the screens, the change of video streams, the modification of the sound, the motion speed, and the different audio streams. Their interaction became a magnificent learning and entertaining experience.

Youths

The youths were very enthusiastic about the show. In fact, they supplied most of the remarks that we have collected so far. In the application, they saw a good platform of interaction and a different way of controlling the media content. Additionally, they spent more time with the applications and have tried to learn more and more about its functionalities. Indeed, they were asking questions all the time and they wanted to use the system to its maximum potential. Their curiosity about the system was amazing; they tried all the possible movements, gestures, facial expressions, and noise. In brief, they tried anything imaginable to get in communication with the application. What was wonderful is that some of them succeeded in mastering the communication techniques in less than ten minutes of interaction. They enormously appreciated the show and were looking forward to seeing the upcoming versions of the application that would include more control, more complexity, and if possible an incremental intelligent behaviour, where the machine understands the level of communication of the user and would react to his commands accordingly. This last point was the most interesting one and is what we would like to talk about in more detail. In fact, their request was that the movement system would recognize the level of interaction the user had acquired and behave respectively to that level. This brings the interaction between the system and the user into the level of human to human interaction, where the degree of understanding is reflected in the degree of interaction and communication. For example, imagine that you are in a foreign country and you would like to talk with people on the street. You would limit your conversation to simple words only those that you know, like yes and no. Moreover, your focus would be on expressing the most important terms, such as food, sleep, and friend. On the other hand, when your are invited to a party with your closest friends, you would all use the English vocabulary that you know, plus all the commonly recognized signs, and probably, some typical expressions used only within your group of friends. Therefore, you can see that the amount of information exchanged and the level of interaction depends enormously on the interlocutor, so reflecting this comment in our system would make it more powerful, more flexible, and more capable of achieving better interaction with the audience.

The Middle-Aged

It was quite an experience for middle aged people, who have seen similar applications in futuristic movies of the late 60s and early 70s, such as Stanley Kubricks 2001: A Space Odyssey (1968). They have enjoyed the interaction, but only to the limit of its usability, meaning that they used no more than what is necessary for them to operate the system and have avoided any additional effort to learn more about the specificities of the interaction. They were delighted by the ability to change the sequencing of the show just by moving. Nevertheless, some of them have complained that the complexity of the installation had prevented them from focusing on the content. They were looking for basic control and more interaction, which is not always easy to achieve. Despite some harsh comments, this audience was amazed by the technological achievements and has mentioned that good handling of the technology would lead to better human-computer interaction. Furthermore, they were optimistic about the future of such a project and have confirmed that they would not be surprised to see related applications in their daily life in the near future.

The Elderly

We were surprised by the attitude of the older generation that we thought would not appreciate our work, but who ended up enjoying the show and the interaction. For them, the kind of interaction that we offered, was much easier than using a computer or a remote control. They were able to interact with the application right way, trigger some actions, and have intuitively deduced many of the applications instructional movements. They were also happy that a little physical interaction allows for complex technological control. For them, our system was more natural; they did not have to use a tool or an appliance to control the system as everything was triggered automatically according to their movements. In addition, they were glad that they did not have to read a user manual or learn anything before they began interacting with the system. In fact, their appreciation comes from the fact that the machine complies to the user and not the other way around, which was one of our objectives. We have always wanted to achieve a free body computing system, where the user is free from the use of any interaction instrument and from complying to the machine requirements. Therefore, we were very enthusiastic to see that some of our initial goals were partially met, which strengthened our determination to pursue our quest for a better free body interaction methodology in order to attain natural and smooth human-computer interactions.

Artists

People with some artistic background were very enchanted by the show but they also had very critical remarks. They find out the experience very exciting and were asking for more forms of interaction and control. They wanted to be able to manipulate the content of the images, the amplitude of the volume, the text messages that appear on the screen, the scanning speed, the color of the background, the camera location, and many other applications.

Computer Scientists

We also invited some students and professors from the Computer Science Department at Concordia University to the show. They were very interested in the technology, the sensors, the communication system, and the software. After asking several questions, they started giving suggestions. For them, the show is a communication network that uses sensors to create a recognition-based interface. For example, we could achieve intense communication between all the units in the network in order to allow both individual and collective interaction. In other words, the show will react to the individuals movements, but will also reflect the collective interaction and the interest showed to each component of the network. They have also proposed a combination of recognition methodology, such as using tracking cameras with ultra-sonic sensors, so that we can both differentiate the shape of the object and know its exact distance from the sensor. Another pertinent proposal is the integration of Artificial Intelligence (AI) technology as way to achieve smart responses and evolutionary interactions. Accordingly, AI will enable the system to adapt and suit the users triggering behaviour. Above all, they were optimistic about the future of our project and about the various fields in which this technology could apply.

Experts

Having some experts come and visit the exhibition was a critical point for our project because their visit would allow us to position ourselves within the different fields of research covered by our study. Also we were aware of the valuable contribution that they would bring to the show through their comments and suggestions. Therefore, we had managed to organize private visits for our experts, so that that they could observe the work at their ease and convenience. In fact, these visits have highlighted many critical points of the show. To start with, we will discuss their comments relating to the sound effects. For many experts, the sound was a very pertinent component of the show and given that the sound tracks of all the monitors are played simultaneously, they have suggested that the show take into consideration this mixture of sounds and manages the sound tracks in a manner that improves the interaction. Other experts have raised the need for online access. In other words, they suggested that our network would be up-to-date, if it was connected to the internet. If we implemented this suggestion, the content of the show would be updated automatically from the server and the users could have access to the latest news related to the show, to further shows, to any additional content or to other online applications. This suggestion was seen as an obligation given the current integration of all the computer appliances into the World Wide Web. Moreover, the implementation of such a suggestion would confirm our primary aim of creating a Free Body Hypermedia system (FBHS- Mohamed Hachem and Jean-Claude Bustros). In reality, the idea is simple. Instead of having a multitude of devices all over the world to achieve what we call ubiquitous computing, we would have media connectors, which take the form of simple screens with an integrated recognition-based technology. These screens represent what we call hypermedia connectors, which are a set of stations able to provide hypermedia recognition-based interaction. Through these connectors, the user can access any media content, watch TV shows, send emails, listen to music, take online courses, make phone calls, perform money transfers or attend museum exhibitions wherever a hypermedia connector is available. Hence, users would be free from using the traditional user interfaces and would be dealing with one media platform. This device would replace the gadgets that surround our daily life, including: computers, laptops, Palm, TV, Mobiles, etc, see Figure 14.

Additional suggestions concerned the recognition technology. We have been advised to use a grouping of movements, gestures, visual-recognition, and speech-recognition techniques in order to permit a natural mode of interaction that is closer to the way we communicate in society. We use our vision, gestures, movements, and speech to express most of our ideas, when interacting with other people. It was clear that empowering our recognition-based methodology will help to increase the amount of data to be exchanged between the user and our system and will shorten the communication time with users. Nevertheless, such a suggestion necessitates a compound approach that integrates all these systems together. Moreover, we had to think about the complexity of the interaction, the feasibility of the implementation and the accuracy of such an application. In brief, expert advice has been very precious in tracing the path of our future enhancement of the system and in discovering the various possibilities of extending our application.

Others

For the other trench of the public that has little background in art and computer science, the show represented a good entertaining platform. Many described their experience as going to the amusement park or to the casino. They have felt free to interact with the system and free to understand whatever was useful for them to understand. For them, this type of interaction, which is free from commitment, made the complexity of the show easier to absorb and smother to approach. Each of them has projected a particular usage of the technology in her or his domain of vocation.

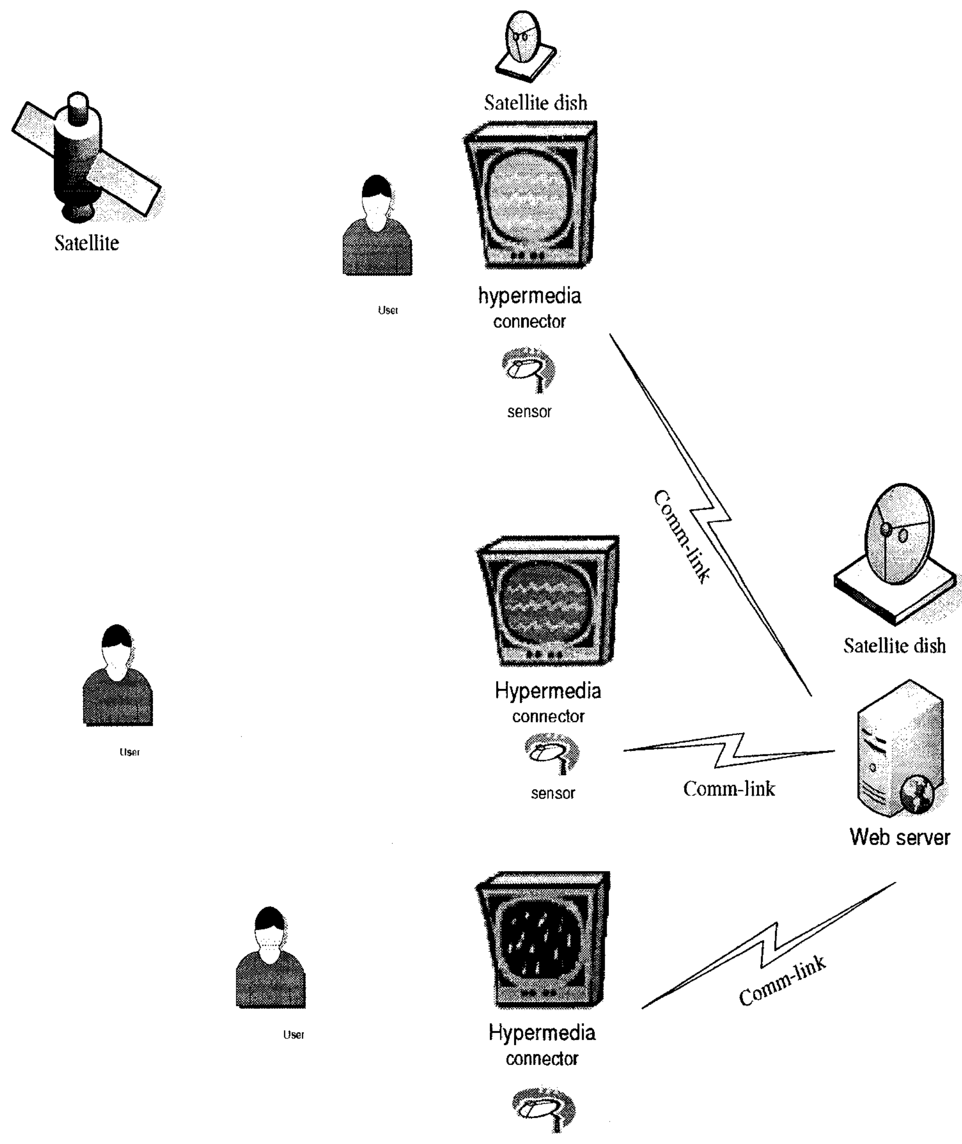


Figure 14: Free Body Hypermedia Interactive System

We have been told that our system could be perfectly integrated in dancing lessons by generating the music according to the dancers movements and vice versa. Other people have suggested that it would be helpful to have our movement recognition system in an assembly line, so each machine would be aware of the speed and exact distance that separated it from the item that it should assemble. Some people were willing to have the system in parking lots or to have it integrated in their cars, so parking and driving will be supported by close detection of the surrounding obstacles. Shop owners wanted to have the system installed in their stores, so those passing by would trigger actions, which will cause the screens to change. They believed that interest in interacting with the system may turn those passing by into valuable customers. Video gamers and those working for video game companies have seen great potential in building games in which the users physical effort, while playing and visualizing the game, will increase the immersive sensation of the game and provide a healthy alternative to using video game controllers. For many physicians, the cheap cost of the technology could be invested in making a detection system to be used by blind people to detect the obstructions around them. They have argued that a simple system, which alerts them about obstacles within 120 inches by voice or sound signals, would allow a blind person to dismiss the use of their cane. In short, these various remarks have made us feel proud to be working on such project and make it clear for us that science has no borders.

4.1.4 Our Observations

We have placed massive interest in the observations because they were meant to representation the true common reaction of the public. Therefore, the feedback was very helpful in evaluating the accuracy of the system and in deciding the future path of development of it latter. We expected that the observations would play a greater role in defining the recurrent behavior of the users and in tracing the upcoming versions of the MHI system. Our observations are comprised of two distinct categories. The first category concerns the users behaviour with the system. In other words, how does the user interact with system? The second category concerns the reaction of the system to the users actions. While discussing user interaction with the system, we became aware that most of the exhibition visitors were unfamiliar with recognition-based applications. Moreover, the best experience that some would have is with basic recognition systems that yield unique reactions. For example, when you move your hands towards the tap, and water comes out of it. This limited experience had a big impact in the users behaviour. It was very different from our expectations. In fact, the main scenario is obsolete. Although we were disappointed to see that our use cases were far from reality, we were happy to make these observations so that upcoming versions of the system could be improved upon the original. Concerning the systems reaction to the users behaviour, we were pretty relieved to see that our application was able to handle most of the users actions, even when they did not belong to any use case. This robustness is due to the solid safety system that was integrated within the application and that could be enhanced to assure more and more stoutness. To conclude, our observations highlight the mismatch between our scenarios and those undertaken by the public due to the visitors unfamiliarity with recognition-based applications. On the other hand, we were satisfied to know that we possess a firm system that can handle most of the unexpected use cases.

Triggering an Action: Assumptions

Our primary observation was that the users ignored the triggering actions, especially in the absence of any user manual or instructional usage of the system. They did not know which action triggered which reaction and when. In other words, it was not obvious to the public when some of their actions lead to a change in the display content. This confusion and lack of information tampered the interest of some users and have played a role in making others ignore many of the systems functionalities entirely. In fact, the learning curve was a bit stiff and we had to adjust it in order to keep the audiences interest high.

Triggering an Action

A common behaviour among most of the visitors was the assumption that they needed to be close to the screens to interact with the system, especially as these screens were small. Indeed, their approach toward the screens narrowed the interval of interaction and lead to triggering fewer actions. Furthermore, it contradicted our main scenario in which we assumed that people would start their interaction from a minimum distance of 120 inches. Hence, this behaviour has not only shortened the interactivity of the show, but also has forced the system to respond to some actions that are not included in the use cases as the users werent aware that they were already interacting with the system. Such behaviour could have been detected earlier if we had done a solid study of the user-model attitudes while building our scenarios. Most of the users are used to interacting with sensors at short distances and watching TV at a short distance. Therefore, a careful study of their attitudes would have detected that the user would move close to the screen and then start backing up. However, it is always difficult to predict the users interaction scenarios, particularly while using recognition-based technology. In brief, knowing that the users have a tendency to start interacting with the system at a distance of a few inches from the display screens has helped us design new scenarios, where the interaction takes into consideration this common attitude

Gestures Instead of Movements

Given that hands are the most popular parts of the body to be used in interacting and modifying our surroundings and that gestures prove to be more intuitive in expressing the thoughts of human beings than movements do, most of the audience members had a penchant to use gestures in some way to articulate their interest. They also employed gestures as a tool to trigger reactions. This penchant was increased because the narrow beam of the sensor could be easily affected in certain circumstances by the hands movement and would, in many cases, provide the sensation that gestures are the right way to interact with the system. This tendency resulted in unexpected interaction scenarios that will be taken into consideration, while designing the interaction standards of the subsequent versions of the project.

Targeting the Sensor

Since the sensors that we have used were ultra-sonic, they had to emit low volume noise in the form of clicks. These clicks raised the attention of many of the audience members, who sought the source of the noise and most of the time, they ended up localizing the sensor which is easy to distinguish. Once the users are aware of the location of the sensor, their main target would be to play with it in order to figure out the different actions that they could trigger. This random discovery of the system reactions inflicts serious problems to the application that needs to deal with unpredicted behaviours. Sometimes the quest of learning about the interaction mechanism and focusing on the sensor leads to erroneous conclusions and limits the interaction possibilities because most of the users stay close to the sensor. This behaviour prompted us to seek adequate solutions for the upcoming updates of the system. So far, we are thinking about making the sensor as invisible as possible or replacing it with another sensor that would not make such a distracting noise.

Large Audiences

The exhibition attracted a large number of visitors, including people with both art and computer science backgrounds. This large audience forced the system to cope with an overpopulated stage, where users utilize of a tight interaction space and each display has to respond to the movements of dozens of spectators. Given that the most sophisticated scenario admitted the presence of three to four people in front of the screen, we were very interested in seeing how the system would react under these stressful conditions. In fact, the systems simplicity in triggering reactions was a key determinant in making it behave accordingly with a large audience. The systems mechanism consists of triggering actions according to the users movements and each movement corresponds to a given distance from the sensor. In the case of a large audience, the system deals only with the closest user because the ultra-sonic waves reach this person first and returns them back to the machine. Hence, the problem of a multitude of users is dealt with through the plainness of the systems sensing devices. Nevertheless, we have observed that the system shifts to the safety state or records a freeze before entering the safety state in cases where the users change their locations frequently. In reality, when there is more than one spectator per display, the system reacts according to the closest user called the key animator. However, in the case of a key animator, who is constantly changing as the spectators shift their positions continually, the system jumps from one state to another in quick succession. These state changes, if triggered recurrently and rapidly, lead to the freeze of the display not because of a software deficiency, but due to a hardware failure. In other terms, the audiences frequent displacement generates the triggering of too many commands in short spaces of time; this huge number of control actions is barely handled by our DVD players. Most of them crash after an instantaneous series of instructions. In brief, large audiences put the system into a high level of interaction that is fairly handled by the method of serving the nearest spectator and ignoring the rest. Nonetheless, this intense interaction may cause some hardware failures that are recoverable after a few minutes of freezing by the systems safety components.

Fast Movements

During the installation, a large body of the audience was formed by youths, who were very energetic and dynamic spectators. Instead of moving in front of the displays, they tended to dance, jump, and run around. These reactions were not accounted for in most of our scenarios. Moreover, the users quick movements led to triggering a greater number of reactions, which brought the show to a situation similar to what we experienced with large public exhibitions. Nevertheless, the cause of the systems malfunctioning in this case is different. Indeed, users who move fast would request not only a larger number of reactions, but also a faster time of reaction. However, these requirements are limited by the hardwares reaction time. Although the sensors have instantaneous reactions in most cases, the same is not true of the DVD player. To follow the users instructions, like changing the video angle, the industrial DVD player that we used requires up to three seconds. The latency of the DVD component pulls the system into an undetermined state each time the user at a rate quicker than that required by the hardware, which generally leads to a freeze followed by a recovery. In short, fast moves of the public require faster hardware dispositions. These fast movements could be compared to a person pressing all the buttons on a remote control at once. If you have ever tried this, you would have noticed that your TV does not reply to all the commands, and most of the time, it will shut down. This is exactly what happens to our system. Instead of shutting down, however, it freezes and relaunches its application after a given period of time.

No Abnormal Actions

Over the ten days of the exhibition, the system had to work in nonstop mode. Due to the systems continuously working state, we anticipated the registration of some abnormal behaviour, especially considering it was going to endure a superior amount of stressful conditions. Luckily, we found that the only unexpected reactions were the freezing of the display and the stopping of the machines. These cases were easily treated by the safety system, which re-enabled the functionalities of the system a short time after the crash. Though it would be interesting to mention that on rare occasions we observed strange reactions related to the non triggering of the required actions, this incident could be simulated by the case of pressing the remote control button twice to switch on the TV. In fact, in some exceptional conditions, repeating the action twice or evacuating the sensor detection zone is necessary to perform a given action or to re-initialize the programs variables. These cases were so rare and irrelevant that we have disregarded their occurrences. To sum up, we expected that our system would generate a huge number of abnormal reactions throughout the exhibition period, but we were glad to discover that the only unexpected behavior was the freezing of or the shutting down of the DVD player. In both cases, however, the system was able to recover and go back to its normal state.

Safety System

Any software project must include a backup plan and a safe state that is used to ensure its normal running in cases when it encounters unexpected conditions. Our safety system was designed to recover from crashes when the shows mechanism crashes is overloaded by information collected from the sensors, the big number of commands sent to the DVD players, or any other abnormal usage of the system, such as running and jumping. The safety arrangement consists of two methods of recovery. First is the preventive method in which the system ensures that all variables are re-initialized and that the system is re-launched thirty seconds after detecting no users in the sensory zone. When re-launched, the exhibition will be limited to the default scenes of the show, if no user is detected. Afterward, it will re-launch and refresh its main variables once in while to correspond with the length of the default portion of the displays content. Second is the post-crash method. In this recovery situation, the system has crashed and the users are facing a situation similar to the one practiced when a computer crashes. As you may have experienced, most of the time when a computer crashes, we have to choose between two main alternatives promptly terminating the process that spins in the memory and is the source of the crash or restarting the computer. In these cases, parallel solutions are offered by our system. In the case of a crash, the user can overwrite the previous commands, which caused the crash in the first place by issuing new commands or evacuating the sensory detection area for thirty seconds or more,which represents the time required for the preventive method to activate. In most crashes, these two methods have been proven successful in achieving full recovery of the systems performance and have been also observed to be the most instinctive behaviour undertaken by the general public. What is good about the safety system is its capacity for keeping the show functioning constantly without the need to restart or shutdown the application. Thus, the safety system boosts the systems robustness and provides the extra features of endurance and self maintenance. In brief, we have built solid recovery system that permit a high level of reliability and includes two methods of recovery: one activated by the system itself and the other is impulsively triggered by the users.

4.1.5 Modification of the Interactivity Process

The exhibition period had left us with a load of suggestions, observations, and innovations to be imported to the project. There were so many things to change that we have decided to focus on the most urgent modifications, which are necessary for the good functioning of the system. Our choice was resumed in five main transformations. First, we have thought about adding little pictorials and some text messages within the display to create a simple communication signaling system between the show and the users. Second, we have learned that it was necessary to reverse the time zones of the shows content to make the most intimate scenes far and the most general scenes when the user stands close to the screen, which is contrary to the set-up that we had before. Third, we have also observed that reversing the scenarios was compulsory to reflect the users typical behaviour with the show and to adjust to the time zones changes. Fourth, the modification of both the time zones and scenarios has imposed a logical alteration of the side effects. Fifth, several incidents that we have crossed through the exhibition period made it clear that we must develop a solid packaging for the set-up. Finally, the show has been very helpful in opening our eyes about some important issues about the primary version of the project and has allowed us to produce a second version of the exhibition, which better reflects the users behaviour and permits higher level of robustness and adaptability.

Adding Signals

While observing the users behaviour, we were concerned by the fact that the users behaved blindly with the system, especially when knowing that they have no clue about the communication protocol and the extent of their interactivity with the system. This was due to the fact that they have been provided with no user interface manual and no usage instructions. Our major remark was that the users were unaware of most of their effects on the show. They did not know when and how they have triggered some actions.

It was clear that there was a kind of interaction, but it was not straightforward for them either to figure out the interactivity mechanism or to understand the full capabilities of the MHI system. This situation would be best described by the interaction that two people from different cultures and with different linguistic capabilities would have if they met in a given location and had to communicate with one another. Since, the formal language would not lead to any exchange of information, these two people would have to agree on some signaling system that would be taken as an alternative and would help in forming the basic common ground of interaction. This is exactly what we needed to solve our problem.

Not only do users need to use simple movements in order to affect the system, but also the later versions of the system have to reply by executing the users commands and showing an understanding of their actions, by using simple signaling system. This signaling method would be an indexing application that would help users discover the different capabilities of the system. The signaling system would be simple and consists of brief text messages and some basic geometric forms, such as squares, rectangles, and triangles, of different colours, mostly red, yellow, blue and green. What is good about this system is its minimalism and its smooth learning curve. Furthermore, this system could be effortlessly inactivated by disabling the graphical display ability for veteran users. Another positive point of the signaling system is its graceful extensibility. Indeed, the signaling system could be simply extended to cover any potential functionality and to adapt to any kind of futuristic interactivity. Moreover, the use of such a system would be vital to inform users of the critical states of the system, like a crash or system freezing, and the proper way to recover from them. Nevertheless, this system will continue to contain no formal instructions for the users and will be limited to a simple indexing methodology that is straightforward and undemanding. Thus, the user would never have to learn about its interactivity with the system through instructions, but uniquely, through direct contact. Accordingly, our system requires no communication bottle neck and opens the doors to a technology that is easy to use and one that provides a high level of quality control. The idea is simple; we want users to exchange a greater deal of information, while using the most intuitive form of communication. To conclude, some of the shows visitors have experienced some difficulties in determining when and how they have triggered some of the systems actions. This confusion and ambiguity have prompted us to seek for a common background of communication that is easy to understand and that would not affect our primary principle of avoiding a user manual for our system. We ended up adding a signaling system that uses simple pictorials and short text messages, an intuitive system that allows for better communication and calls for no engagement from the users part. It was also used to enforce the safety system by informing the users of the significant states of the system and the right way to overcome them.

Chapter 5

Conclusion

5.1 Computer Science Approach

During the last decade, the computer has widely expanded its boundaries from the business realm and industry to entertainment, education, and communication. They imposed themselves as multimedia displays capable of dealing with digitized media and services, such as phones, TVs, radios, magazines, banking, etc. Moreover, this position has allowed them to become the main source of interaction within the digital world. This evolution in the functionality of the computers was not followed by steady changes in the human-computer interaction technologies. Rather, these technologies remained constant for more than twenty years prior to this revolution. They were conceived to allow the user to write text documents or to send execution commands to certain applications. Today, however, they are used to control complex media content and handle sophisticated methods of communication. In fact, the mouse and keyboard are stationary tools of interaction that lack the dynamic interactivity when applied to multimedia content, whether it be navigating through a movie, dynamically editing a song, or exploring three-dimensional content. This situations has brought us to think about new technologies of interaction. We come up with CPS, which represents multimedia display, which adopt a mobile form of human-computer interaction. It was conceived to support non-tangible user interfaces and movement-based, human-computer interaction that permits the dynamic non-linear navigation of media. It represents an alternative to classic human-computer interaction. Moreover, it promotes the idea of Free Body

Hypermedia Interaction (FBHI), where the interaction between the user and the multimedia display is user-centric. User-centric means that the technology is there to serve the user, rather than to impose any change in the way this later normally interact with his milieu. In fact, the interaction should adopt some natural form of communication through movement, gesture, speech, etc. It has also to be intuitive, so the user will not be prompted to behave in certain ways or learn anything in order to interact with the computer. Additionally, the user does not have to wear any gadget or use any tool during the interaction process. The last point is stressed by FBHI as the users senses are free from any control from the computer side.

5.2 Cinematic Approach

CPS is meant to reflect a new form of expanded cinema that calls for quadratic interaction between the user, the environment, the screen, and the image. It places itself in the free body media artistic interaction before that of augmented and mediated reality. It offers the possibility of using the environment and the human body as a medium of communication and interaction with the display and the artistic content.

5.3 CPS

CPS consists of a display, a sensor, and computer program. The users movement are recognized by a sensor, treated by the computer program, and reflected by the display. CPS is composed of a simple event system that allows real- time interaction between the user and the display content. The show consist of a documentary movie that is browsed through the user‘s movement. The interaction allows also for access to extra information, the addition of video and audio effects, and the appearance of animation, text, and graphic signals.

5.4 Future Enhancements

CPS initiative has welcomed and received research grants, which will allow for its development probably by the integration of other methods of recognition-based user interfaces. Moreover, the implementation of a causal engine over the actual event system will permit more flexible interaction.

Chapter 6

CPS Version 2.0

6.1 Introduction

CPS Version 1.0 has captured the interest of many parties, including the Conseil des Arts et des Lettres Quebec, the Conseil des Arts de Montreal, the Canada Council for the Arts, Parisian Laundry, Groupe Molior, and Hexagram. The support of these institutionshas allowed us to conceive and implement CPS Version 2.0, which was on an exhibition from 26 of October till December 2, 2006 at the Parisian Laundry museum (Montreal), see Figure 15. Version 2.0 represents an improvement of the earlier release, see Figure 16.

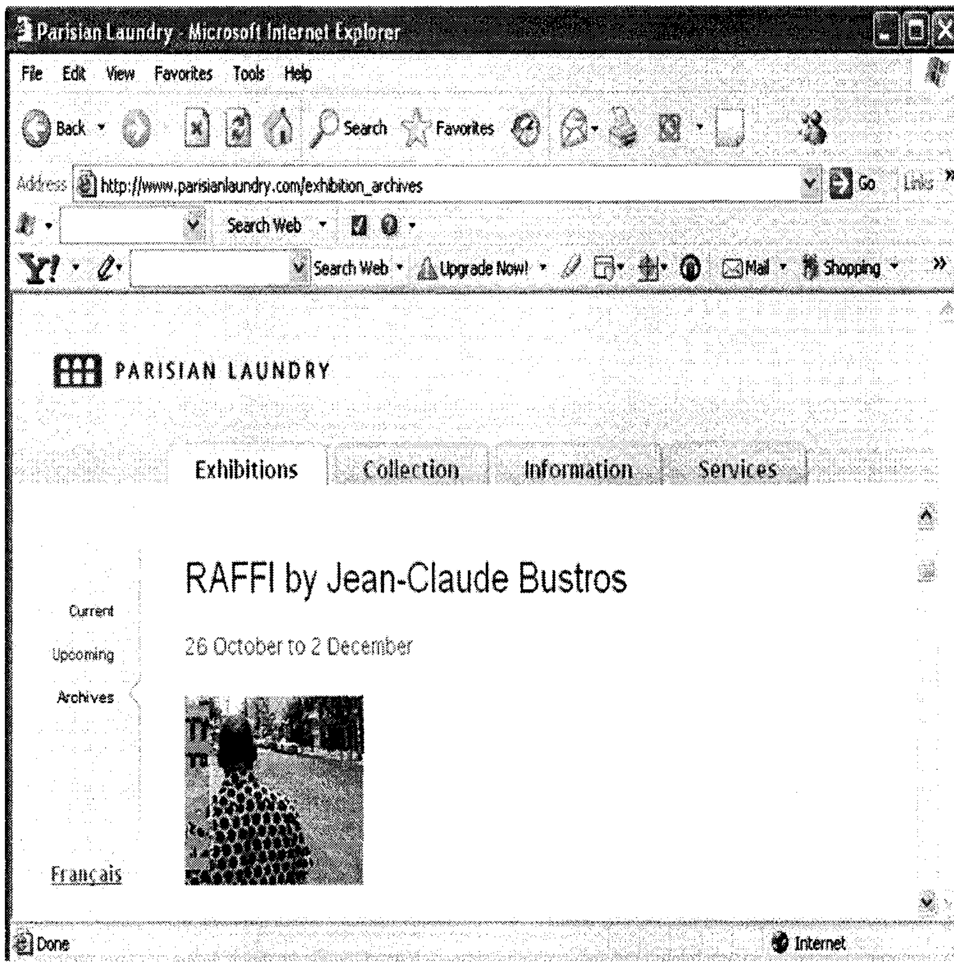


Figure 15: CPS Online Advertising for the Exhibition

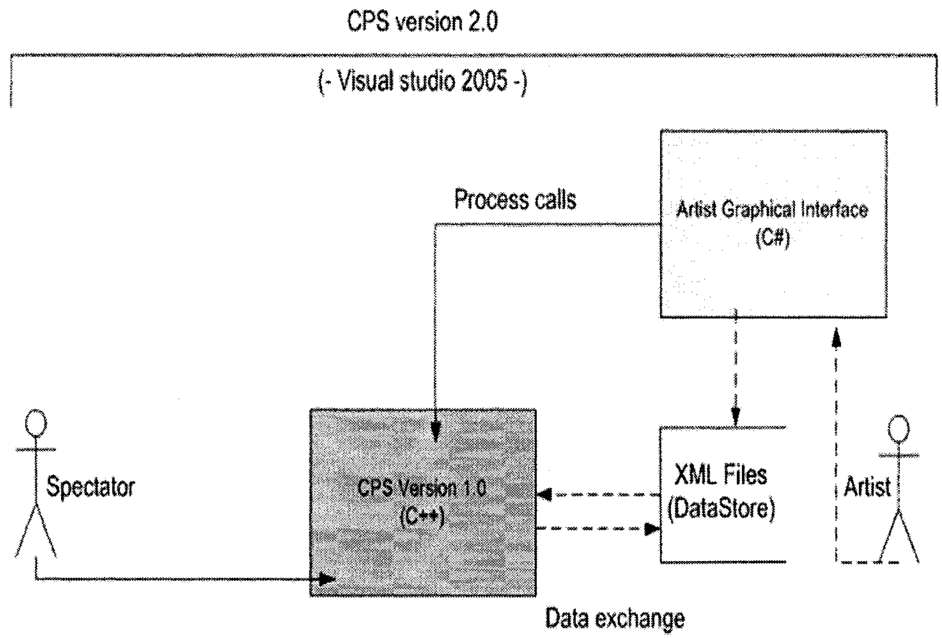


Figure 16: CPS Version 2.0s Global Structure

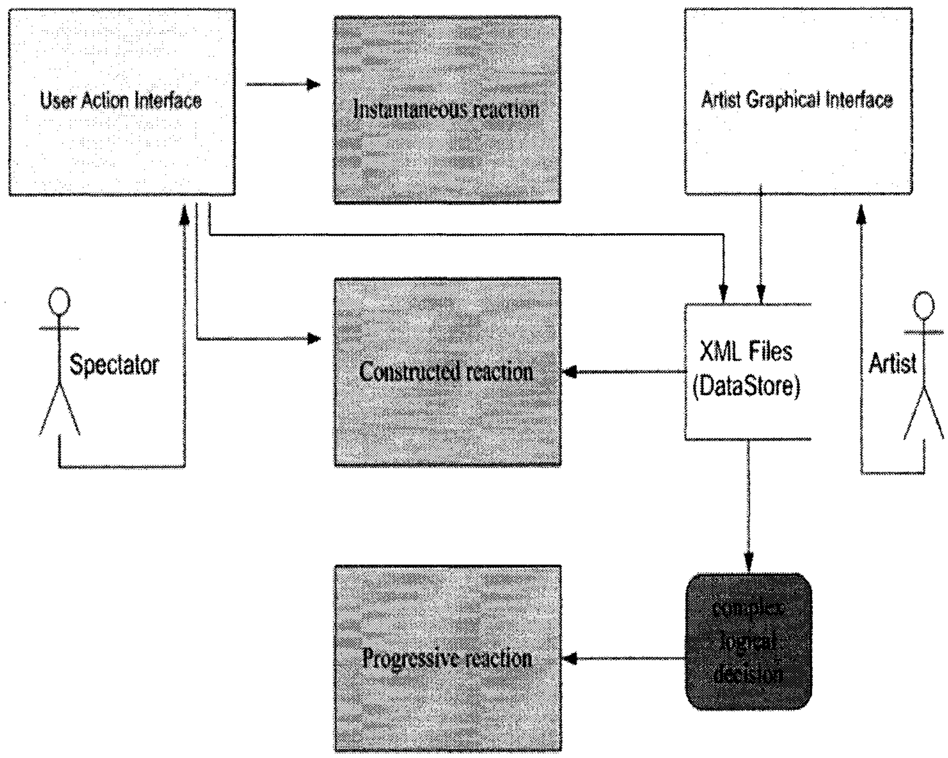


Figure 17: CPS Systems Reaction Diagram

6.2 System Reaction

The main enhancement of the system concerned its reaction that was organized into three principle responses that are instantaneous, constructed, and progressive. First, the instantaneous reaction is employed when the system instantly responds to the users movements; data is received from sensor and treated instantly. For example, the users advance toward the display screen and a red square immediately appears on the display screen. Second, the constructed reaction occurs when the system response results from an analysis of the users movements. It is a short term reaction based on a compound logical interpretation of the users current actions, combination of sensory data, data hold in the cache memory. For instance, the users presence in the different zones of the show is detected and the time spent watching each zone is recorded. This information is put together in order to decide which sequence of the show should be displayed in the current presence zone. Third, the progressive reaction is a complex logical decision made by the system after analyzing the users actions over an extended period of time, data is provided from xml files where interaction scenarios and user behavioral reactions are saved. Case in point, the system records the time that the user has spent watching the different sections of the show, the popularity of zones visited by the users, and the presence time in each zone during one day of exhibition and would decide which interactive scenarios ought to be adopted on the next day of the show. For more insight about the systems reaction, please refer to Figure 17.

6.3 Collaborative Behaviour

Another add-in CPS Version 2.0 is the implementation of collaborative behaviour between the different components of the system. Actually, each display unit keeps one XML record of its behaviour (content, effects, duration, etc) and another XML record of the user reactions (presence, moves, distance, etc.). These records can be accessed (read) by any of the other applications. The exchange of information between the applications through the use of XML files allows some applications to make decisions concerning, which part of the DVD content would be more interesting or less interesting to the audience. It is also used by the projector to decide, which the parts will be chosen for display. It is true that this collaborative behaviour has injected more dynamism to

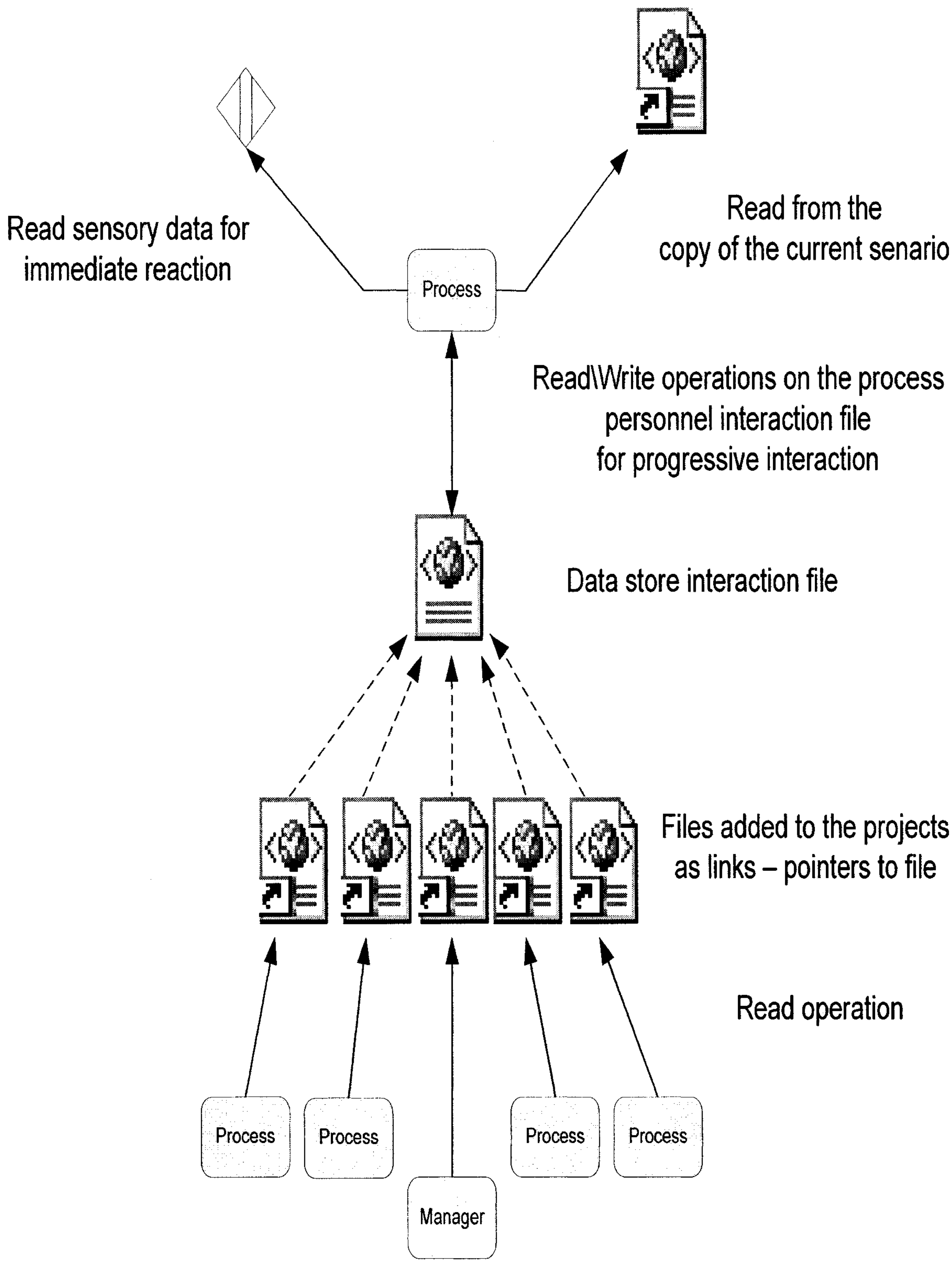


Figure 18: Data exchange methodology leading to collaborative behavior

the show by having an internal interaction (system-system) plus the reactions (system-users). However, the real gain is still to come because the collaboration is still in its genuine state (there is no restriction in reading the files, the logic of interaction is based on simple rules, and there is no learning involved). For insight on the collaborative behaviour mechanism, please refer to Figure 18.

6.4 Analysis of User Behavior

The CPS Version 2.0 offers enhanced analysis of user behaviour. Indeed, recording the users behaviours in XML files has helped the system make complex decisions based on the users historical interaction. As an illustration, if users frequently perform a particular behaviour, like going into and out of a certain zone, the system would be able to detect that behaviour and could react to it. Though the data is available, our analysis procedures were short and would be improved in the future CPS versions.

6.5 Loading Interactive Scenarios

The system has included new interactive features, which load interactive scenarios on the fly and are written in XML files. Loading interactive scenarios allows the systems behaviour to change dynamically. In fact, a given users action would lead to different system reactions according to the time of the action and the history of the interaction. This combination brings the interaction into a higher level of communication between the user and the system. Moreover, in this level, the relation between an action and a reaction is not always straightforward.

6.6 XML Documents

The program uses XML documents to hold user interaction data and the data put by the artist to orient the behavior of the system. The data is either collected from the sensors and form what we call the interaction data store or from the scenarios that the artist has saved through the CPS graphical user interface and is called the scenarios data. Both forms of data have specific schemas that could be extended or customized to the needs of the show, for more insight about the schemas and the data structures please refer to appendices A and B. the data is saved in xml files and extracted on the fly into objects or the other way around using serialization so each schema has corresponding object class. For example, the interaction data store schema has parallel class see appendix C. The interaction and scenario XML documents are small (quick to serialize) and hold only the necessary data. Illustrations of the XML documents contents are provided in appendix D.

6.7 Graphical User Interface

CPS Version 2.0 has been empowered by a friendly graphic interface that allows the artist to design new interaction scenarios, see Figure 19, and save them, see Figure 20. To set the variables of a show, the interaction zones and the different sequences of the DVD content to be shown, to launch and stop the show when necessary, see Figure 21.

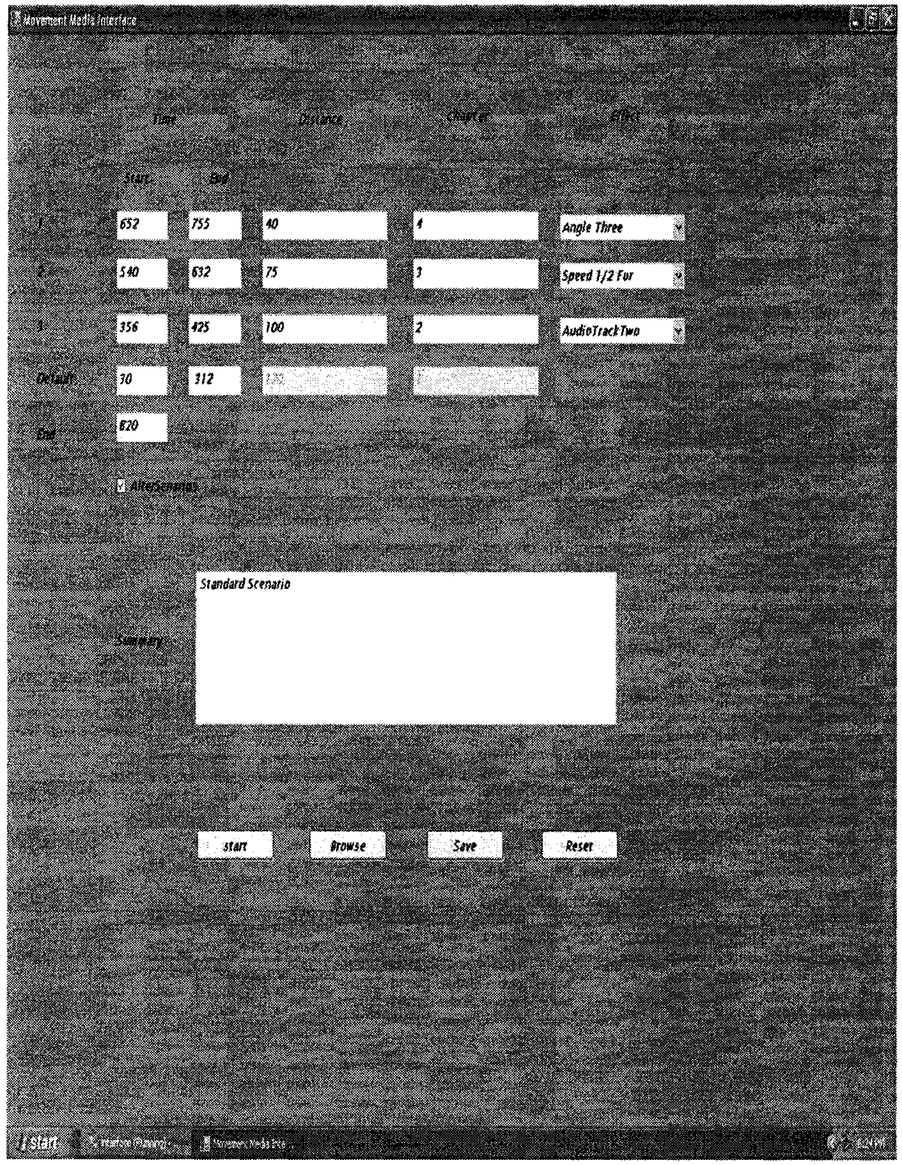


Figure 19: CPS Artist Interface

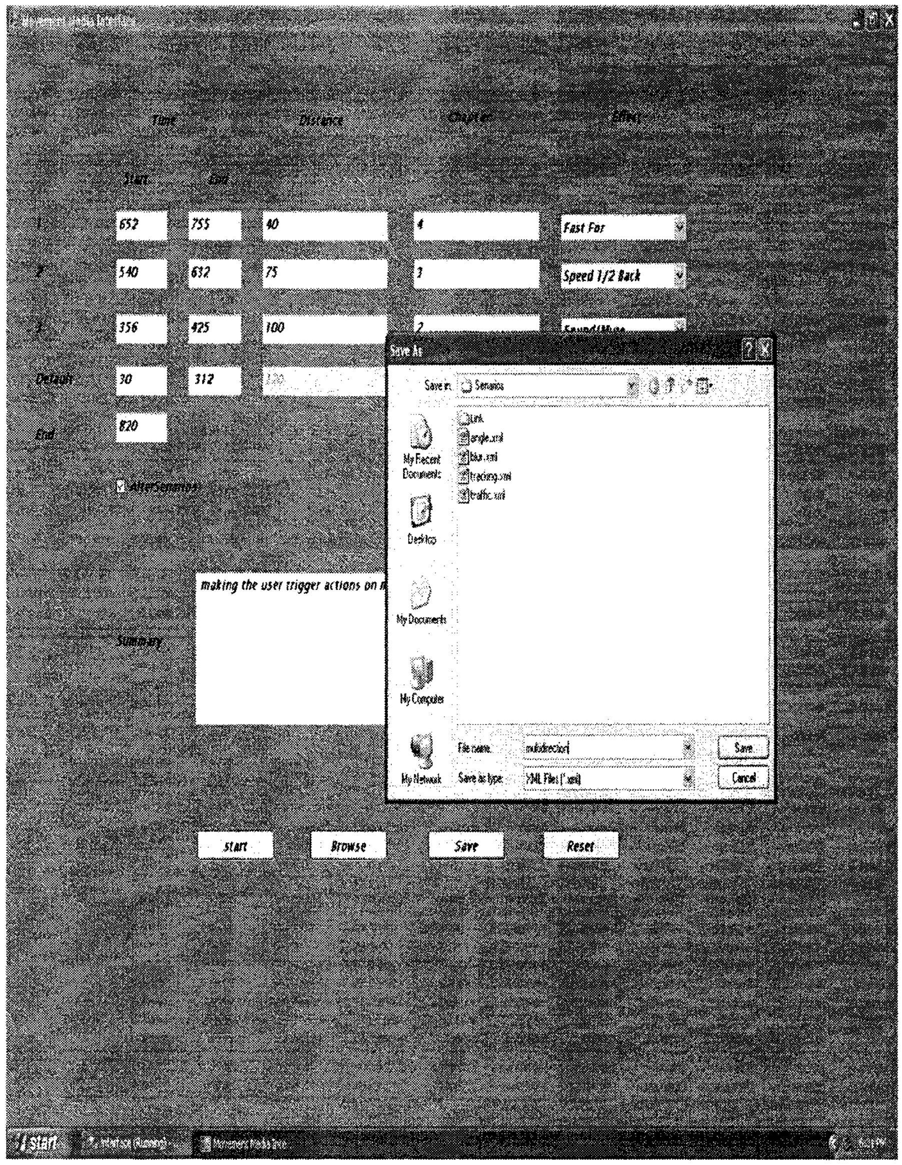


Figure 20: CPS Artist Interface: Saving Procedure

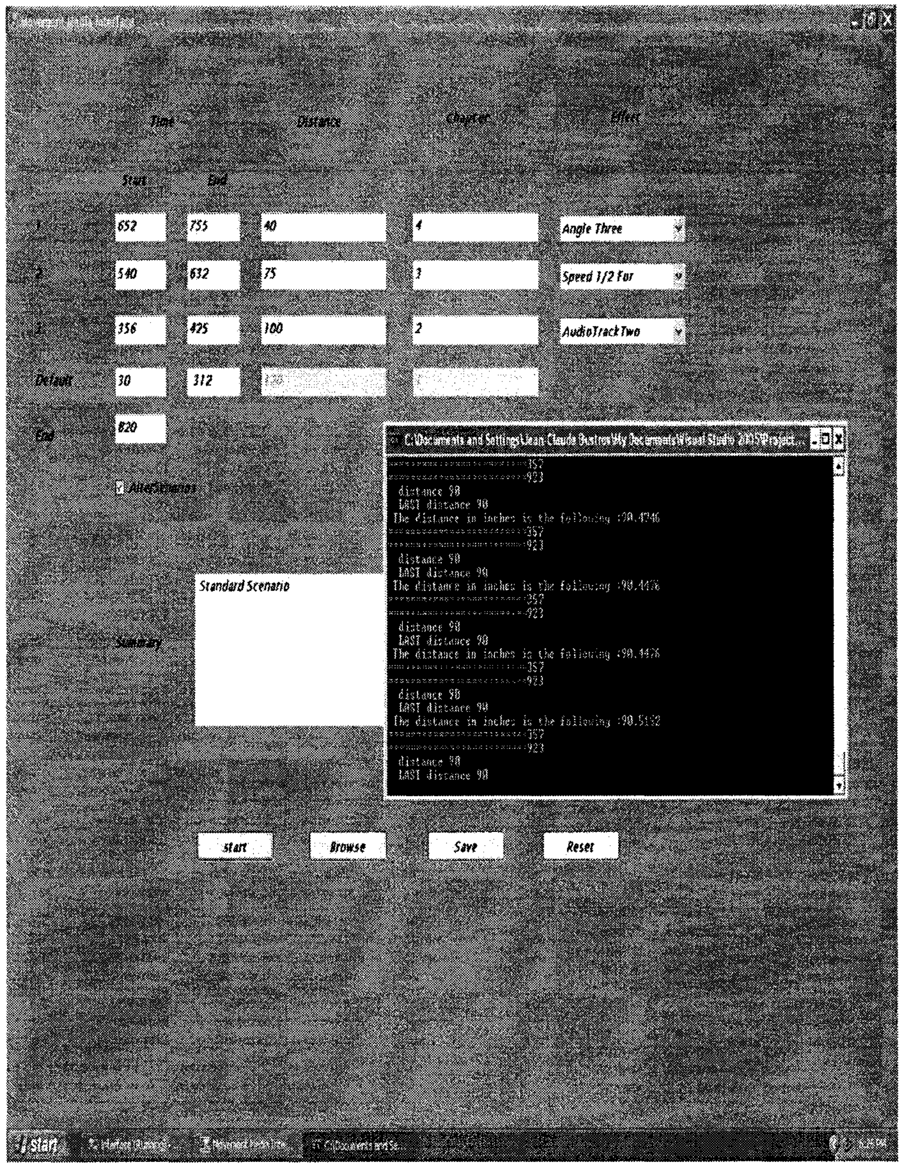


Figure 21: CPS Artist Interface, Start Procedure

Appendix A

Schema Structures

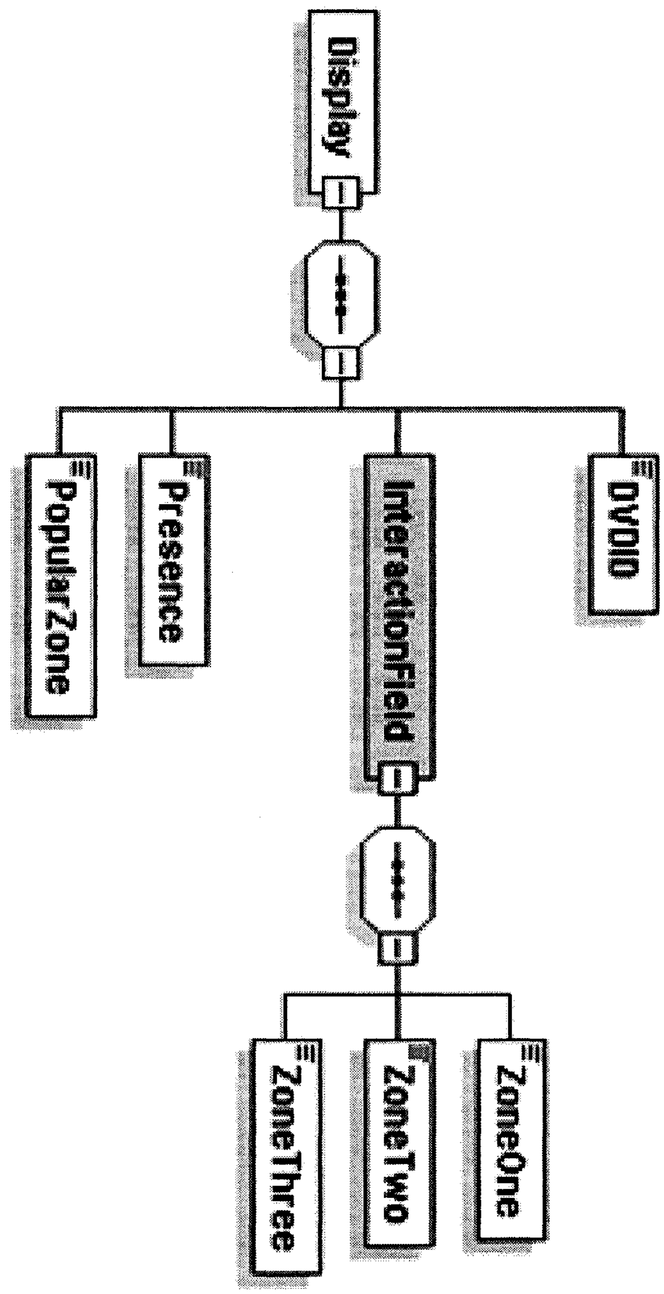


Figure 22: Data store interaction structure

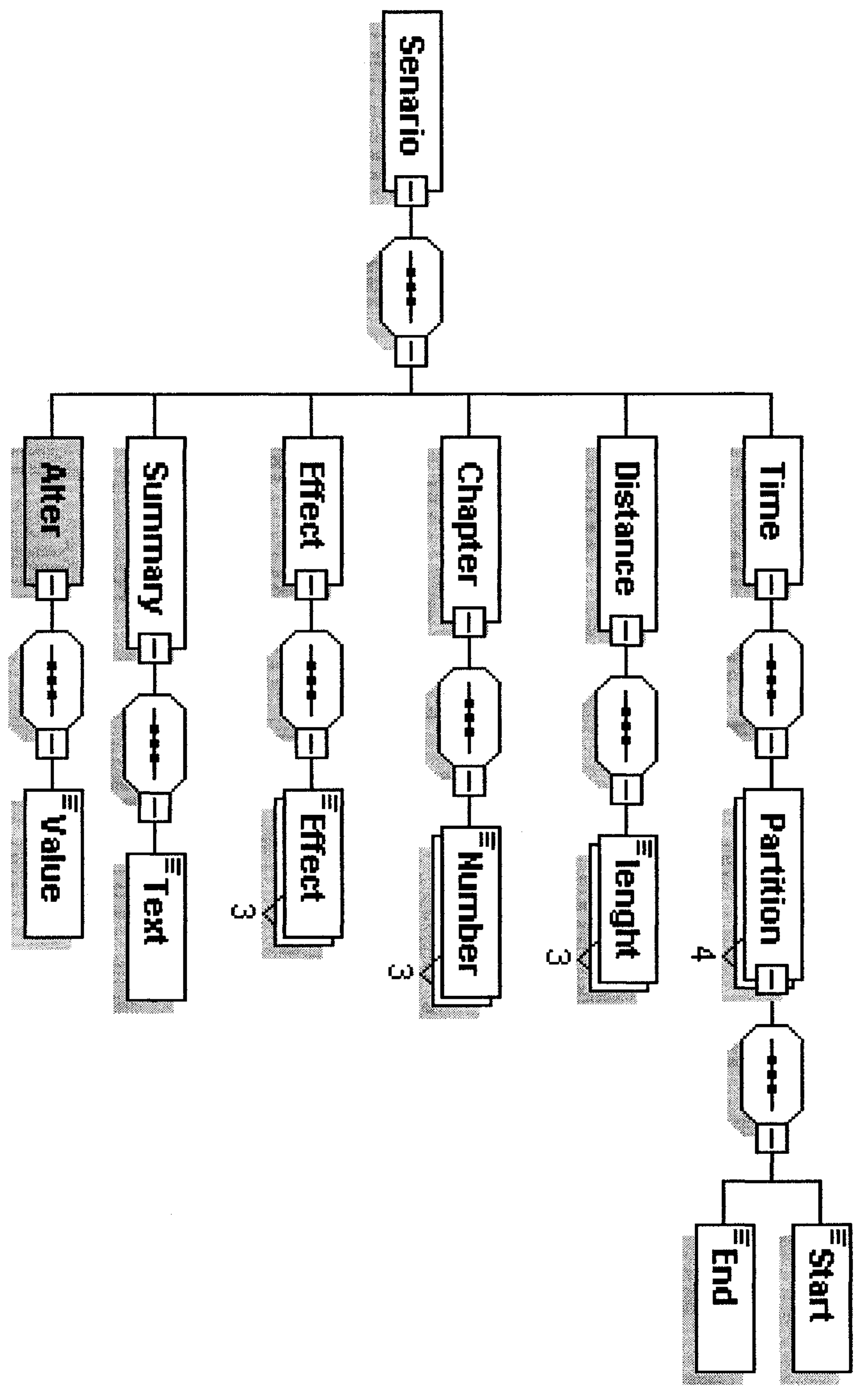


Figure 23: Senario schema structure

Appendix B

Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  - <xs:element name="Display">
    - <xs:complexType>
      - <xs:sequence>
          <xs:element name="DVDID" type="xs:string" />
        - <xs:element name="InteractionField">
          - <xs:complexType>
            - <xs:sequence>
                <xs:element name="ZoneOne" type="xs:integer" />
                <xs:element name="ZoneTwo" type="xs:integer" />
                <xs:element name="ZoneThree" type="xs:integer" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="Presence" type="xs:boolean" />
          <xs:element name="PopularZone" type="xs:integer" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

Figure 24: Interaction dataStore schema

```
<?xml version="1.0" encoding="UTF-8" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  - <xs:element name="Senario">
    - <xs:complexType>
      - <xs:sequence>
        - <xs:element name="Time">
          - <xs:complexType>
            - <xs:sequence>
              - <xs:element name="Partition" minOccurs="4"
                  maxOccurs="4">
                - <xs:complexType>
                  - <xs:sequence>
                      <xs:element name="Start" type="xs:integer" />
                      <xs:element name="End" type="xs:integer" />
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        - <xs:element name="Distance">
          - <xs:complexType>
            - <xs:sequence>
                <xs:element name="lenght" type="xs:integer"
                  minOccurs="3" maxOccurs="3" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        - <xs:element name="Chapter">
          - <xs:complexType>
            - <xs:sequence>
                <xs:element name="Number" type="xs:integer"
                  minOccurs="3" maxOccurs="3" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        - <xs:element name="Effect">
          - <xs:complexType>
            - <xs:sequence>
                <xs:element name="Effect" type="xs:integer" minOccurs="3"
                  maxOccurs="3" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        - <xs:element name="Summary">
          - <xs:complexType>
            - <xs:sequence>
                <xs:element name="Text" type="xs:integer" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
```

```
- <xs:element name="Alter">
  - <xs:complexType>
    - <xs:sequence>
        <xs:element name="Value" type="xs:boolean" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Figure 25: Interaction senario schema

Appendix C

Schema generated class

```
using System.Xml.Serialization;
/// <remarks/>
[System.CodeDom.Compiler.GeneratedCodeAttribute("xsd", "2.0.50727.42")]
[System.SerializableAttribute()]
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.ComponentModel.DesignerCategoryAttribute("code")]
[System.Xml.Serialization.XmlTypeAttribute(AnonymousType=true)]
[System.Xml.Serialization.XmlRootAttribute(Namespace="", IsNullable=false)]
public partial class Display {

    private string dVDIDField;

    private DisplayInteractionField interactionFieldField;

    private bool presenceField;

    private string popularZoneField;

    /// <remarks/>
    public string DVDID {
        get {
            return this.dVDIDField;
        }
        set {
            this.dVDIDField = value;
        }
    }

    /// <remarks/>
    public DisplayInteractionField InteractionField {
        get {
            return this.interactionFieldField;
        }
        set {
            this.interactionFieldField = value;
        }
    }

    /// <remarks/>
    public bool Presence {
        get {
            return this.presenceField;
        }
        set {
            this.presenceField = value;
        }
    }

    /// <remarks/>
    [System.Xml.Serialization.XmlElementAttribute(DataType="integer")]
    public string PopularZone {
        get {
            return this.popularZoneField;
        }
        set {
            this.popularZoneField = value;
        }
    }
}

/// <remarks/>
[System.CodeDom.Compiler.GeneratedCodeAttribute("xsd", "2.0.50727.42")]
[System.SerializableAttribute()]
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.ComponentModel.DesignerCategoryAttribute("code")]
[System.Xml.Serialization.XmlTypeAttribute(AnonymousType=true)]
public partial class DisplayInteractionField {
```

```
    private string zoneOneField;

    private string zoneTwoField;

    private string zoneThreeField;

    /// <remarks/>
    [System.Xml.Serialization.XmlElementAttribute(DataType="integer")]
    public string ZoneOne {
        get {
            return this.zoneOneField;
        }
        set {
            this.zoneOneField = value;
        }
    }

    /// <remarks/>
    [System.Xml.Serialization.XmlElementAttribute(DataType="integer")]
    public string ZoneTwo {
        get {
            return this.zoneTwoField;
        }
        set {
            this.zoneTwoField = value;
        }
    }

    /// <remarks/>
    [System.Xml.Serialization.XmlElementAttribute(DataType="integer")]
    public string ZoneThree {
        get {
            return this.zoneThreeField;
        }
        set {
            this.zoneThreeField = value;
        }
    }
}
```

Figure 26: Interaction datastore schema generated class

Appendix D

Sample XML Documents

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Display xsi:noNamespaceSchemaLocation="InteractionDataStoreSchema.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <DVDID>A</DVDID>
  - <InteractionField>
      <ZoneOne>54</ZoneOne>
      <ZoneTwo>30</ZoneTwo>
      <ZoneThree>41</ZoneThree>
    </InteractionField>
    <Presence>true</Presence>
    <PopularZone>1</PopularZone>
  </Display>
```


Figure 27: DataStore interaction senario XML

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Senario xsi:noNamespaceSchemaLocation="SenarioSchema.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <Time>
    - <Partition>
        <Start>259</Start>
        <End>333</End>
      </Partition>
    - <Partition>
        <Start>211</Start>
        <End>244</End>
      </Partition>
    - <Partition>
        <Start>129</Start>
        <End>142</End>
      </Partition>
    - <Partition>
        <Start>42</Start>
        <End>414</End>
      </Partition>
    </Time>
  - <Distance>
      <lenght>40</lenght>
      <lenght>75</lenght>
      <lenght>100</lenght>
    </Distance>
  - <Chapter>
      <Number>1</Number>
      <Number>2</Number>
      <Number>3</Number>
    </Chapter>
  - <Effect>
      <Effect>14</Effect>
      <Effect>8</Effect>
      <Effect>8</Effect>
    </Effect>
  - <Summary>
      <Text>Sample testing</Text>
    </Summary>
  - <Alter>
      <Value>true</Value>
    </Alter>
  </Senario>
```


Figure 28: Interaction senario XML

Bibliography

[Agrawala *et al.*1995] Agrawala, Beers, and Levoy. Symposium on interactive 3d graphics. *3d painting on scanned surfaces*, 1995.

[Ayala *et al.*2003] P. Ayala, I. Barandiaran, D. Vicente, and M. Grana. Exploring simple visual languages for real time humen-computer interaction. In *VECIMS 2003 - International Symposium on Virtual Environments, Human-Computer Interface, Nad Measurement Systems*, 2003.

[Aylett and Cavazza2001] Aylett and Cavazza. Intelligent virtual environment: a state-of-the-art report. *Eurographics*, 2001.

[Bimber *et al.*2002] Bimber, Gatesy, Witmer, Raskar, and Encarnao. Merging fossil specimens with computer-generated information. *IEEE Computer*, 2002.

[Boening *et al.*2006] Guido Boening, Klaus Bartl, Thomas Dera, Stanislavs Bardins, Erich Schneider, and Thomas Brandt. Mobile eye tracking as a basis for real-time control of a gaze driven head-mounted video camera. *Department of Neurology, Ludwig-Maximilians University of Munich, Germany*, 2006.

[B.Reilly1998] Richard B.Reilly. Applications of face and gesture recognition for human-computer interaction,dept. of electronic and electrical engineering, university college dublin, belfield, dublin 4, ireland. *ACM*, 1998.

[Cavazza *et al.*2004] Cavazza, Lugrin, Hartley, Libardi, Barnes, Le bars, Le Renard, Bec, and Nandi. New ays of worldmaking: the alterne platform for vr art. new york, usa. *ACM Multimedia*, 2004.

[Cavazza *et al.*2005] Marc Cavazza, Jean-Luc Lugrin, Simon Hartley, Marc Le Renard, Alok Nandi, Jeffrey Jacobson, and sean Crooks. Intelligent virtual environments for virtual reality art. *ELSEVIER*, 2005.

[Debevec1998] Debevec. Rendring synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. *Computer Graphics Annual Conference Series*, 1998.

[D.Morrison2005] Gerald D.Morrison. A camera-based input device for large interactive displays. In *IEEE Computer Society*, 2005.

[Foskey *et al.*2002] Foskey, Otaduy, and Lin. Art-nova: Touch-enabled 3d model design. *IEEE Virtual Reality Conference*, 2002.

[Grasset *et al.*2005] Raphael Grasset, Laurence Boissieux, Jean-D. Gascuel, and Dieter Schmalstieg. Interactive mediated reality. *HITLAB NZ University of Canterbury, Private Bag 4800, Christchurch, New Zealand. ARTIS/GRAVIR,INRI Rhone-Alpes,655, Avenue de l'Europe, 38330 MontBonnot, France. Interactive Media System Group, Vienna University of Technology, Favoritenstrasse 9-11/188/2,A-1040,Vienna, Austria*, 2005.

[Grau2003] Grau. Virtual art: from illusion to immersion. *Cambridge: MIT Press*, 2003.

[IKEDA *et al.*2003] Sei IKEDA, Masayuki KANBARA Tomokazu Sato, and Naokazu Yokoya. Telepresence system using high-resolution omnidirectional movies and a reactive display,vision and media computing laboratory nara institute of science and technology 8916-5 takayama,ikoma, nara 630-0192, japan. 2003.

[I.MacKenzie and C.Ware1993] I.MacKenzie and C.Ware. Lag as a determinant of human performance in interactive systems,human factors in computing systems. 1993.

[Jacobson and Hwang2002] J Jacobson and Z Hwang. Unreal tournment for immersive interactive theater. *Communication of the ACM*, 2002.

[Jun *et al.*2006] LIU Jun, CHEN Yi-qiang, XIE Chen, and GAO Wen. Ssc: Gesture-based game for initial dementia examination. *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China and Graduate School of the Chinese Academy of Sciences, Beijing 100080, China and College of Software Technology, Zhejiang University, Hangzhou 310027, China*, 2006.

[Kallio *et al.*2003] Sanna Kallio, Juha Kela, and Jani MBntyjBni. Online gesture recognition system for mobile interaction. Technical report, VTT Technical Research Centre of Finland, 2003.

[Kalnins *et al.*2002] Kalnins, Markosian, Meier, Kowalski, Lee, Davidson, and Hughes Webb, and Finkelstein. Wysiwyg npr: drawing strokes directly on 3d models. *ACM SIGGRAPH*, 2002.

[Liu *et al.*2003] Yong Liu, Moritz Storring, Thomas B. Moeslund, Claus B. Madsen, and Erik Granum. Computer vision based head tracking from re-configurable 2d markers for ar. *IEEE*, 2003.

[Mohamed *et al.*2006] Mohamed, Haag, Peltason, Dal-Ri, and Ottmann. Disoriented pen-gestures for identifying users around the tabletop withoutcameras and motion sensors,institut fr informatik, albert-ludwigs-universitt freiburg georges-khler-allee, geb. 51, d-79110 freiburg, germany. *IEEE*, 2006.

[Morris2006] Meredith Ringel Morris. Supporting effective interaction with tabletop groupware,stanford university. *IEEE*, 2006.

[Moser1996] MA Moser. Immersed in technology: art and virtual environments. *Cambridge, MA: MIT Press*, 1996.

[Ryokai *et al.*2004] Ryokai, Marti, and Ishii. I/o brush: Drawing with everyday objects as ink. *ACM CHI*, 2004.

[Schmalstieg *et al.*2002] Schmalstieg, Fuhrmann, Hesina, Szalavari, Encarnao, Gervautz, and Purgathofer. The studierstube augmented reality project. *Presence - Teleoperator and Virtual Environments*, 2002.

[Shinjo *et al.*2006] Hiroshi Shinjo, Ui Yamaguchi, Akio Amano, Konagi Uchibe, Atsushi Ishibashi, and Hideki Kuwamoto. Intelligent user interface based on multimodal dialog control for audio-visual systems. *Hitachi Review Vol. 55*, 2006.

[Torpus and Buhlmann2005] Jan Torpus and Vera Buhlmann. Lifeclipper. University of Basel, Vogelsangstrasse 15,CH-4058 Basel, Switzerland, 2005.

[V.Cheng and N.Kehtarnavaz2000] V.Cheng and N.Kehtarnavaz. A smart camera application: Dsp - based people detection and tracking,spie j. electronic imaging. 2000.

[Wu *et al.*2006] Mike Wu, Chia Shen, Kathy Ryall, Clifton Forlines, and Ravin Balakrishnan. Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. *Mitsubishi Electric Research Laboratories Cambridge, MA and Department of Computer Science University of Toronto*, 2006.