A Unified Multi Modal Framework for a Multiuser Multiple Virtual Environments in a Virtual Reality Centre

Dr P Jayarekha, Prameetha Pai, Dr Dakshayini M

Abstract: The use of ubiquitous devices for Pervasive computing and Virtual Environments (VE) for multi modal inputs is widely used technologies these days. This study focuses on the need of a Multi Modal framework for use of ubiquitous devices in a multiple Virtual Environment (experience zones) Virtual Reality (VR) center, allowing users to move from one experience zone to another without the need to swap their input devices as well as allowing for multi user environment with remote monitoring capability. This study elaborates the need for a unified multi modal framework for a multi user, multiple VE in a VR center and is largely based on i*Chameleon framework [1]. This study concludes with a viable approach that could be used to develop the required framework.

Index Terms: Human Computer Interaction, Multi modal input, Multi user, Remote monitoring, Ubiquitous devices, Virtual Environment, Virtual Reality

---

1. Introduction

The purpose of multimodal human computer interaction is to study how computer technology can be made more user-friendly, through the interplay among three aspects: user, system and interaction [2].The term "multimodal" has been used in many contexts across several research areas [3][4]. Sharon Oviatt has summarized the common myths associated with multi modal interactions [5]. Mark Weiser coined the term pervasive computing, the third wave in computing describing the creation of environments with communication capability, yet gracefully integrated with human users [6]. Today, mature distributed systems and well developed hardware are available for Virtual Environments. Virtual Environments extensively use motion capture, speech and gesture recognition devices along with data gloves, VR peripherals like wand, 3D mouse, joysticks, etc. With proliferation of smart phones and tablets, multi touch input system along with reasonable computing capabilities makes pervasive computing / interaction on these ubiquitous devices very intuitive way to navigate around / use the Virtual Environment.

However, in a large VR center with multiple Virtual Environments or multiple interactive softwares available on a single immersive VE then the user should not be asked to switch their devices to interact with the new VE software. This calls for a unified framework to support multiple input devices in a multi user environment with dynamic loading of meta data related to the simulated VE on the pervasive computing devices.

This paper is organized as follows: Section 2 highlights the desirable system features and requirements. Section 3 presents a sample of existing systems &prior work in this area, followed by proposed system in Section 4. System architecture is explained in Section 5 followed by results in section 6. Brief concluding remarks are given in Section 7.

2. System Requirements

Generally, the software for Virtual environments (figure 1) are designed such that a controller application interfaces with the input devices, a UI application interfaces with the User / Administrator and a simulation engine interfaces with these two applications and the 3D graphics simulation software.

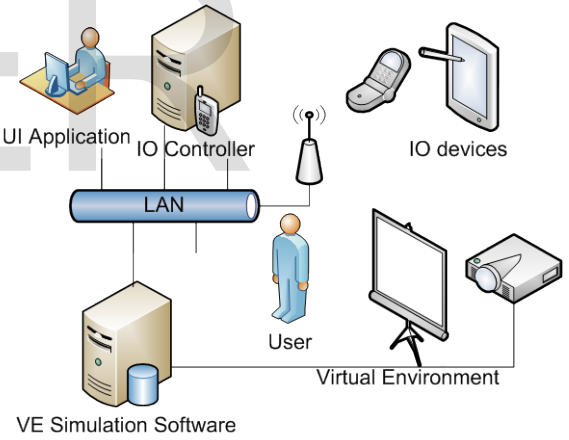

Figure 1: VE software deployment

The above deployment architecture calls for a very tight coupling between the UI application software, IO controller and simulation engine. The coupling also exists between the IO devices and the controller application. This tight coupling does not allow new IO devices to be added dynamically, nor will the simulation engine allow multiple users to connect and disconnect dynamically while switching from one VE to another. Keeping these constraints in mind, we highlight the desirable system features and requirements for a multimodal framework. Based on these requirements and prior work in this field, we propose our solution. Some of the important features of this system are as follows

A.Heterogeneity[1]

One key theme in multimodal human computer interaction is the integration of different modalities into a single

system [7]. A multimodal framework should provide a standardized protocol to utilize a variety of widgets ranging from traditional keyboard and mouse, game controller, motion capture camera to future devices.

B. Pervasion

Rather than a tightly coupled UI application with the simulation software, a mechanism is required where any ubiquitous device within the VE can open up the UI and interact without installing any new software.

C. Real time interaction

The pervasive components should not hinder real time interaction. The selected communication protocol should ensure real time communication well within the acceptable response delays. The packet size as well as load on the computing should be as low as possible.

D. Multi User support

The framework should ensure a multi user environment along with dynamic movement between Virtual Environments (logging in & out). It should also ensure that new sessions are created for every new user and there is no crosstalk between users.

E. Scalability[1]

As pervasive computing continues to gain prominence, the variety in types and number of widgets to be supported will continue to grow. The framework should be able to support large number of widgets of the same modality and/or different modalities, to properly support distributed and ubiquitous components and user collaboration.

F. Remote Monitoring

The framework should also support non pervasive interactions over the internet for support, reports, logs, etc.

G. Extensibility and Flexibility[1]

Regional factors, time constraints, personal preferences, etc., affect the way users interact with applications. The framework should allow developers to utilize the interaction implementation that best fits. It should also be domain independent, minimizing the changes required to adapt to another domain [8].

H. Deployment and Configuration[1]

The framework should be based on industry-standard and hardware-independent protocols, so that developers can deploy the framework to the application server easily. The environment should allow even non-programmers to create widgets and customize the configuration without prior knowledge of the background process, through the use of a drag-and-drop graphical user interface [9].

3. Existing Systems

Over the years many frame works have been developed for multi modal inputs. Of these the well-known multimodal systems are ICARE [10] and OpenInterface [11]. OpenInterface is a component-based tool for rapidly deploying multimodal input for different kinds of applications. The framework includes a kernel, a builder and a design tool. After coding their own multimodal application and device drivers, developers will port and deploy them on the platform. However, it does not support plug-and-play and cannot readily support collaborative interaction over the Internet.

Numerous experimental virtual reality systems and multiplayer games have been developed for real time interaction in shared virtual environments. Unfortunately, most existing systems do not scale well to large numbers of simultaneous Real Time applications Reality Built For Two [12], VEOS [13], and MR Toolkit [14] are multi-user virtual reality systems that maintain consistent, states among N workstations by sending a point-to-point message to each of N-l workstations whenever any entity in the distributed simulation changes state. This approach yields $O(N^2)$ update messages during every simulation step, and thus does not scale to many simultaneous users before the network gets saturated.

SIMNET [15], NPSNET [16] use broadcast messages to send updates to all other workstations participating in a virtual environment. Although, this approach cuts down on the total number of messages transmitted to O(N), every workstation still must process a message whenever any entity in the distributed simulation changes state. Since every workstation must store data and process update messages and/or simulate behavior for all entities during every simulation step, these systems do not scale beyond the capabilities of the least powerful participating workstation.

Promising and intuitive interaction methods with heterogeneous input devices and interaction over the web browser have been developed by W3C. However, there remains much room for the development of multimodal interaction for pervasive systems [1].

i*Chameleon[1] framework, allows users to customize their interaction, capable of adapting to changes in the environment as well as user requirements according to the features of applications or the needs of different users. i*Chameleon focuses on the design considerations of multimodality for pervasive computing. This web-services based framework with a separate analytic co-processor for collaborative multimodal interaction provides a standard and semantic interface that facilitates the integration of new widgets with a wide variety of computer applications.

4. Proposed Systems

i*Chameleon framework architecture comes closest to meet our requirements, but it does not address our requirement to manage multi user simulation sessions, remote monitoring of the VR centre over the internet, browser based dynamic UI downloaded from the simulation engine to hop from one VE to another.

We propose to follow web services architecture itself with extensions to support remote monitoring and development of a manager to support multi user sessions. To support ubiquitous devices for touch based UI interactions it is proposed have XML based descriptor files (meta data) for each of the simulation modules. These descriptor files will aid in dynamic configuration of web pages that are accessible by any of the devices that has a browser.

5. System Architecture

The system comprises of the following components :

WebPages: These are hosted on a WebServer which are accessed via a browser. These webpages renders/display data retrieved from the Database and the Simulation Engine

HttpHandler : HttpHandler intercepts commands to the Simulation Engine. HttpHandler communicates the commands received from the Client Device to the Simulation Engine.

Database : The metadata is stored in the database.

Simulation Engine : This is the runtime which provides a base library and api's to enable visualization.

FileServer/Shared Drive : This is a location where the data related to the virtualization is stored. This data is stored in the system as Xml files.

Client Device : This is a client machine where the user primarily interacts with the system.

Input Device : This is a input device which the user provides inputs to the system. Eg keyboard, Mouse, JoyStick

The client device interacts with the Simulation Engine via a WebServer which intercepts commands to the Simulation Engine. The client device communicates with the WebServer over HTTP protocol. The data received by the WebServer from the Client is persisted into the Database. A socket is opened by the Simulation Engine. Commands received by the WebServer are translated and communicated with the Simulation Engine via a Socket. Simulation Engine receives data sent to the Socket by the WebServer. The Simulation Engine executes the commands received resulting in action on the Simulation Runtime. The System has a database which contains metadata of the simulation runtime. This metadata is used by the Simulation Engine and the WebServer to identify the virtualization environment which needs to be loaded on the Server. The metadata stored in the Database is the data identifiers about the simulation runtime stored on the File Server/Shared Drive. The user actions on the simulation runtime is traced by the simulation engine for every action executed and persists the trace record into an xml file. The usage data generated by the Simulation Engine is made available to the user via reports.

Figure 2 depicts the basic architecture of the proposed system

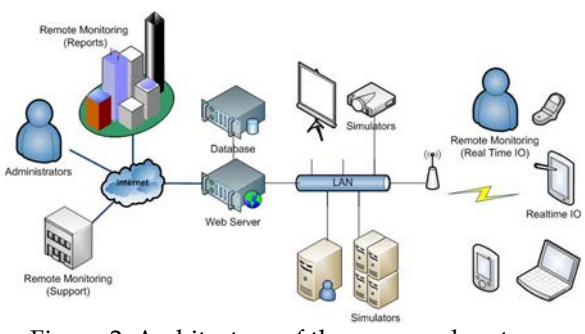

Figure 2: Architecture of the proposed system

Commands communicated to the Simulation Engine are

1. Rotate Camera
2. Pan Camera
3. Zoom In/Out

These commands are available as user actions on the WebPages. The user performs an action on the webpage which in turn posts a command to the Simulation Engine. The HTTPHandler intercepts the command and communicates the command to the Simulation Engine. The Simulation Engine executes the command resulting in the change in state of the simulation runtime visible as an action executed on the Monitor.

6. Results

In this product the web services and co-processor portions of the framework were deployed on a desktop with an Intel i3 3.07GHz processor and 4GB RAM while the client is running on a Samsung tablet. The internet information service (IIS) is configured first. Next asp.net was used to host a webpage. The advantage of doing this was that a user's device can be browser independent i.e the product can be used by a user having a tablet with any browser. Also this was not developed as a application hence there is no need for the user to download anything. All he has to do is use an existing browser and give the website name and use the simulator directly.

Keeping security issues in mind authentication of a user has also been incorporated. Once the user requests for the website a user login page is displayed as in figure 3

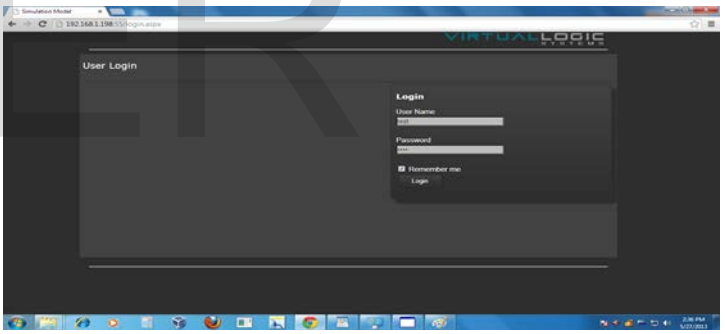

Figure 3:Login Page

Following authentication are done:

A. Username and password authentication
B. If wrong user name or password is entered appropriate message is displayed
C. If correct, Then it is checked if the DLSI background application(server) is running
D. If not, appropriate message is displayed to the user.
E. If yes, check if other users are using the system currently.
F. If yes, appropriate Message is displayed to the user.
G. If not, the user is allowed to login.

Once user logs in, he will be taken to the Home page. No simulation data is displayed as simulation is not selected.

User can select the desired simulation from the list of existing simulations in the database .Once he selects a particular

simulation, the appropriate Graphics Engine will be loaded in the server using socket communication. If the background application(server), which listens to the socket command to launch the graphics engine, is not running, then the user will be alerted and asked to re login to establish the connection with the server. User will be taken to the home page with all the data available of the selected simulation

a.Flythrough
b.Viewpoint
c.Scene
d.Animation

User can send commands for camera and object navigations and appropriate actions will be taken in the GE.If the graphics engine is not running hen the message will be displayed to the user. The user can save the changes he has made to the simulation.

Often, the primary problems that is reported after field release of a product are not system crashes or incorrect system responses, but rather system performance degradation or problems handling required system throughput. Nonetheless , software performance testing becomes an extremely significant issue for many large industrial projects. When queried, it is not uncommon to learn that, although the software system has gone through extensive functionality testing, it was never really tested to assess its expected performance. Keeping these aspects in mind this product is checked for software performance .At a given time this tangible widget (tablet) Can trigger 3 input signals (zoom in , zoom out, pan) we randomly generated one of the three input signals to the application server continuously. As a result, a certain number of events are created for the next 60 seconds when the throughput reaches the maximum. Considering a normal rate of sensory input widgets it was seen that the application server was capable of supporting atleast 20 widgets simultaneously. In multimodal interaction this implies that the framework can support a collaborative working environment accommodating six to seven different users.

7. Conclusion

In this study we have concluded that web services based i*Chameleon framework comes closest to our requirements for a unified multi modal framework for a multi user, multiple Virtual Environments in a VR center. Hence we are planning to go ahead with the development of our framework based on the web services architecture similar to that of i*Chameleon, but with the necessary modifications to suit our requirements. This study does not intend to cover exhaustively all issues related to multi modal interfaces / frameworks nor does it address the issues related to networking, error handling etc. All these aspects can be the basis for future work.

References:

Reality Annual International Symposium ,September,1993,450-455

[16] Zyda, Michael J., David R. Pratt, John S. Falby, Chuck Lombardo, and Kristen M Kelleher, The software Required for the computer Generation Of Virtual Environment March 1993

[1] K.W.K. Lo, W.W. Tang, H.V. Leong, A.T.S. Chan, S.C. Chan, and G. Ngai, "i*Chameleon: A unified web service framework for integrating multimodal interaction devices" ,in Proc. PerCom Workshops, 2012, pp.106-111.

[2] A. Jaimes and N. Sebe, "Multimodal human-computer interaction: a survey". Computer Vision and Image Understanding, 108(1-2):116-134, 2007.

[3] N.O. Bernsen, "Foundations of multimodal representations: a taxonomy of representational modalities". Interacting with Computers, 6(4):347-371, 1994.

[4] N.O. Bernsen, "Defining a taxonomy of output modalities from an HCI perspective". Computer Standards & Interfaces, 18(6-7):537-553, 1997.

[5] Sharon Oviatt, "Ten myths of multimodal interaction". ACM, Volume 42 Issue 11,Pages 74-81, Nov. 1999.

[6] M. Satyanarayanan, "Pervasive computing: vision and challenges". Personal Communications, IEEE, 8(4):10 -17, 2001.

[7] S.L. Oviatt, A. De Angeli, and K. Kuhn, "Integration and synchronization of input modes during multimodal human-computer interaction". In Proc. ACM CHI '97, pp. 415-422, 1997.

[8] A. Costa Pereira and F. Hartmann, and K. Kadner, "A distributed staged architecture for multimodal applications". In Proc. European Conference on Software Architecture, pp. 195-206, 2007.

[9] K. Henricksen, J. Indulska, T. McFadden, and S. Balasubramaniam,"Middleware for distributed context-aware systems". In Proc. OfCoopIS, DOA, and ODBASE, LNCS, pp. 846-863, 2005.

[10] S. Oviatt, "Advances in robust multimodal interface design". IEEE Computer Graphics and Applications, 23(5):62-68, 2003.

[11] J.L. Lawson, A.-A. Al-Akkad, J. Vanderdonckt, and B.M. Macq, "An open source workbench for prototyping multimodal interactions based on off-the-shelf heterogeneous components". In Proc. ACMSIGCHI Symposium on Engineering Interactive Computing System, pp. 245-254, 2009.

[12] Blanchard, C., S. Gurgess, Y. Harvill, J. Lanier, A. Lasko, M. Oberman, and M. Teitel, Reality Built for Two: A Virtual Reality Tool. ACM SIGGRAPH Special Issue on 1990 Symposium on Interactive 30 Graphics, (Snowbird,Utah), 1990, 35-36.

[13] Bricken, William and Geoffrey Coca The VEOS project. Technical Report, Human Interface Technology Laboratory,University of Washington

[14] Shaw, Chris, and Mark Green, The MR Toolkit Peers Package and Experiment. Proceedings of IEEE Virtual Reality Annual International Symposium.

[15] Calvin, James, Alan Dickens, Bob Gaines, Paul Metzger, Dale Miller, and Dan Owen, The SIMNET Virtual World Architecture. Proceedings of the IEEE Virtual

- *Dr Jayarekha P is Associate Professor, department of Information Science, BMS College Of Engineering, India.*
- *Prameetha Pai is currently pursuing Mtech in computer network engineering , BMS College of engineering, India, prameetha@gmail.com*
- *Dr Dakshayini M is Professor , department of Information Science, BMS College of engineering, India.*

- *Dr Dakshayini M is Professor, department of Information Science,BMS College Of Engineering, India.*