Eurographics Italian Chapter Conference (2007)
Raffaele De Amicis and Giuseppe Conti (Editors)

Flexible Multimodal Architecture for CAD Applications

M. Dellisanti[1], M. Fiorentino[1], G. Monno[1], A.E. Uva[1]

[1]POLITECNICO DI BARI, Bari, Italy

Abstract

Multimodal interfaces are proved to increase the input\output bandwidth for a more efficient interaction in 3D activities. This work presents a novel framework designed to simplify the configuration procedures and to provide a high level of flexibility in multimodal interfaces. We propose an innovative system architecture to decouple input and output configuration from the application level and a multimodal engine designed to manage and synchronize the different inputs. The flexibility of the system is ensured by the use of a hierarchical XML-based configuration structure. This framework can be applied to test the advantage of multimodal interfaces in VRAD applications and provides tools to explore the synergy of different input\output techniques

Categories and Subject Descriptors: H.5.2 [User Interfaces]: Interaction styles; J.6 [Computer-Aided Engineering]: Computer-aided design (CAD)

1. Introduction

During the last 10 years CAD applications have improved and diversified (so called CAX software and PDM) to cover the entire product lifecycle. However, current available CAD systems are based on desktop 2D interface, despite the modelled geometry is three dimensional and usually augmented with extended information, multimedia attributes and other forms of knowledge.

Multimodal Interfaces (MI) and Virtual Reality (VR) are candidate to be the ideal interface for complex geometry modelling and product data management. Such interfaces put on practice the so called pervasive computing, which simulates human-like sensory perception by interpreting continuous simultaneous input from different sources such as human speech, pen input, gesture, gaze, and other natural behaviours. Two advantages are achieved: first the interface benefits from natural human skills, secondly, combining different input modes, the strengths of each modality match the weaknesses of the others. Multimodal interface have been proved to get better input accuracy and reduce task completion time [HHR06]. For example, voice input, differently from gesture, is not related to a specific spatial reference, so it can be used regardless the specific visual output. This is particularly important in CAD applications where geometries may be complex, not completely visualized, occluded by others objects or even hidden.

On the other hand, tangible input devices, such as real push buttons, sliders, wheels etc. can provide higher precision and control, but with some drawbacks: weight, visual occlusion, connections, cables, etc. By a synergic combination of different types of input the CAD developer can maximize the efficiency of the interface and broadening the range of tasks that can be done in an intuitive manner.

In the last few years, Virtual Reality became a reliable technology and currently the market proposes a large number of input\output devices e.g. multi-wall systems, theatres, power walls, holo-benches, Augmented and Mixed Reality displays, haptic rendering systems, etc. This extreme diversification of devices makes the development of a Virtual Reality multimodal application a challenging task because most of the time is spent in testing and debugging different set-ups.

We developed a Virtual Reality multimodal CAD application (VRAD) [FdASM02] and we faced system configuration problems in switching different input and output modalities. A large number of parameters should be set for developing and optimizing the interface, for adapting the system to subjective preferences (i.e. user learning curve and skills) and to specific CAD tasks (i.e. sketching, object data visualization, model navigation, etc.) [FMU04].

This work presents a novel framework designed to simplify the configuration procedures and to provide a higher grade of flexibility in multimodal interfaces. We worked on three different levels:

- a new system architecture has been developed to decouple input and output configuration from the application level;
- a multimodal engine has been designed to manage and synchronize the different inputs (devices, data type, sampling rate, etc.);
- a configurable XML based layer has been implemented for managing complex user configurations in a hierarchical way.

2. Related Works

One of the first contributions to address multimodal input was proposed by Vo and Weibel [VW97]. They developed a Java based framework and a grammar structure for a broad class of multimodal applications.

A relevant approach to multimodal VRAD was proposed by Gadh and al. [CDG97] [AG00]. They presented an intuitive and natural interface for detailed shape modeling using hand motions, gestures and voice commands.

An interesting multimodal CAD interface was presented by Adler and Davis [AD04]. Their work studied the behaviour of people while modelling in 2D using sketching and speaking. They derived a set of rules for segmenting and aligning the signals from both modalities, with the result of a more natural interface.

Milota in [Mil04] studied the general strategies employed in pen interactions when combining the speech and gesture channels.

Gutierrez et al. [GTV05] presented a system for real-time configuration of multimodal interfaces to Virtual Environments (VE). Their framework uses portable XML descriptors to define the I/O channels of a variety of interaction devices.

Although several approaches have been presented to address multimodal interaction configuration, there is a lack in literature of flexible and comprehensive architectures for VRAD systems.

3. System Configuration

A multimodal CAD application requires a large number of parameters to be set by the user before starting the application. The basic idea is to simplify the system configuration by an extensive use of XML files. We divided the application in four functional blocks which are configured separately: input, output, system and application. One main configuration file loads each sub-section in a hierarchical way as shown in Figure 1.

The *Input* module takes care of the input devices, which need hardware-specific parameters for a correct setup. To provide a higher level of abstraction, each device input is described by a special set of tags which declare what kind of services it is able to provide, i.e. type, stations, precision, working area, update-rate, etc.

Since no standard is available for 3D devices, we developed our XML Document Type Definitions (DTD). For example a tracking system can provide only position and rotation in the space, while a num-pad can send numbers.

The *Output* functional module deals with visual rendering of the virtual scene, and must be configured to accommodate the desired visualization mode (i.e. stereo or mono) and the type of display (multi side cave or desktop mode).

The *System* configuration file allocates low level parameters as: update rates, processor idle-time management, etc.

The *Application* configuration allows the user to change the control logic of the VRAD multimodal interface.

The use of XML configuration files as shown, delivers the flexibility to test different configurations and setups. This allows to use different input sources or to switch among different display modes. For example, it is possible to switch from a CAD Desktop Environment (mouse and keyboard) to a VR Immersive Environment with optical tracking, leaving application unchanged. This approach is very useful for the final user and for the developer which can test different configurations (also to create debug VR simulators with desktop devices) without re-compiling the application.

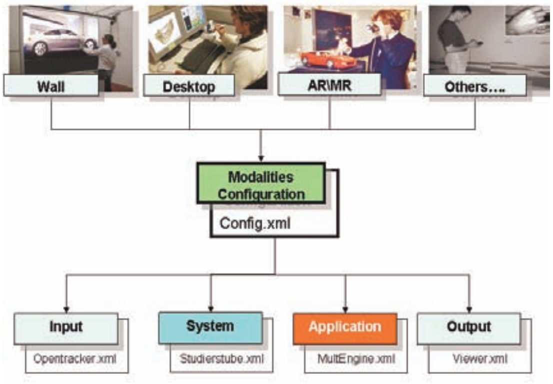

Figure 1. *General multimodal architecture.*

We would like to plug in our system a large number of existing input\output devices using OpenTracker library [RS05] and let the system choose the most appropriate one. The supported devices are properly grouped according to their functions.

3.1. Tracking

Several tracking systems can be combined mixing magnetic and optical technologies, letting the application switch from a system to another according to precision, range, availability, importance, etc. We defined a ranking for each available device, in order to choose the best source.

Our system supports almost all commercial tracking systems such as: ARToolKit, Polhemus Fasktrak, Flock of birds, Dtrack, etc..

3.2. Speech

Being the dominant channel of human communication, speech can have great potential in a CAD system. Speech input can be self-sufficient and thus, in theory, the only interface in a CAD, but many factors can lead to a high false command rate: workplace noises, user pronunciation, out of context dialogues, etc.

Therefore according to our experience, speech input must be restricted to a limited list of commands, in order to increase the reliability of the interface. The use of a small vocabulary will not hamper the user, because of the presence of others input commands. Our system is flexible and uses both Microsoft® Speech SDK and Dragon NaturallySpeaking® SDK.

3.3. Gesture

Pinchgloves by fakespace™ can be an alternative input by using fingertips movements. A quite large set of commands can be theoretically mapped to this device, but the drawback is that the user has to remember the association and physically wear the glove(s), which is not generally accepted for long working sessions. Future interfaces will use camera based contactless finger recognition systems.

3.4. Pen devices

A physical 3D mouse device is used as main interaction pointer in the CAD workspace. This metaphor, a natural extension from 2D desktop, is one of the most used in VR environments. The pointer, usually a pen-like tracked device is equipped with push buttons and other controls physically handled by the user. We developed a novel device called Senstylus, a wireless pen capable of rumble feedback and analogic input [FMU05]. Phantom® Omni by Sensable™, is supported as a desktop interface, offering also haptic capability.

3.5. Output

Our system is based upon Studierstube [SFH*02] VR library, thus very flexible in display settings. It supports all kind of visualization systems from multi-screen displays to HMD. The viewer configuration is performed by a single XML file and it allows different stereo and mono modes.

4. Framework Architecture

We implemented a CAD software architecture with the main goal of decoupling the modelling section from the multimodal input controller.

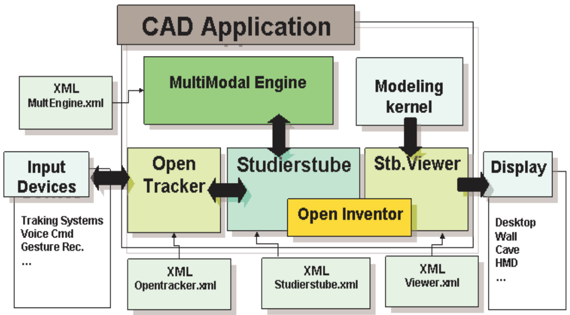

Figure 2. *Framework Architecture scheme.*

The proposed *CAD Application* is based on the *Studierstube library*. StudierStube is a framework for the development of VR applications developed at Graz University of Technology. It offers functions and components designed to manage the 3D scene (using to *OpenInventor* library), to interface with a large set of input devices (using the *OpenTracker* component), and to manage different display modes (using the *StbViewer* classes).

The OpenTracker component of the Studierstube library is a collection of “modules”. Each module is designed to manage the communication with a specific input device. Active devices are polled periodically to generate input events. Input events are transferred from OpenTracker to the *Multimodal Engine* through the Studierstube “Event Bus” system. The integration of new devices can be achieved by developing additional OpenTracker modules.

The StbViewer classes easily adapt the 3D rendering process to different displays, (such as Desktop displays, HMD, Wall and CAVE systems, etc.). The *Modeling Kernel* is the geometrical engine for the CAD operations. It provides an abstraction layer for interfacing with libraries like ACIS or OpenCascade. The scene graph is processed by the active StbViewer class which uses the *OpenInventor* library.

All the components can be configured by means of XML files as described in section 3.

5. Multimodal Engine

The Multimodal Engine is the core of our framework. It collects input events from real and virtual devices and routes them to the CAD Application (Figure 3). The routing rules are dynamically set using a knowledge based approach which brings a higher degree of configurability and effectiveness to the man-machine interaction.

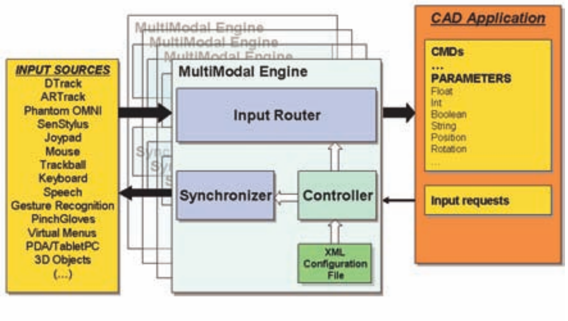

Figure 3. *MultiModal Engine Architecture.*

A set of *Input events* from physical or virtual devices is connected to the system. Each event carries attributes to notify the system about the kind of data provided (such as data type and data range).

The *CAD Application* exports a set of functionalities which can be triggered by commands (CMDs). Each functionality requires a specific set of parameters. The CAD Application notifies the MultiModal Engine about the actual required event (such as a command or a set of parameters). For each request, the CAD Application specifies the expected attributes.

The MultiModal Engine is made up of the following modules: the router, the controller and a synchronizer.

The *MultiModal Router* redirects the events received from the sources to the CAD Application. It performs typecasts or conversions, when needed.

The *MultiModal Controller* drives the router according to the requests of the CAD Application.

The Controller selects the event which best fits the request from the set of compatible inputs. It manages an in-memory database of rules which is loaded from an XML file. The Controller assigns a priority value to each of the compatible devices (Figure 4). This priority is the result of a "distance" calculation in the vector space of attributes (such as data type, precision, range, availability, etc.) between the CAD Application request and the available input source. The "closest" is the attribute, the higher is the priority. For example considering 6DOF tracking, a desktop device (like the Phantom Omni) can have a higher priority in a desktop virtual reality environment and a lower priority in an immersive hand-free virtual reality environment.

The *MultiModal Synchronizer* ensures consistency between different input devices which may fulfil the same CAD application request. As an example the display of a PDA input box must be synchronized with the number entered by speech recognition.

This architecture is also designed for a collaborative multiuser environment where each user logged in the system instantiates his/her own MultiModal Engine with user-defined configurations.

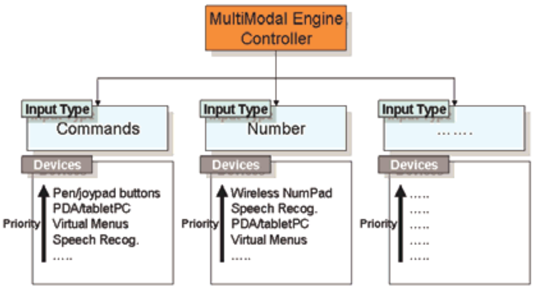

Figure 4. *Input types and compatible devices.*

6. Examples

In this section we present some examples to explain the behaviour of the multimodal engine while performing operations like receiving a command for the CAD application or entering a number. Finally we describe a simple CAD task like the creation of a box with given dimensions.

6.1. Command routing

Let's assume that the application is in idle state, waiting for a user command. This request is notified to the MultiModal Engine Controller which looks for input events of type "command" (Figure 4). This event type can be generated by several devices like: a physical push button, a gesture or speech recognition engine, as well as a virtual menu on a PDA/tabletPC. The Controller filters the incoming events and triggers the Input Router to connect the selected input source to the "CMD" input socket of the CAD Application.

6.2. Entering a number

It is common in a CAD Application to request the user to enter a number (dimensions, scale factors, rotation angles, etc.). In this case the request of a number is notified together with a list of expected attributes, such as data subtype (float or integer), precision, range, etc. The Controller browses its knowledge database and selects the events compatible with the request. Possible devices could be a numeric keypad or a virtual menu on a PDA/tabletPC as well as a speech recognition engine or a joypad slider or a mouse wheel. Since the CAD Application specifies the desired value of precision and a subtype, the Controller assigns a priority value to each of the compatible devices (Figure 4).

6.3. A simple test case

As an example we describe the use of the proposed architecture for a simple CAD task: creation of a box with given dimensions.

- The *CADApplication* is on idle and requires a command.
- The *MultiModal Engine* waits for the command from the available command source (see section 6.1) (buttons, virtual menus, voice, gesture, etc.), gets the command "box" from the virtual menu and sends it to the application.

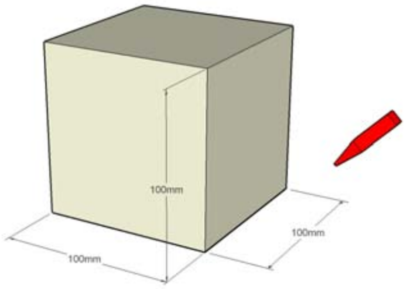

Figure 5. *Example: default box creation.*

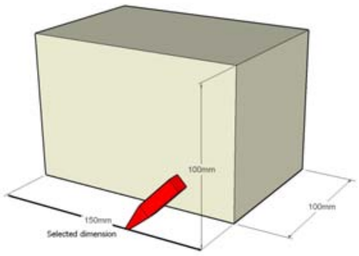

Figure 6. *Example: modify dimensions.*

- The *CADApplication* displays a cube with the default dimensions and the three quotes (Figure 5).
- The *CADApplication* is on idle and requires a command.
- The user sends a voice command "edit dimensions", it passes through the *MultiModal Engine* and reaches the *CADApplication*.
- The *CADApplication* requires the selection of one of the dimensions to be changed.
- The user can go with the pointer on the quote to be changed or select it from a menu or call an option ("width", "height" or "depth") vocally.
- The *CADApplication* requires a number (float) to change the dimension.
- The user can set the number with the modalities explained in section 6.2; he/she uses the wireless numpad.
- The *CADApplication* changes the dimensions and updates the visualization (Figure 6).
- The *CADApplication* requires the selection of one of the other dimensions to be changed or a command to exit the previous command .
- The *CADApplication* is on idle and requires a command.

Note that in our CAD application we decided to create a default cube and then modify its dimensions, but the drawing metaphor can be different; for example, the user can set two points to define the diagonal of a rectangle (Figure 7), and then extrude it on a perpendicular direction for a given length set with the same modalities (Figure 8).

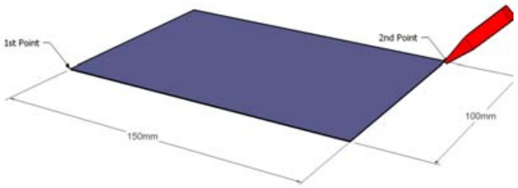

Figure 7. *Example: rectangle.*

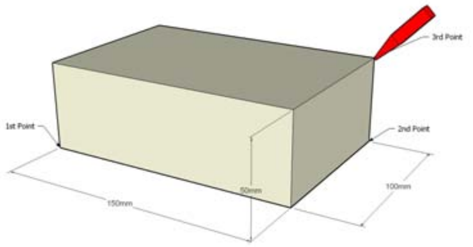

Figure 8. *Example: extrusion.*

7. Conclusions

We have designed a novel XML based framework for multimodal VRAD applications in order to simplify the configuration procedures and to provide a high level of flexibility. The main features of this architecture consist of:

- multimodal input decoupled from the CAD core;
- multimodal engine designed to manage and synchronize different inputs;
- hierarchical XML-based configuration.

This framework can be applied to test the advantage of multimodal interfaces in VRAD applications and it provides tools to explore the synergy of different input\output techniques.

8. Acknowledgements

We would like to thank Florin Girbacia (University of Transylvania, Romania) for implementing\testing the voice command module.

9. References

[AD04] ADLER A., DAVIS R.: *Speech and sketching for multimodal design*. Proceedings of the 9th international conference on Intelligent user interface (2004) http://people.csail.mit.edu

[AG00] ARANGARASAN R., GADH R.: *Geometric Modeling and Collaborative Design in a Multi-Modal, Multi-Sensory Virtual Environment* - Proceedings of DETC’00 ASME 2000 Design Engineering Technical Conferences and Computers and Information in Engineering Conference Baltimore, Maryland, (September 10-13, 2000). http://cae.wisc.edu

[CDG97] CHU C.C.P., DANI T.H., GADH R.: *Multimodal interface for a virtual reality based computer aided design system* - 1997 IEEE Conference on Robotics and Automation, 1997. Proceedings.,(1997) - http://ieeexplore.ieee.org

[FdASM02] FIORENTINO M., DE AMICIS R., STORK A., MONNO G.: *Spacedesign: A Mixed Reality Workspace for Aesthetic Industrial Design*. Proc. of ISMAR 2002 IEEE and ACM International Symposium on Mixed and Augmented Reality, Darmstadt, Germany pp. 86-94, (Sept. 30 - Oct. 1, 2002),. http://ieeexplore.ieee.org.

[FMU04] FIORENTINO M., MONNO G., UVA A.E.: *SmartSnap: addressing 3D pointing anisotropy in Virtual Reality CAD application*. International conference on the Computer Graphics and Vision, GRAPHICON '2004, Moscow, Russia (September 6 - 10, 2004) http://www.graphicon.ru/2004/Proceedings.

[FMU05] FIORENTINO M., MONNO G., UVA A. E.: *The SenStylus: A Novel Rumble-Feedback Pen Device for CAD Application in Virtual Reality*. Proceedings of WSCG2005, The 13-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision'2005. (2005) http://wscg.zcu.cz

[GTV05] GUTIERREZ M., THALMANN D., VEXO F.: *Semantic Virtual Environments with Adaptive Multimodal Interfaces* 11th International Multimedia Modelling Conference (MMM'05)

[HHR06] HALEVY G.,HECHT D., REINER M.: *Multimodal Virtual Environments: Response Times, Attention, and Presence* - Presence: Teleoperators & Virtual Environments (October 2006), http://www.mitpressjournals.org

[Mil04] MILOTA A.D.: *Modality fusion for graphic design applications*. Proceedings of the 6th international conference on Multimodal interfaces (2004) ACM Press

[RS05] REITMAYR G., SCHMALSTIEG D.: *OpenTracker: A flexible software design for three-dimensional interaction*. (September. 2005), Virtual Reality Journal Vol.9 Num.1, Springer London.

[SFH*02] Schmalstieg D., Fuhrmann A., Hesina G., Szalavari Z., Encarnacao L.M., Gervautz M., Purgathofer W.: *The Studierstube Augmented Reality Project*. PRESENCE - Teleoperators and Virtual Environments, Vol. 11, No. 1, pp. 32-54, (2002) MIT Press.

[VW97] VO M.T., WAIBEL A.: *Modeling and Interpreting Multimodal Inputs: A Semantic Integration Approach*. Technical Report , School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, (December 1997) CMU-CS-97-192.