Design review and visualization steering using the INQUISITIVE interaction toolkit

L. Sastry, D. R. S. Boyd and M. D. Wilson
Information Technology Department
CLRC Rutherford Appleton Laboratory,
Chilton, Didcot, OX11 0QX. UK
{m.sastry|m.d.wilson|d.r.s.boyd}@rl.ac.uk

Abstract. This paper describes the architecture of an interaction toolkit for creating virtual environments. The toolkit contains interaction techniques, interaction objects such as menus, spanner and a runtime manager to interface to virtual reality development tools such as Maverik. The toolkit's use with a number of real-world applications in science and engineering and with different virtual reality development tools is also described. Future plans to provide an interactive interface are described.

1 Introduction

As novel technologies are absorbed into the conventional systems development methodology they usually pass through three phases: firstly, they are demonstrated in specialised stand alone systems; secondly they are demonstrated in rapid prototype environments used to elicit requirements from users unfamiliar with the potential of the technology in their domain; and thirdly, system representations and documentation for each contractually important development stage, and methods for moving between them are defined incorporating best practice. As knowledge based technologies did in the late 1980's, so Virtual Reality (VR) technologies are now moving from the second to the third of these phases, and establishing manageable system development practices that can be subject to general contractual obligations [1].

The current market for VR development tools ranges from public domain toolkits to high cost prototyping and development environments. Equally, some tools are continuing to address a generic range of VR applications, while others are becoming more focused to one application. For example, being linked as a real time interaction environment to a 3D CAD modelling tool for engineering design. In this market, VR developers have the choice of expensively maintaining the skills to support a range of tools to meet broad customer requirements, cheaply exploiting a single tool very well but thereby limiting their market, or developing their own custom toolkit layer that can be applied to a range of delivery vehicles, thereby meeting the needs of a wide customer base while also limiting the skills required by their developers to a single toolkit. The additional cost of this third option is that it usually requires development overheads calling on systems level skills that do not overlap with the VR designer's. An additional benefit of this approach in other technologies that have joined the

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

systems development mainstream, is that such custom toolkit layers usually define API's that drive the development of market tools and form the basis of standards [2]. Rapid prototyping tools for the 2D WIMP-based (Windows, Icons, Menus and Pointer) applications aided much useful research into usability issues and eventually design guidelines. Similarly, rapid prototyping tools for the creation of interaction rich virtual environments for the experimentation of competing interaction styles or techniques are essential to the understanding of the full potential and hence the exploitation of interactions in virtual environments [3].

The aim of the Interaction Toolkit development within the INQUISITIVE project is to provide support for developing a rich set of interaction techniques for use in VR-based applications. We are approaching this from the points of view of the user's need for an appropriate, consistent and effective set of interaction techniques to carry out application-specific interaction tasks and the application developer's need to deliver this cost-effectively. The design should strive to provide easy configuration of the interaction techniques to use both existing and new input devices [4]. Interaction techniques should be flexibly adaptable by the application developer to meet the needs of a wide range of application scenarios. The toolkit should interface to existing VR run-time systems and input device drivers through defined application programming interfaces (APIs). It should support the portability of interaction techniques and input device configurations across both VR systems and hardware platforms.

The next section describes the basis from which requirements for such an interaction toolkit is derived. The architecture and components of the toolkit is described in Section 3 followed by some demonstration applications in Section 4. Future Work is briefly covered in Section 5 and Conclusion in Section 6.

2 Application and Toolkit Requirements

Our customers for VR applications are scientists and engineers [5] who have two main applications. Firstly, they wish to design and construct buildings and apparatus using 3D CAD engineering tools and undertake group design reviews using interactive real time navigation of them in VR. Secondly, they wish to visualise the data arising from scientific experiments and control that visualisation, maybe changing parameters of the visualisation process, or even steering the experiment generating the data in real time.

The users' objective is to achieve their task goals, and they are only open to using VR when it can be shown to help achieve that objective more effectively, or efficiently than alternative means. One of the advantages to the users of VR is that it allows them to do unreal things that they could not do in the real world, such as measuring between unreachable locations, or moving immovable objects. However, even in these unreal cases, the interaction in immersive or semi-immersive 3D may facilitate a speed of navigation, a precision of interaction, or a perception resulting in insight that are unattainable otherwise [6].

These users are experts in the real world domain tasks, and familiar with the real world objects and actions in their domains, but they are also highly computer literate in their own specialist tools, and accustomed to many computer domain user interface conventions. For the visualisation task, the data and its relationships have no real world representation to imitate in a virtual reality, therefore most aspects of the visualisation scene are either domain conventions or even just conventions of a previous computer representation. Consequently, the users are open to the full range of realism, and VR interaction from full presence to 2D interaction with 3D graphics.

The style of use of different user groups varies considerably; some users wish to distribute VR applications to large communities to be used around the world, so they wish to use public domain code. Others require fully certified and supported development and interaction environments that will conform to the quality control constraints on general contracts for the development of multi-million satellites systems. Therefore no single VR interaction environment will meet these requirements, so we currently use both MAVERIK [7] as a public domain tool and Parametric Technology's dvMockup VR kernels.

Local variations also exist in the requirements that cannot easily be met by both these environments requiring the development of further interaction components. For example, in one application where a public domain interaction tool is required, users wish to use an eye-level viewpoint for gross navigation around buildings and equipment, but wish to place the viewpoint at a fixed location for detailed study of experimental behaviour. Large development environments, but not the public domain interaction tools provide such facilities. A common level of implementation of these is required to meet the user requirements so as not to lock designs into the capabilities of a single development environment.

3 INQUISITIVE Interaction Toolkit

The interaction toolkit will improve support for developing user interaction within task-oriented virtual environment applications. Analysis of the interaction technique has led us to a modular design for the toolkit with defined interfaces to input devices and existing commercial and public domain VR system kernels. The interaction toolkit is being developed to provide application developers and human factors researchers with a portable toolkit of interaction techniques for navigation, selection and manipulation within virtual environments.

3.1 Toolkit Architecture

All application tasks, however complex, can be implemented in terms of a combination of tasks from the four basic classes of user interaction - navigation, selection, manipulation and data input, in virtual environments [8,9]. Each basic interaction task can be realised using a number of possible interaction techniques. For example movement can be implemented using the *magic carpet* or *point-fly* techniques. Each application will identify one or more interaction techniques appropriate for carrying out the tasks required in that application. This in turn will

guide the definition of the interaction techniques needed to realise those techniques. A suitable combinations of these interaction techniques are used to achieve the application tasks. The main functional components that the toolkit must provide to cater for this are:

- a set of interaction techniques for the four classes of basic interaction tasks;
- a set of generic virtual interaction objects such as toolbox;
- a run-time interaction framework.

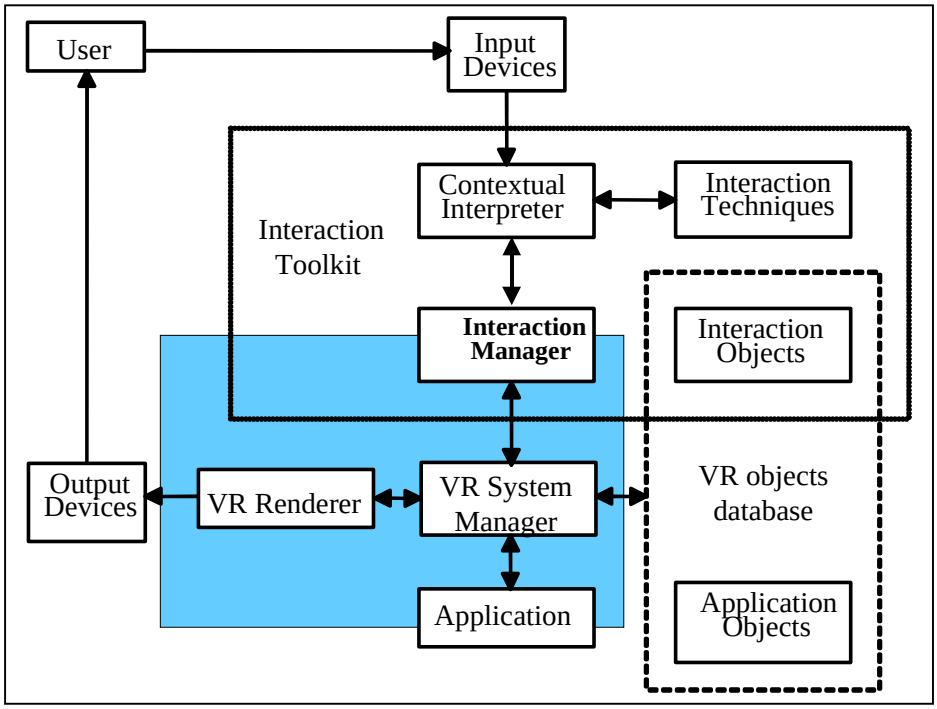

Fig.1. Relationship between interaction toolkit, input devices, VR system and application (the shaded portion represent the components of the VR kernel)

Figure 1 above depicts the detailed architecture of the interaction toolkit and how it maps on to a typical VR kernel. The toolkit provides interaction techniques for the four classes of basic interaction tasks identified above supporting a number of common interaction techniques for each. In navigation, for example, there will be interaction techniques which move the user through the VE and change his viewing direction in response to user-driven inputs from, say, a spacemouse and tracking devices attached to the user. The toolkit must also provide the capability for an application developer to implement new interaction techniques to meet specific application requirements.

Interaction objects are virtual objects with which the user is able to interact in the VE [10]. They contain methods for describing both their functional and presentational properties. The same object in different VR systems will have the same functional description because its behaviour is the same but the description of its presentational properties will be different because each VR system has its own native format for describing the perceptual aspects of virtual objects. A simple example of an interaction object is a virtual spanner, one of a number of objects which might be found in a virtual toolbox used for a maintenance training application.

The run-time interaction framework defines how the input devices and the interaction techniques are dynamically configured and how the outputs from the interaction techniques are mapped into the run-time processes provided by the VR system for implementing behaviour such as collision detection, for updating the VR object database containing the dynamic state of the VE, including the interaction objects and the virtual user, and for rendering the VE.

In its simplest definition, the interaction toolkit is a set of library routines that can be called on to implement an interaction behaviour to an object within a virtual environment. Figure 2 below shows a typical interaction technique mapped on to a specific VR kernel to elaborate how the above architecture works in practice. The example shown considers a 2D/3D mouse and virtual hand based interaction with a generic window-pane which can be used to create simple head up displays, menus, labels etc. and their characteristics and functionality on the fly as required.

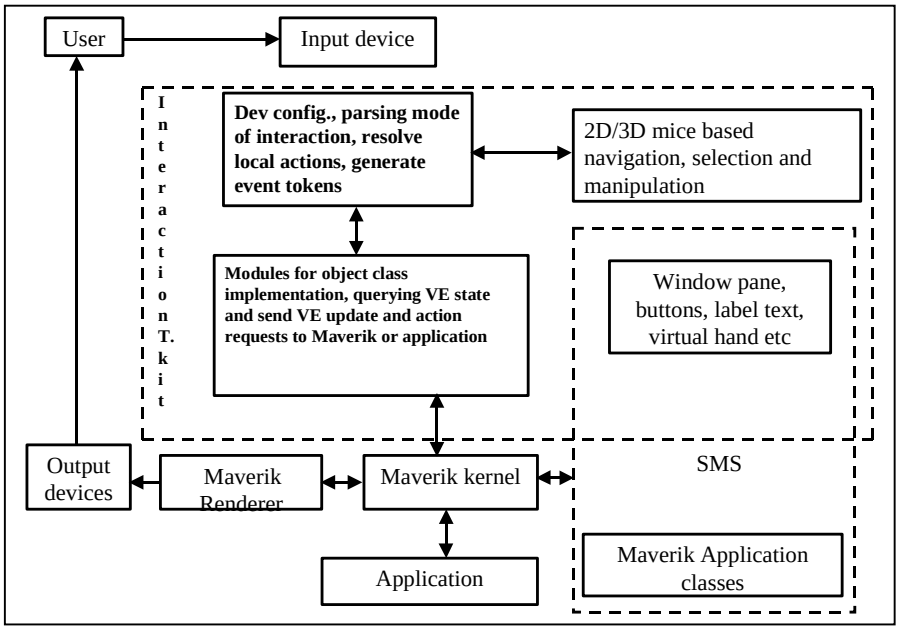

Figure 2. Relationship between window-pane/menu object class, virtual hand, the Maverik VR kernel and application.

Table 1 below shows some classes of interaction objects in the interaction toolkit. All classes of interaction object in the interaction toolkit allow VR application designers to change appearance of widgets.

When instantiated as virtual application objects the interaction objects can be: distance measuring tools, meters to read values of temperature, radiation etc.. from database underlying CAD model etc.. In the later case the display on a meter can be on a windowpane or on hand held display moving with the user depending on which interaction object is chosen. This flexibility in instantiating application objects in different ways shows the power of the interaction toolkit to both meet interaction requirements, and to allow their investigation through rapid prototyping when required.

Interaction Objects		
Window pane	Billboard – turn to user viewpoint	
	Fixed at location and presentation angle	
	Fixed to head up display location	
		Information presentation screens
		Selectable buttons – for menu of application commands, e.g. VR world creation or editing
Attachment	Red Pin calling information presentation screen on selection	
Pointer - single handed	Laser Beam to select or manipulate remote objects	
Pointer - double handed	Laser Beams to select groups of objects in 3D space	
Slider & scale	To set and show values	
Dial	To set and show values	
Constrained object	Objects with behaviour constraints (e.g. hinges).	
Constrainable Object	Objects constrained by the environment (e.g. spanner constrained to move in limited directions where the axis of movement constraint is inherited from an object it is attached to.	

Table 1: Classes on Interaction Objects in the Interaction Toolkit

4 Toolkit Demonstrations

A sample set of interaction techniques will be presented (demonstrated) which include application/user centred navigational and/or object manipulation, real-time interactive editing, querying and steering of the virtual world.

The testbed demonstrator applications include an engineering design review with an architectural walk-through and a visualisation and three-dimensional browsing of Cluster-II satellite data implemented using the interaction toolkit.

4.1 Design Review Demonstration

Engineers need to construct a new building, particle beam target, and experiments on beam lines off the target. Each component is being developed by different teams throughout Europe. The CAD models of the components have been integrated and imported into a VR environment to perform design reviews where groups of engineers jointly view the model in an auditorium with a facilitator navigating and interacting with it to identify and correct design errors.

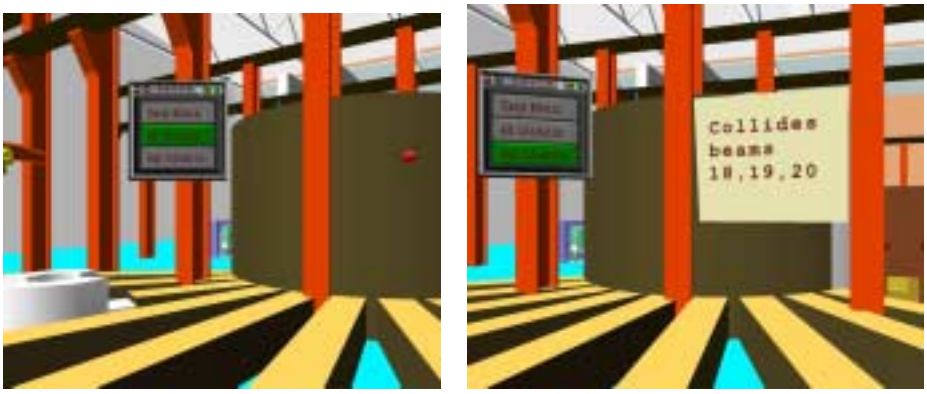

Figure 3 shows a redlining tool for engineering design review. The red pinhead in the left-hand image is indicative of an attached annotation that can be activated as shown in the right-hand image

The group of designers undertake a variety of interactions with the model: simple navigation around the model shows design flaws (e.g. the vertical red pillar in Figure 3 that collides with the beam housing); potential design problems identified by individual designers and marked with red pinheads must be resolved by the group; distances, radiation levels, temperature levels etc. need to be measured in the overall design and checked against requirements. These interactions require the use of several of the interaction objects supported by the toolkit, and the use of the toolkit allows their development more quickly than ad hoc development has in the past. The use of the toolkit allowed the selection of interaction technique appropriate to the facilitator and group of designers at a particular meeting to investigate problems that were not specified in the initial requirements but could be responded to on the fly.

4.2 Data Visualisation Demonstration

Scientists need to study instrument calibration on a satellite, which is always a critical issue on space physics missions. To do this they wish to visualise the instrument data, navigate through it, focus in on subsets of interest, then go back to the whole set and consider it again.

Figure 4 below shows observational data from AMPTE-UKS spacecraft in NASA Common Data Format (CDF) in a hierarchy with associated metadata. The objective of the experiments is to study the distribution of chemicals in the atmosphere based on the electron density distribution in Earth's Ionosphere. The basic measurement made by the electron instrument on *AMPTE-UKS* was to count the numbers of incoming electrons simultaneously in each of eight directional sensors. The AMPTE electron measurements can be considered as a sequence of measurements of the three-dimensional velocity space distributions of the electrons - with one distribution being measured every T seconds, where T is a suitable integration time. T must be greater than or equal to the spin period of the spacecraft (approx. 5 seconds).

Interaction to select different data files, to zoom in and out of the time period being observed, and the density of the data, and to inspect the metadata – which instrument used, when by whom, which sensor, data type, references to related data etc.. so that scientists can evaluate the reliability of the data, and its relation to other data.

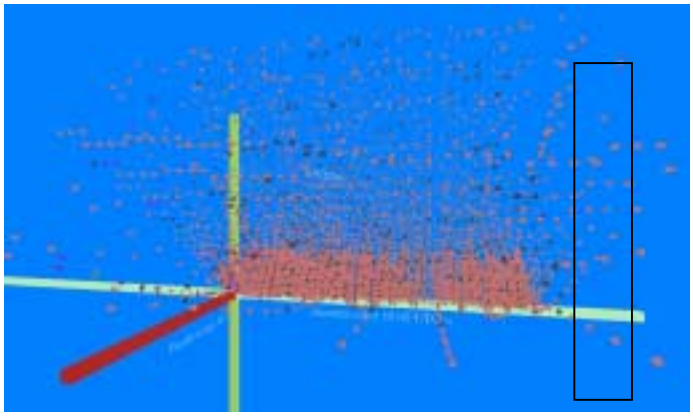

Figure 4: Visualisation of an electron density distribution over time (x-axis) during spacecraft flight (y & z axes) showing an event requiring detailed investigation (black rectangle).

Such discontinuous data was previously studied in 2D graphs of energy plotted against universal time where the information from the 8 different sensors are either amalgamated or presented individually. Scientists need data from all 8 sensors in single view to see how sensors are measuring data in space. Using 3D plotting in IDL gives supports inspection of data from the 8 sensors, but does not support real time navigation unless the data is so diluted as to be hard to interpret, and data context is

lost from the sort buckets after focussing in. If the whole data set is inspected, then the data quantities are overwhelming.

Consequently a semi-immersive display (Crystal Eyes 3D glasses with head position sensor and SpaceMouse) driven by a VR runtime system supports the interactive real time selection and investigation of whole and part data sets. Current interaction objects for navigating through and selecting data, and requesting metadata do not meet their needs, but the INQUISITIVE interaction toolbox does, and supports tuning the interaction objects to individual user needs.

5 Evaluation and Future Work

Besides inhibiting the adoption of VR-based interaction techniques, the lack of design environments that do not constrain designs to their limited capabilities also inhibits human factors research into issues such as the utility and usability of 3D techniques for achieving user interaction goals. We will need to understand these issues in order to be able to develop design guidelines for 3D interaction comparable to those which exist for 2D desktop applications, and the INQUISITIVE Interaction Toolkit will provide a starting point in developing these.

The toolkit's functionality will be evaluated for the perceptual interfaces and interaction techniques of the testbed applications and usability guidelines developed.

This toolkit, it is hoped, together with guidelines for its use based on the demonstrations, will help user interface designers to produce more useable and productive applications thereby accelerating the exploitation of VR technology for real-world engineering product design reviews and scientific visualization applications within the Laboratory.

The toolkit is to be extended to provide a graphical user interface for low end toolkits emulating facilities provided by current high end environments for the creation from virtual worlds from within themselves.

6 Conclusion

The INQUISITIVE method and toolkit meet some of the needs of the VR developer at the stage we are in the evolution of the technology where there have been attractive demonstrations, some industrial applications, and we are moving towards the incorporation of VR technologies into conventional system development. However, both the method and the toolkit are early attempts, with the need for further refinement, and the development of design guidelines to clarify the mapping from user requirements to the design tradeoffs of VR application development.

Parts or all of the INQUISITIVE Toolkit will be made freely available to facilitate take up of the INQUISITIVE method, and promote standardisation of technology and methods in the VR field. Please contact the first author for license information.

Acknowledgements

The work reported in this paper was partly funded by the UK EPSRC through grant GR/L52406 to the INQUISITIVE (INcreasing the Quality of User Interaction for Strategic Interactive Tasks in Virtual Environments) research project.

References

1. Wilson, M.D., Duce, D.A., Simpson, D. Life cycles in Software and Knowledge Engineering: A comparative review. Knowledge Engineering Review vol.3(4) pp.189-204 (1989).

2. Duce, D., Kansy, K., Wilson, M.D. Report and Recommendations from the VRML Workshop 29/30 January 1997, Abingdon UK. ERCIM Research Report, 02/97-R049, ERCIM, France.

3. Jacob, R.J.K. A visual language for non-WIMP user interfaces, In Proc. IEEE Symposium on Visual Languages, pp.231-238 (1996). IEEE Computer Society Press.

4. Hinckley, K., Pausch, R., Goble, J.C., Kassell, N.F. A survey of design issues in spatial input. In Proc. ACM UIST'94 Symposium on User Interface Software and Technology, Marina del Rey, California, 213-222, Addison-Wesley/ACM Press.

5. Sastry, L., Boyd, D. Virtual Environments for Engineering Applications Virtual Reality vol.3 (4) pp.235-244 (1999).

6. Bowman, D., Hodges, L.F., & Bolter, J. The virtual venue: user computer interaction in information rich virtual environments. Presence,Teleoperators and Virtual Environments vol. 7(5), pp.478-493 (1998).

7. Maverik User Guide http://aig.cs.man.ac.uk/systems/Maverik

8. Hand, C. A survey of 3D interaction techniques. Computer Graphics Forum, vol. 16(5), pp.269-281 (1997).

9. Boyd, D. and Sastry, L. Development of the INQUISITIVE Interaction Toolkit - Concept and Realisation. In Workshop on User Centered Design and Implementation of Virtual Environments, (Eds.) Smith, S. and Harrison, M. pp.1-6, 30th September, 1999, University of York, York.

10. van Dam, A. Post-WIMP user interfaces, Communications of ACM, vol. 40(2), pp.63-67(1997).