

Cooperative Virtual Environments: lessons from 2D multi user interfaces

Gareth Smith
Computing Department
Lancaster University
Lancaster, UK
gbs@comp.lancs.ac.uk

ABSTRACT

Existing Cooperative Virtual Environments present the same shared world to each of the cooperating users. This is analogous to the use of strict-WYSIWIS in early 2D interfaces. Research in the area of shared 2D interfaces has shown a strong trend to support individual tailoring of the shared views, and move away from the strict-WYSIWIS abstraction. This paper argues that the development of Cooperative Virtual Environments can gain from the experience of research into in shared 2D interface systems, and presents a model to manage the use of subjective views in Cooperative Virtual Environments.

Keywords

WYSIWIS, Shared interfaces, view coupling, VR

INTRODUCTION

Early 2D multi user interface systems supported shared interfaces by presenting exactly the same image of the application to all users. This simple replication of the system's image secured a founding abstraction for multi user interfaces: What You See is What I See (WYSIWIS). However, the early applications based on this abstraction, such as Boardnoter [21] and Cognoter [9], suffered from a number of problems. Experiences with these systems in the CoLab environment highlighted problems with the WYSIWIS approach, Stefik concluded that "*WYSIWIS (What You See is What I See) is too inflexible, if strictly interpreted, and must be relaxed to better accommodate important interactions in meetings*".

Stefik suggested that the shared interfaces which conformed to the WYSIWIS abstraction should be relaxed along a number of dimensions. The notion of relaxed WYSIWIS provides each individual user with the ability to configure their shared user interface to best suit their working needs. User interface tailoring is now generally accepted in the areas of single and multiple 2D user interface systems [10].

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee.

Computer Supported Cooperative Work '96, Cambridge MA USA
© 1996 ACM 0-89791-765-0/96/11 ...\$3.50

Cooperative Virtual Environments (CVE) such as Dive [5] and MASSIVE [11] are an emerging technology which provide a shared 3D space that may be populated by users and any type of three dimensional artefact. CVEs may be considered as shared *three dimensional* interfaces. CVEs evolved directly from single user virtual reality systems. That is, they are single user systems which have been extended to include a number of users, and lack a number of cooperative features.

BACKGROUND AND MOTIVATION

Currently, CVEs present the same shared data to each of the cooperating users. This is analogous to the use of strict-WYSIWIS in early 2D interfaces. This paper argues that the flow of development from single-user virtual environments to cooperative virtual environments can gain from the experience from a similar trend within 2D based user interface systems. This section briefly reviews these systems, prior to presenting a model for supporting diverse views of shared virtual environments.

The development of applications that support a number of 2D interfaces across a community of users has played a prominent role in Computer Supported Cooperative Work (CSCW). Different researchers have investigated techniques and architectures to support the real time management of these interfaces. Research has focused on the development of facilities to co-ordinate and manage interaction across the different interfaces. In general, one of two alternative strategies have been adopted for the construction of multi-user interfaces, and cooperative applications have been characterised as either *collaboration transparent* or *collaboration aware* [13].

Collaboration transparent interfaces focus on sharing the user interface of an application among a group of users without modification. This is achieved at the cost of limiting the amount of control users have in managing the cooperative properties of the interface. This approach also relies on an assumed model of cooperation based on many readers but only one writer. The responsibility for managing the interaction involved lies solely with an external agent that uses a floor control policy to co-ordinate interaction. This control policy normally applies to the

application interface as a whole with each user having permission to interact with the application in turn by being given the floor. Two dimensional systems that exploit collaboration transparency include Rapport [1], SharedX [12], and MMConf [6].

In contrast, collaboration aware applications provide extensive facilities to allow users to both *configure and control interfaces*. The cost of this increased management is the need for the cooperative applications to be directly responsible for the facilities provided. This incurs a significant development cost and facilities developed tend to be specific to an application and little consideration is given to providing management and tailoring facilities across a range of applications.

Collaboration transparent systems rely on the distributed nature of an application. The application is ignorant of multiple users and the windowing systems merely replicate the shared interfaces. This is directly comparable to the current CVE situation, where the shared virtual world is merely distributed to each of the users. Not surprisingly, similar shortcomings prevail.

This paper presents an approach to multi-user interface construction for 3D environments that supports dynamic prototyping of shared world representations for individual users. The adopted approach is to consider the different world views presented to members of the user community as being projected from a common shared world definition. This allows the management of cooperative features to be achieved by providing facilities that control the derivation of the projected world. This builds upon previous approaches based on 2D interfaces [18]. As these cooperative facilities are external to detailed behaviour within the world, the adoption of general facilities across a range of environments is encouraged.

APPROACH

Current CVEs allow each user to perceive the shared world model from different orientations. However, each user currently sees exactly the same shared world model (following the strict WYSIWIS abstraction). There are a few exceptions to this, which are implemented within the user's renderer, such as viewing the whole world as either: wireframe; filled polygons or textured polygons.

The aim of this work is to provide a set of facilities that allow the realisation of effective cooperative worlds. Central to the development of these facilities is the provision of appropriate mechanisms for the management and control of different users' world views. Users and developers should be provided with the flexibility to quickly amend their views of the world to effectively support the cooperative nature of the tasks taking place.

Many 2D based multi-user interface systems exist, including Suite [7], SOL [18], Mead [3], Rendezvous [15] and GroupKit [16]. All of which support tailorable user interfaces across a community of users who may have

different preferences. The model offered by these systems allows the shared data to be viewed in different ways by a number of users. However, there is a range of degrees to which individual interfaces may be tailored. For example, an aircraft in the Mead system may be represented by a flight strip or a radar blip, allowing each user to select the representation which best suits their task at hand, see figure 1.

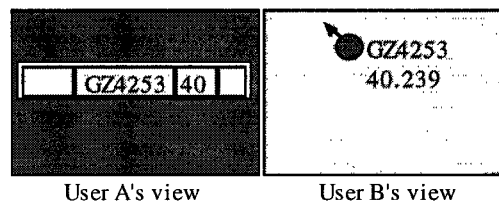


Figure 1 : Two radically different views of the same data

To achieve the high degree of tailorability, in systems such as Rendezvous and Mead, a great deal of overhead is incurred to specify each of the views. Other systems, such as Suite and SOL, provide less radical control over user interface tailoring but allow a user to tailor their interface more subtly and simply. It is this type of system which is built upon in this paper, which argues that only subtle changes in the shared space are needed and this class of 2D system provides a simple and dynamic model of interface tailoring. The following section discusses the problems that occur when using shared interface systems which employ a relaxed WYSIWIS approach, and the limit to which collaborative aware systems are useful.

Problems with Relaxing Spatial Frames of Reference

As discussed previously, many 2D multiuser interface toolkits allow each user to tailor the layout of their interface to suit their working requirements and preferences. This arrangement can cause a number of problems when users collaborate, specifically when the spatial nature of the user interfaces is important. For example, the use of telepointers in relaxed WYSIWIS based environments requires a new telepointing model, as directly copying the location of a user's cursor across a range of user interfaces may lead to telepointers pointing at a number of different (and irrelevant) objects. For example, consider the case depicted in figure 2, The second user (on the right) has altered the state of their interface by moving the locations of the 'open' and 'close' buttons. The first user may be unaware of this and assume that the other user can clearly see that they are pointing at the open button. If the first user suggested to the second user "press this button" (through some audio device), this would result in an incorrect action as the 'close' button would be selected and not the 'open' button which the first user is pointing at.

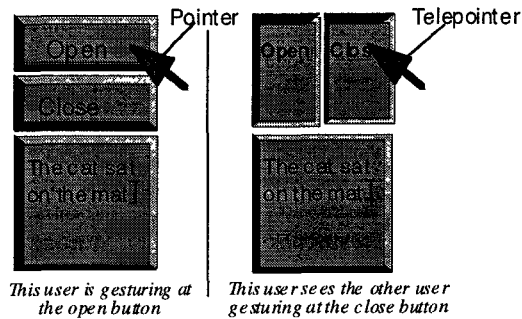


Figure 2 : The problems occurred when using spatial based interaction techniques in a spatially relaxed interface.

These spatial aspects which can be considered a side issue in 2D interfaces (unless systems such as telepointing are used) are the essence of 3D user interfaces, as spatial information is paramount [2]. Populated virtual environments are three dimensional spaces filled with representations of users, in this case each representation of a user may be deemed as a 3D telepointer [20].

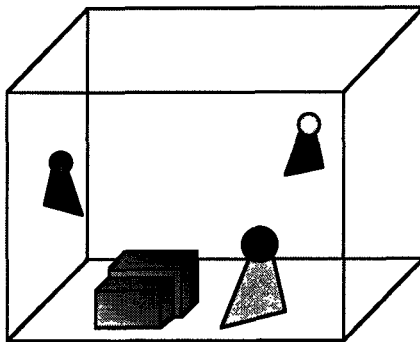


Figure 3 : The location of three users within a shared world

Relaxing the spatial arrangements in 3D user interfaces introduces a range of problems and inconsistencies within the virtual space. A solution offered by Snowdon to overcome this and control radically differing 3D spaces employs transformations. Such transformations are potentially complex to define, manage and validate. Arguably, wildly differing 3D cooperative spaces with possibly no spatial consistency have specialised use and can only be effectively used if the awareness of other users and their activities is effectively transported. It is likely that such a goal will be difficult to achieve.

Therefore this paper proposes that the vast majority of spatial information within shared virtual spaces should be the same for all users, and that subtle changes to each user's view of the world is sufficient for many uses of cooperative VR applications. The undertaken approach is to manage only the representations of objects within the virtual worlds, and not their locations.

A 2D Approach to Control Interface Sharing

SOL [18] is a 2D user interface system which supports rapid construction of cooperative user interfaces and supports dynamic end-user tailoring of individual displays. A number of key features which are supported by SOL, are also relevant to 3D user interface systems. Hence, these features provide a driving set of needs for a system to tailor individual 3D interfaces. Important goals for any user interface controlling system are:

- Maintain a clear separation between behavioural characteristics of the application and cooperative aspects of the user interfaces;
- Allow end users to readily tailor their user interface;
- Provide a set of simple and consistent management facilities which can be applied across a range of multi-user applications.

SOL manages a wide range of user interface issues including individual user interface tailoring, interactional behaviour control, and fine grain control over user interface sharing. To achieve this SOL utilises an access model and an interactional policy mechanism. The scope of this paper is the definition of individual subjective views of a shared 3D definition (a CVE); and is not concerned with management of the behavioural aspects of such environments. Hence, this paper concentrates on the model used within SOL to tailor individual interfaces

The essence of the SOL approach to interface control is an access model based around a canonical shared interface. An initial definition of the virtual world is interpreted by the access model which has a set of user configurations. The access model then uses these configurations to define a representation of the interface for each user, see figure 4.

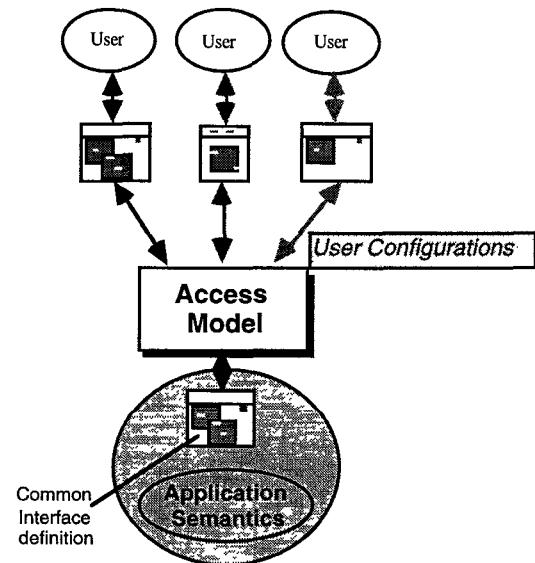


Figure 4 : How SOL manages individual representations of a shared interface

This paper describes a model to manage subjective views of CVEs which embodies the essence the approach taken by SOL.

Using an Access Based Approach

The approach taken by SOL allows a common shared definition to be tailored for a number of users. Use of an access model grants (or denies) users permissions to see particular objects within the shared world. This approach is taken by a new model, derived from SOL, which is tailored specifically for use within CVEs. SOLVEN (SOL - Virtual Environment extension) augments the core SOL model with additional features, which may be required within CVEs. These features are rationalised and described later in this paper.

As with SOL, the essence of the SOLVEN approach is an access model based around a canonical *shared world*. An access model interprets user's configurations, and defines a view of the shared world for them (see figure 5).

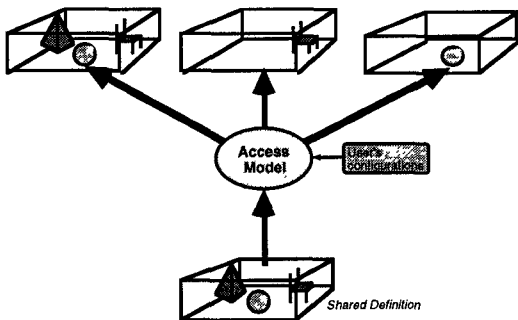


Figure 5 : The essence of the SOLVEN model

SOLVEN exploits the object oriented approach taken by many VR frameworks, such as dVS [8] and Dive [5]. In these systems geometric primitives including lines, polygons, cubes and spheres may be arranged to construct a single high level object. Control over an object's representation should be possible if that object is a simple primitive (such as a cube) or a more complex grouping (such as a car). As many objects are likely to exist within a VE, and not all of these need to be represented subjectively, a mechanism to support subjective views needs to be able to dynamically manage which objects should have subjective facilities and which should not.

Further requirements for such a mechanism may be ascertained by analysing a number of hypothetical situations in CVEs, where subjective views may be of use. The following sections describe such scenarios.

CASE SCENARIOS

This section describes a range of uses of Cooperative Virtual Environments, and how the use of subjective views may be used to improve a users representation of the world to suite their working needs.

Shared Virtual Cities

Consider a CVE representing a town, which is populated by 'virtual tourists'. A cooperative tool may allow many groups of users to collectively tour a proposed holiday destination. This allows the would-be tourists to familiarise themselves with their intended holiday resort. A tourist operator may also be present in the virtual town to guide the customers around.

Subjective views may be utilised to augment the virtual town with additional entities, such as information posters which provide the *operator* with useful knowledge concerning the current location. This information may then be conveyed to the rest of the party by the operator. In addition to being able to see the information posters, the operator may be provided with an immersive tool to help them navigate within the CVE. This tool may allow the operator to answer questions posed by the virtual tourists, such as "where are the cafes?". The query may be answered by highlighting all the cafes in the town, differentiating them from the other shops. The cafes will only be highlighted in the views of the operator and that group of tourists (the CVEs may be populated by any number of parties, each with a operator or guide). In addition to highlighting elements, the operators view may be augmented with a path. The path may be represented by a line arrows, which point to the next location of interest in the CVE.

Abstract Data Visualisation Tool

Q-PIT is an existing tool, which supports a number of users within a Populated Information Terrain [14]. Q-PIT allows users to browse, query and modify the attributes of a set of objects within a 3D virtual space. Three attributes of the objects are mapped onto each of the three dimensions and a further attribute defines the object's shape. Users may select objects to view their attributes. A selected object is denoted by a wireframe sphere surrounding it.

Normally each user's representation of the world includes all users' current selections. However, this is not always desirable as it becomes increasingly difficult for each user to differentiate their selections from all the others. The use of subjective views has been implemented within Q-Pit to restrict the presentation of user's selection of objects, in such a way that a user sees only their selections, see figure 6.

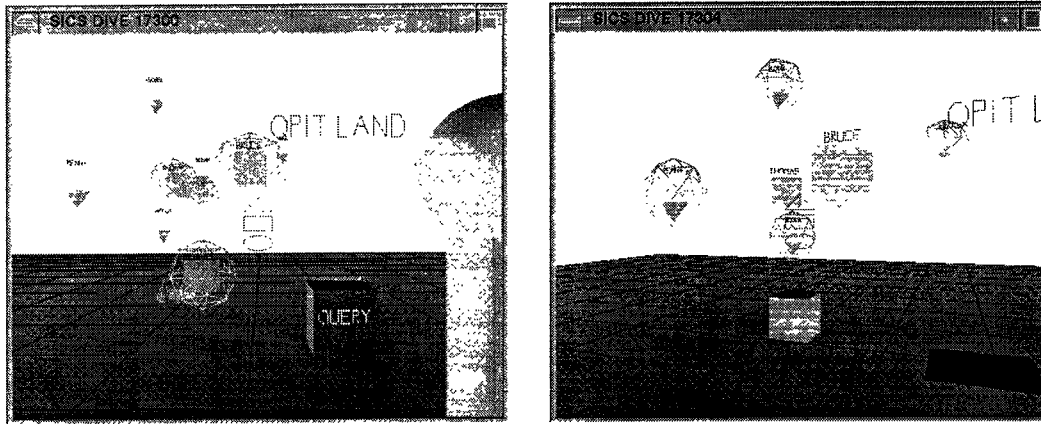


Figure 6 : Two users view of a common virtual world supporting subjective views

The notion of highlighting selected objects through encasing them was derived from the non-subjective CVE. This approach allowed a number of users' selections to co-exist, by adding concentric spheres. With the ability of subjective views, other highlighting mechanisms may be explored, such as changing the colour or increasing the radiance (brightness) of the selected items (or conversely decreasing the radiance of the non-selected items).

Multi-Lingual Shared Spaces

Use of shared virtual spaces is increasing, and many of the participants are at remote sites in different countries. Currently, text within shared worlds is in a single language, and multilingual support can exist only if the translation (in the new language) is *added* to the virtual world. That is, the information is replicated as is found in the instruction manuals of many international products, such as televisions and cameras.

A more desirable option would be to place a single sign in a virtual world, and each user is presented with *only one* text representation in their chosen language. For example, consider a shared corridor with a number of gateways into other worlds, such as meeting rooms, virtual museums and simulation demonstrators, where each is identified by a textual label. Rather than each gateway having a list of labels (one for each language), the users would only see a single label in their preferred language.

Another use of subjective views of text is to change the semantics of the text (with or without changing the language). For example, in the Q-Pit example, each element is identified by a name, which is usually the name of the person it represents. Users within this virtual space may prefer to identify the object through a different attribute, such as their home-town or age. In this case, a number of users would share the same spatial information structure, but it would be labelled differently.

The use of multilingual spaces can also appear in non-shared virtual environments, such as those based on the World Wide Web. Here, VRML is used to describe a 3D space. This definition is downloaded onto individual user's client machines. Even though these 3D spaces are

not populated by other users, the case still arises where a user may wish to view the virtual world with text appearing in their choice of language.

The Model

The requirements for multi-perspective virtual environments involve the manipulation of objects in terms of their presentation to users. From the case scenarios above, objects within virtual environments should be invisible to a subset of users, while also being visible to others. The nature of the presentation should also change, in that an object should be able to be emphasised or de-emphasised, possibly by changing its luminance or colour. This is further extended in the case of textual items within virtual worlds, as the contents of the text field may also change. Hence the actual representations of these objects may be different for any number of users.

Two independent factors may be extracted from the above description of the needs:

- Differing geometric definition (such as different textual definitions).
- Highlighting and de-emphasising abilities (such as making an object appear brighter)

These two factors, termed 'appearance' and 'modifier', may be arranged orthogonally into a 'view-matrix' which defines an object's range of possible representations, see figure 7.

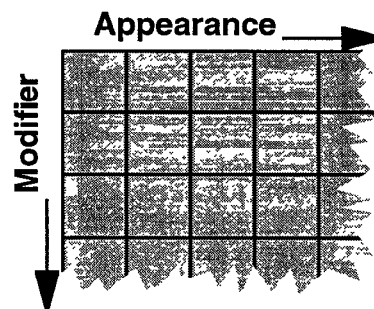


Figure 7 : A generic view-matrix of an object

The appearance of an object is used to describe its 3D geometry. This may be anything from a simple cube to a more complex building. The modifiers are independent effects that may be applied to any of the appearances. The default set of modifiers are taken from the sample scenarios above, namely:

- *Off*. The object should not be visible to this user
- *Dim*. The object should be visible but less obvious, such as made darker or more transparent.
- *Normal*. Display this object normally, i.e., no modifications.
- *Bright*. The object should be presented to the user and emphasised within their visualisation by means such as increasing its luminosity.

For example, consider a simple house, which has only one appearance. It's view-matrix may be represented by:

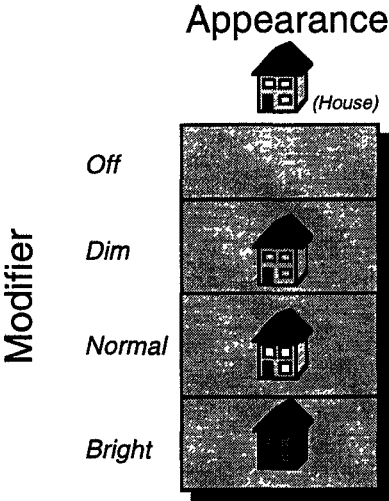


Figure 8 : The house object's view-matrix¹

As modifiers are *functions* which act upon the appearance data, the object need only supply this information (its 3D geometry), which it must do in any system. If two or more appearances are required then the object must supply the geometric data for each.

Not all objects within the virtual world will have an associated view-matrix, but for each object that does there are at least four possible versions of that object. Rather than ensuring that each user has a definition for all these objects, a view-matrix supports the specification of a default entry. The default entry defines the view (appearance and modifiers) of an object if no specific definition can be found for a user. The default view may be any one appearance-modifier combination but if none is specified, then the first appearance with the normal modifier is used.

In order to record the correct representation of an object for a user, SOLVEN utilises the same core access model as

¹ Due to the difficulty of effectively representing luminance in printed text, stronger/darker colours are used to simulate brightness.

defined by SOL. Central to this model is a 'permission matrix'. A permission matrix identifies a user's range of preferences across two orthogonal factors. A permission matrix is used within SOLVEN to record the correct appearance and modifier for each user.

The required view (appearance and modifier combination) for a user is specified by marking an entry within that user's permission matrix. Unlike its use in SOL, the access model in SOLVEN restricts the permission matrix so that only one selection may be marked by any one user. For example, consider the situation where two users exist in the same world containing the house, as depicted in figure 8. One user wishes to see the house normally, while the other wishes it to be highlighted. The permission matrices of each user in this case are:

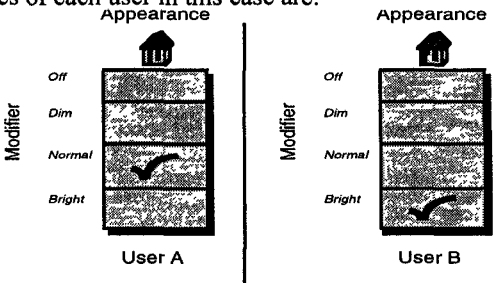


Figure 9 : Two permission matrices for two users

Supporting Text

Text based objects are manipulated using the same techniques discussed earlier. Each different textual representation defines an appearance column in that object's view-matrix. This supports the use of different textual representations while also supporting the range of modifying effects. For example, consider two users in the same CVE, which contains a sign to a bathroom. The sign is capable of presenting itself in a number of different languages or dialects, including English, American and French. In this case, its fleshed-out view-matrix can be represented by:

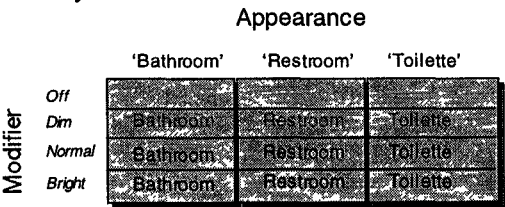
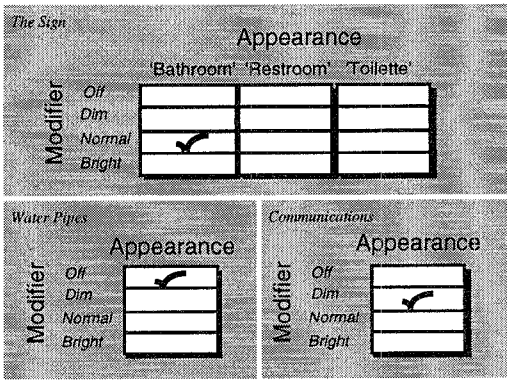


Figure 10 : The view-matrix of a text object

If one user was British and the other French, then two possible permission-matrices are:



And hence, their view is as represented by figure 14.



Figure 14 : The Electrician's view of the room.

FUTURE WORK

A prototype demonstrator of the SOLVEN model currently exists in the DIVE system. A second generation implementation is currently under development, which will support simpler specification of the permission matrices through either a desktop window or immersive panel. It is intended to integrate the desktop version of the permission specification interface into 'AC3D', a world building tool based on VR-MOG [4]. This allows users to construct virtual worlds and prototype both the permission matrices and the view matrices of particular objects.

A further possible use of appearances in an object's view-matrix is to specify the same object's geometry but in differing levels of detail. The notion of levels of detail allows a complex object to be rendered with less detail in certain circumstances, such as when there is insufficient computing power to render the object quickly enough. Some single user systems provide a means of specifying levels of detail manually, and the renderer chooses the correct model depending on computing resources. Other approaches are based on dynamic systems [17] which degrade a given 3D object at run-time. The use of such dynamic techniques remove the need to define a number of 3D structures for each object. Whichever approach is taken

the use of extensible appearances in the SOLVEN model allows each user (or automatically by their rendering system) to specify the correct level of detail; for all objects within the world or for particular objects which are of little or particular importance.

CONCLUSIONS AND SUMMARY

This paper presents motivation for simple but effective use of subjective views within cooperative virtual environments. It has also highlighted the problems of relaxing the sharing constraints within these environments. A means of supporting subjective views within virtual environments is described, and a model to achieve this is presented.

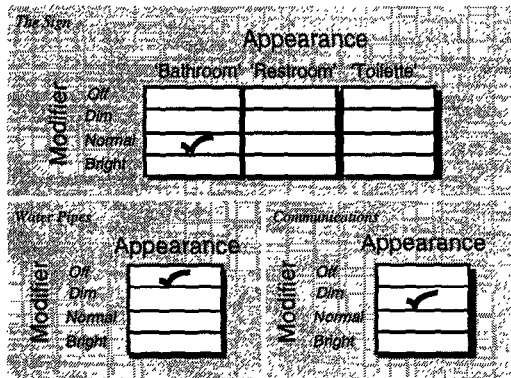
The use of subjective views within virtual environments can be of great benefit to its users, but if overused then the lack of awareness between the users may introduce additional problems. For example, one user may be gesturing towards an object that does not appear in another user's view of the same shared world. In this case the gesturing user may not be aware that the other users cannot see the particular object, and conversely the other users may not be able to comprehend the gestures made. Hence, the use of subjective views must be finely tuned to gain the benefits of such functionality while maintaining necessary awareness between cooperating parties.

ACKNOWLEDGEMENTS

The research was partly funded by the ACTS programme (COVEN AC040). Thanks are due to many colleagues at Lancaster primarily Tom Rodden and Andy Colebourne.

REFERENCES

1. Ahuja, S. R., Ensor, J. R., & Lucco, S. E. (1990). A comparison of applications sharing mechanisms in real-time desktop conferencing systems. *Proceedings of the Conference on Office Information Systems*. Boston 238-248.
2. Benford S., Fahlen L. Viewpoints, Actionpoints and Spatial Frames. *Proc. HCI 1994*. 409-423.
3. Bentley, R., Rodden T., Sawyer P., Sommerville I. (1992). An Architecture for Tailoring Cooperative Multi-User Displays. *Proceedings CSCW '92*. ACM Press. 187-194.
4. Colebourne A., Rodden T., Palfreyman K., VR-MOG: A toolkit for building shared virtual worlds. *Proc. Conf. FIVE working group*, London, 1995. Ed. Mel Slater. 109-122
5. Carlsson C, Hagsand. (1993) "DIVE - Multi-User Virtual Reality System", *VRAIS '93, IEEE Virtual Reality Annual International Symposium*. 394-400.
6. Crowley, T., Milazzo, P., Baker, E., Forsdick H., Tomlinson R. (1990). MMConf: An infrastructure for building shared multimedia applications, *Proceedings*



And hence, their view is as represented by figure 14.

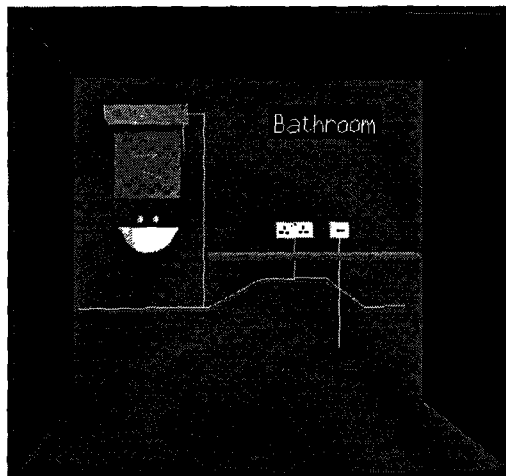


Figure 14 : The Electrician's view of the room.

FUTURE WORK

A prototype demonstrator of the SOLVEN model currently exists in the DIVE system. A second generation implementation is currently under development, which will support simpler specification of the permission matrices through either a desktop window or immersive panel. It is intended to integrate the desktop version of the permission specification interface into 'AC3D', a world building tool based on VR-MOG [4]. This allows users to construct virtual worlds and prototype both the permission matrices and the view matrices of particular objects.

A further possible use of appearances in an object's view-matrix is to specify the same object's geometry but in differing *levels of detail*. The notion of levels of detail allows a complex object to be rendered with less detail in certain circumstances, such as when there is insufficient computing power to render the object quickly enough. Some single user systems provide a means of specifying levels of detail manually, and the renderer chooses the correct model depending on computing resources. Other approaches are based on dynamic systems [17] which degrade a given 3D object at run-time. The use of such dynamic techniques remove the need to define a number of 3D structures for each object. Whichever approach is taken

the use of extensible appearances in the SOLVEN model allows each user (or automatically by their rendering system) to specify the correct level of detail; for all objects within the world or for particular objects which are of little or particular importance.

CONCLUSIONS AND SUMMARY

This paper presents motivation for simple but effective use of subjective views within cooperative virtual environments. It has also highlighted the problems of relaxing the sharing constraints within these environments. A means of supporting subjective views within virtual environments is described, and a model to achieve this is presented.

The use of subjective views within virtual environments can be of great benefit to its users, but if overused then the lack of awareness between the users may introduce additional problems. For example, one user may be gesturing towards an object that does not appear in another user's view of the same shared world. In this case the gesturing user may not be aware that the other users cannot see the particular object, and conversely the other users may not be able to comprehend the gestures made. Hence, the use of subjective views must be finely tuned to gain the benefits of such functionality while maintaining necessary awareness between cooperating parties.

ACKNOWLEDGEMENTS

The research was partly funded by the ACTS programme (COVEN AC040). Thanks are due to many colleagues at Lancaster primarily Tom Rodden and Andy Colebourne.

REFERENCES

1. Ahuja, S. R., Ensor, J. R., & Lucco, S. E. (1990). A comparison of applications sharing mechanisms in real-time desktop conferencing systems. *Proceedings of the Conference on Office Information Systems*. Boston 238-248.
2. Benford S., Fahlen L. Viewpoints, Actionpoints and Spatial Frames. *Proc. HCI 1994*. 409-423.
3. Bentley, R., Rodden T., Sawyer P., Sommerville I. (1992). An Architecture for Tailoring Cooperative Multi-User Displays. *Proceedings CSCW '92*. ACM Press. 187-194.
4. Colebourne A., Rodden T., Palfreyman K., VR-MOG: A toolkit for building shared virtual worlds. *Proc. Conf. FIVE working group*, London, 1995. Ed. Mel Slater. 109-122
5. Carlsson C, Hagsand. (1993) "DIVE - Multi-User Virtual Reality System", *VRAIS '93, IEEE Virtual Reality Annual International Symposium*. 394-400.
6. Crowley, T., Milazzo, P., Baker, E., Forsdick H., Tomlinson R. (1990). MMConf: An infrastructure for building shared multimedia applications, *Proceedings*

- of CSCW '90, October 7-10, Los Angeles, Ca. ACM Press. 329-342.
7. Dewan P. (1990). A tour of the Suite User Interface Software. *Proc. of the 3rd ACM SIGGRAPH Symposium on User Interface Software and Technology*. October 1990. 57-65.
 8. Division Ltd. (1993) "dVS Technical Overview, Version 2.0.4"
 9. Foster G., Stefik M. Cognoter, theory and practice of a colab-orative tool. *Proc CSCW 96*, Austin, Texas, 1986
 10. Greenberg, S. (1991). Personalisable Groupware: Accommodating Individual Roles and Group Differences. *Proceedings of ECSCW '91*, Bannon, L., Robinson, M., Schmidt, K. (Eds.), Sept. 1991. 25-27, Kluwer. 17-31.
 11. Greenhalgh C., Benford S., MASSIVE: a Collaborative Virtual Environment for Teleconferencing, *ACM transactions on CHI*, March 1995
 12. Gust, P. (1988). Shared X: X in a distributed group work environment. *2nd Annual X conference*. MIT, Boston, January 1988.
 13. Lauwers, J. C., Lantz, K. A. (1990). Collaboration awareness in support of collaboration transparency: Requirements for the next generation of shared window systems. *Proc. CHI 1990*. 303-310.
 14. Mariani J, Rodden T, Colebourne A, Palfreyman K, Smith G. Q-PIT: A populated information terrain. *IS&T/SPIE Symposium on Electronic Imaging: Science & Technology*.
 15. Patterson, J. F., Hill, R. D., Rohall, S. L., Meeks, W. S. (1990). Rendezvous: An architecture for synchronous multi-user applications, *Proceedings of CSCW '90*, October 7-10, Los Angeles, Ca., ACM, 1990. 317-328.
 16. Roseman, M., Greenberg, S. GroupKit. A groupware toolkit for building Real-Time Conferencing Applications. *Proc of ACM CSCW '92*. ACM Press. 43-50
 17. Schraft R., Flaig T. A Fuzzy Controlled Rendering System for Virtual Reality Systems Optimised by Genetic Algorithms. *Proc. Conf. FIVE working group*, London, 1995. Ed. Mel Slater. 179-187
 18. Smith, G., Rodden T. SOL: A shared object toolkit for cooperative interfaces. *Int. J. Human-Computer Studies (1995) 42*. 207-234.
 19. Smith G., *A Shared Object Layer to Support Cooperative User Interfaces*. Ph.D. Thesis, Computing Department, Lancaster University, UK. 1995
 20. Snowdon D., Greenhalgh C., Benford S. What You See Is Not WHat I See: Subjectivity in Virtual Environments. *Proc. Conf. Framework for Immersive Virtual Environment 1995*. 53-69.
 21. Stefik, M., Foster, G., Bobrow, D. G., Kahn, K., Lanning, S., Suchman, L. (1987). Beyond the chalkboard: computer support for collaboration and problem solving in Meetings, *Communications of the ACM*, 30(1), January 1987. 32-47.