KUNGL TEKNISKA HÖGSKOLAN

STOCKHOLMS UNIVERSITET

CID-155 • ISSN 1403-0721 • Department of Numerical Analysis and Computer Science • KTH

How not to be objective; Collaborative Virtual Environments

Kai-Mikael Jää-Aro, Dave Snowdon

CID, CENTRE FOR USER ORIENTED IT DESIGN

Kai-Mikael Jää-Aro, Dave Snowdon

How not to be objective; Collaborative Virtual Environments

(InCollection, Churchill: CVE, Springer Verlag 2001)

Report number: CID-155
ISSN number: ISSN 1403 - 0721 (print) 1403 - 073 X (Web/PDF)
Publication date: August 2001
E-mail of author: kai@nada.kth.se

Reports can be ordered from:

CID, Centre for User Oriented IT Design
NADA, Deptartment of Numerical Analysis and Computer Science
KTH (Royal Institute of Technology)
SE- 100 44 Stockhom, Sweden
Telephone: + 46 (0)8 790 91 00
Fax: + 46 (0)8 790 90 99
E-mail: cid@nada.kth.se
URL: http://cid.nada.kth.se

Chapter 1

How not to be objective

Kai-Mikael Jää-Aro, Dave Snowdon

1.1 Introduction

There is today a spate of Collaborative Virtual Environments (CVEs) on the net (`http://www.digitalspace.com/avatars/` indexes them), similar to the Metaverse in Neal Stephenson's *Snow Crash* (Stephenson 1992). Some of these allow the users to construct buildings in the world, though in the majority the environment is fixed. These buildings serve as meeting points and atmosphere creators, but do not have any other meaning. *Information* can be presented in these spaces as imported objects or as placards with text and images placed in the environment.

Spaces such as these are being evaluated by various organisations as meeting places to increase social cohesion among geographically distributed colleagues. But while these environments lead to increased awareness and spontaneous communication, they do not give a good feeling for what one's coworkers are *doing* (Lenman 1999). This is fairly obvious, as the actual work activity is not supported in, and therefore takes place outside, these virtual environments.

We will here outline a way in which a certain type of work can be brought into a virtual environment and the concomitant requirements that places on the environment. To be more specific, we are concerned with 3D visualisations of abstract data, such as the contents of data bases or other discrete data. We further make the case that in order to afford useful work such and environment must allow *subjective* representations of data.

Finally, we will describe a prototype implementation we have done of such a 3D data visualisation with subjective presentation.

1.2 Populated Information Terrains

Consider a relational database. The data contained in it have no intrinsic geometric shape, so in order to be able to represent them in 3D virtual environment, a *mapping* from data to 3D space and 3D geometry has to be done. Since we also represent the users of these data in the environment, this type of data visualisations have been termed *Populated Information Terrains* (PITs) (Benford, Snowdon and Mariani 1995, Mariani 1998). As indicated in the introduction, the aim is that this type of shared data space enable the important features of *peripheral awareness*, *informal meetings* and *learning by watching*.

Peripheral awareness It is well documented (e g (Heath and Luff 1991, Bentley, Hughes, Randall, Rodden, Sawyer, Shapiro and Sommerville 1992, Bowers, Button and Sharrock 1995)) how physically co-located workers, in e g a control room or an open plan office, keep track of each others' work progress in order to take into account the effects of this on their own tasks. Workers also often use body language and tone of voice to indicate to others that they *should* monitor some unfolding event in that worker's task.

In a PIT, workers within an organisation are visible to each other, even if not physically co-present, as is increasingly more common in contemporary white-collar work where employees may be geographically distributed, telecommuting or just in an office of their own. It has been shown that even fairly simple avatars allow the expression of important human interactions (Bowers, Pycock and O'Brien 1996). Furthermore, if the users' interactions with data are visualised in an effective manner, they will be apparent to other users, displaying what user X is doing with datum B.

Thus a PIT user will be able to gain a feel for what work currently is being done within her organisation. Likewise, needed help can easily be raised among those present by gesturing, speaking louder or other such relatively unobtrusive means.

Now, while we would not promote computer-mediated communication in preference to everyday face-to-face contact, we still note that having to work in the same space as many others has drawbacks, both in terms of lost privacy and a noisy environment. A PIT could give the best of both worlds, in that workers can have an awareness of each other, but that their actions are *filtered* through the computer system and that they therefore need not give up more of their privacy than they choose to.

Informal meetings Unplanned and informal meetings are essential in order to coordinate work and disseminate information in an organisation (Kraut, Fish, Root and Chalfonte 1990, Whittaker, Frohlich and Daly-Jones 1994) and since even being on separate floors drastically decreases the chances for informal encounters (Kraut, Egido and Galegher 1988), there has been some development of CSCW tools that are intended to increase the frequency of chance encounters in distributed work groups (Isaacs, Tang and Morris 1996, Huxor 1998).

Learning by watching Another process that is missed by dispersed members of a work group is the mutual transfer of practical knowledge of, in this case, the computer system. Informal studies have shown that users of a CVE observe the actions of other users, and pick up knowledge of previously unknown functions in the system.

1.2.1 Ease of use

An underlying assumption of the previous statements is that the CVEs are in more or less continuous use, as the described benefits require awareness of the environment over some time. Few current CVEs are designed to remain in the background of a desktop environment for long periods of time, if for no other reason, then due to the lack of screen space. It may be that useful utilisation of this kind of environments will require multiple monitor screens on or around one's desk (Lenman 1999, Stevens and Papka 1999).

1.3 Subjectivity

One could imagine several techniques for visualising the same data set. If a data set has no "natural" representation (unlike a 3D CAD model for example) the choice of a particular technique will depend on the task at hand or on the preferences of the user. It is therefore natural for visualisation systems to present the users with a number of choices which govern the nature of the visualisation presented to them.

However, in most current systems all users are forced to use the same representation of the information and thereby trade flexibility for the ability to collaborate in the use of the information.

An alternative to this is to allow each user to have her private view of the data, even though it be a collaborative application. We refer to this as a *subjective* environment (Snowdon, Greenhalgh and Benford 1995).

We define subjectivity as the ability to add viewer dependent features to a system (where a *viewer* may be a "real" user or a software agent). In other words the information presented to a viewer may depend on the nature of the viewer.

This viewer dependence could either be on the basis of individual viewers or on the class or role that the view belongs to.

1.3.1 2D Subjective User Interfaces

The strictly objective WYSIWIS (What You See Is What I See) paradigm for CSCW implies that co-workers participate in a tight loop of interaction where everyone has to see the object(s) of discourse and comment on each other's changes to it/them. This is based on a conception of work as being in front of a whiteboard or by a draughting table where a group of co-workers jointly develop a design. However, even in these non-computerised cases work is not tied to a single representation—co-workers may leave and enter the room to retrieve additional material, may do alternative sketches on private notepads, or simply be thinking of other things; furthermore the work may be organised in such a way as to not always entail the concurrent, synchronous work with identical data. This has been acknowledged in several CSCW systems that may be nominally synchronous, yet allow different representations and/or perspectives of data, a "Relaxed WYSIWIS" (Stefik, Bobrow, Foster, Lanning and Tatar 1987, Greif and Sarin 1987, Shu and Flowers 1994).

A number of groupware systems support configurable user interfaces, for example Mead (Bentley, Rodden, Sawyer and Sommerville 1992) and Rendezvous (Patterson, Hill, Rohall and Meeks 1990). These systems allow users to view the shared data in radically different ways. For example, the air traffic control application Mead is capable of representing aircraft either as a flight strip or as a "radar blip" and allows each user to select the representation most appropriate for their needs.

Bowers *et al* (1998) have argued that customised displays are detrimental for the above-mentioned peripheral awareness. If all co-workers have personalised information displays, it will be difficult for others to grasp the situation at a glance. We claim that this is irrelevant for the case at hand, since we don't expect the members of the work group to be in the same physical room, the view they get of the environment is *through* the computer system, and thus the stable, familiar view of one's own customised environment. However, we readily acknowledge that a computer-mediated presence and situational awareness might not be the best for all situations and that a display layout in physical space may well be the best way to create a working environment. The decision of when one or the other is better is however beyond the scope of the current discussion.

1.3.2 Subjective Programming Environments

Subjectivity is not only applicable to user interfaces. Smith and Ungar (1994) argue that subjectivity could be a valuable addition to a programming environment and give two examples where subjectivity might be useful in a programming environment:

1. Supporting group programming. A group of programmers may independently make changes to a system without conflict if each person's changes are only apparent when used by that person. By explicitly using someone else's view another person could elect to use that person's version of the system without having to merge different versions of the program source code.

2. Supporting *capabilities*. Many systems provide different users with different capabilities. For example only certain people may be allowed to read certain files, while others files may be read by all users but only modified by the system administrator. In a strictly objective environment these mechanisms must be provided by dedicated software. The use of subjectivity would allow such capabilities to be implemented without the need for special purpose software.

Smith and Ungar also describe a subjective object-oriented language called "US" (an extension to the Self language (Ungar and Smith 1987)) in which the effect of a message sent to an object depends on the *perspective* through which the message is sent. In US perspectives are extra levels of indirection between the sender and the receiver of a message which allow a message to be processed differently depending on which perspectives it passes through.

1.3.3 The Need for Subjective Virtual Environments

Most current multi-user virtual reality systems provide a highly objective virtual environment. That is, all users see the environment in the same way, albeit from different viewpoints, and all users see the same objects in the same places with the same appearances. This is partly due to the fact that multi-user VR systems evolved from single-user systems which have been extended to support a number of users. This aspect of the evolution of VR systems is similar to the way that groupware systems evolved from systems such as Shared X (Gust 1988), that simply replicated an application's user interface to multiple users. We argue that, just as strict objectivity proved too constraining in other groupware systems, a degree of subjectivity is necessary to support collaboration in virtual environments. Smith (1996) shows how an existing 2D groupware toolkit, SOL, can be extended to provide customisable subjective virtual environments.

Compared to WYSIWIS even an objective VE could be considered to support some degree of subjectivity. We shall first indicate some "subjective" features found in standard VEs and then give some examples of situations in which it might be useful to allow more radically different subjective views.

- Each user will typically have an independently controlled viewpoint from which they can inspect the virtual environment. This is our experience of the real world and as such we expect people to be skilled at relating to viewpoints and reasoning about these kinds of subjective effects. (Though admittedly current virtual environments tend to confound this ability by, *int al*, the limited field of view (Hindmarsh, Fraser, Heath, Benford and Greenhalgh 1998, Hindmarsh, Fraser, Heath and Benford 2000).)

- Level-of-detail (LOD) effects. A common technique in interactive computer graphics is to provide several different representations of a given artefact, each with a different level of complexity. When viewing the object from a distance the user is shown a simple version of the artefact which is replaced by more detailed representations as the user approaches. It should be noted that the purpose of level-of-detail effects is to improve the interactive performance of the display software rather than to support subjective effects—in fact the normal aim is for users not to notice when one representation of an artefact is substituted for another. Also, since these alternative representations of an artefact can be packaged as a single data item for the purposes of distribution (or generated automatically by the renderer), adding LOD effects requires only that the renderer need be modified and the rest of the VR system can remain strictly objective.

Unfortunately, independently controlled viewpoints and LOD effects are special cases and support for these features does not allow more radically different subjective views to be created. It is therefore necessary to modify an existing VR system if truly subjective effects are desired. We shall now give some examples to justify why this additional implementation effort is worthwhile.

A 3D chat space benefits from an agreed-on layout as it simplifies navigation for its inhabitants, they can learn landmarks in the environment, agree on meetingpoints etc. Apart from the design of landmarks in order to make the landscape easier to navigate, the landscape may be shaped in order to increase the probability of particular types of encounters (Bowers 1995).

However, a database visualisation has quite different requirements. By its very nature it will not be amenable to landscaping. While the underlying relations (assuming a relational database for the sake of argument) may be fairly stable, their use is not—information is continually dynamically combined in new ways and the mappings from data to 3D space cannot be known beforehand.

It then seems reasonable that users may have different preferences for how particular views of data should be displayed, it may even be that different users need to experiment independently with alternative displays, even if their ultimate goal is to have the same common view of data.

On a more prosaic level, different users may have different levels of access to particular pieces of data, wherefore not all users can be allowed to see all data items. Moreover, users may carry around displays and tools in the virtual environment which are entirely private to themselves and therefore should not be displayed to others.[1]

If the virtual environment does not support subjective views then the users are denied the ability to work in parallel with different representations of the same data, and forced to work with those subsets of data that all users are allowed to see—or those subsets of users that are allowed to see all data. If on the other hand the virtual environment is capable of supporting subjective views then users are free to choose their own preferred visualisation style.

Leaving business data bases aside, another use for subjective views is in computer aided design. Consider an architectural review of a building. Whilst the people taking part in the review would need a overview of the building they might only want to see features relating to their speciality in detail—for example, an electrician might only want to see wiring plans in detail and not see the plans for the plumbing except in cases where there was a potential conflict (Leigh, Johnson, Vasilakis and DeFanti 1996).

A final example would be the use of subjectivity to enable multi-lingual text display in a virtual environment (Smith 1996). In a subjective environment users would be able to explore the same basic environment, but see all textual annotations in their preferred language.

1.3.4 Degrees of Subjectivity

Figure 1.1 shows various classes of systems according to how much subjectivity they support. At the entirely objective end of the scale we have WYSIWIS user interfaces and normal objective programming environments (typified by the Self language in this example). Relaxed-WYSIWIS is not entirely objective and 3D virtual environments (assuming they support independent viewpoints and LOD) are slightly less so. At the entirely subjective end of the scale we have relational databases (since any number of views of the same data can be constructed to

[1]An illustrative historical note: In the first versions of the DIVE system all graphics were indeed visible to all users, so that one could see the navigational icons of a user floating immediately in front of the face of that user's avatar. In later versions such displays have been "privatised" and are no longer visible to others.

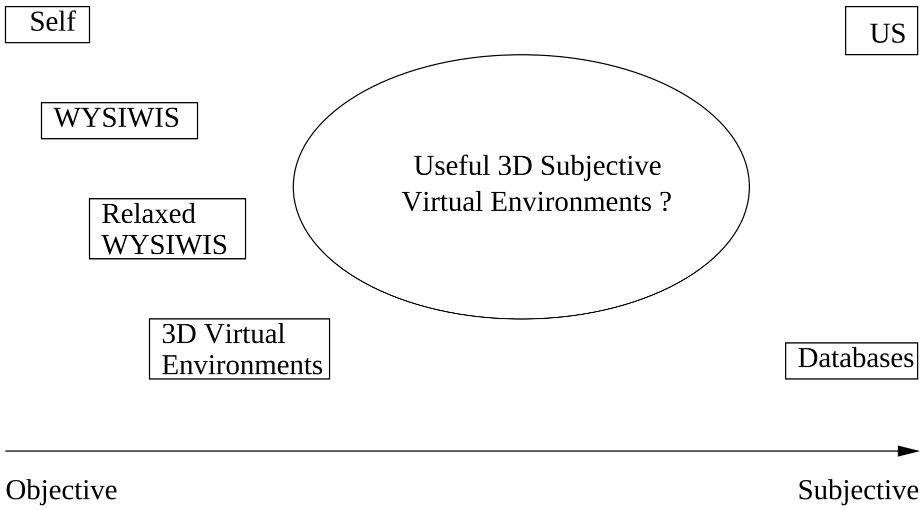

Figure 1.1: Scale showing degrees of subjectivity of several classes of system.

show different combinations and sub-sets of the database's field and contents) and subjective programming environments (typified by US in this example).

We have already argued that some degree of subjectivity is useful in collaborative virtual environments—however it seems obvious that if we take subjectivity to extremes then users will effectively be inhabiting different environments and will no longer be able to collaborate effectively (assuming they are still even aware of each other). We therefore hypothesise that *useful* subjective virtual environments span a range that is slightly more subjective than existing VEs but less subjective than databases and subjective programming environments (such as US).

Even so, in a shared subjective environment with a visual representation (be it 2D or 3D), where users wish to indicate objects to each other, a user may not see any object or, worse, see a completely different object at a spot indicated by another user with a different object layout. Smith has solved this problem by two methods: First, by letting a low-lighted version of the other user's world view become visible, if two users seem to be interested in related items; second, by highlighting indicated objects directly, rather than using pointers in space (Smith and O'Brien 1998).

1.4 Creating a Subjective Environment

In this section we describe some of the ideas and concepts behind our prototype environment which allows us to maintain some awareness between users who may have different views of the environment. We shall first describe *artefact centred coordinates* which allow us to represent users to other users with different subjective views and *body centred configuration*, a technique to allow users to specify their preferences for the display and behaviour of applications in a subjective virtual environment.

1.4.1 Artefact-Centred Coordinates

It is our experience that embodiments or avatars, visual representations of the users, are necessary in collaborative virtual environments. The embodiments indicate the presence and activities of a particular user to other users and also serve as the focus for communication—allowing "virtual face-to-face conversations" (Benford, Greenhalgh, Fahlén and Bowers 1994, Bowers et al. 1996). Since the artefacts in one user's view may be in very different locations in another user's view we cannot place user embodiments at the same world coordinates in each view, but must come up with some other mechanism for positioning user embodiments in subjective views. An empty space conveys no useful information; it is the artefacts, other users in the space and their actions in relation to one another that provide us with information. We therefore consider the position and orientation of a user in relation to the artefacts or other users in the environment rather than in terms of the user's location in world coordinates. We do this using a technique which we term *artefact centred coordinates*, which uses the artefacts the user is aware of to determine the position and orientation of that user in other subjective views.

The basic concept behind artefact-centred coordinates is to compute a user's *awareness* of a set of artefacts, find which of these artefacts exist in the subjective view we want to represent the user in (the *target* view) and place the representation of the user (the *pseudobody*) in a position and orientation determined by the location of the artefacts that the user is aware of in the target view, as shown in Figure 1.2.

The concept of awareness has been introduced in the Spatial Interaction Model (Benford, Bowers, Fahlén, Greenhalgh, Mariani and Rodden 1995). Briefly, all active objects in a CVE have, for each communication modality, a *focus*, a *nimbus* and, optionally, an *aura*. Conceptually, the focus is the representation of an object's *presence* in space, whereas the focus is the object's *attention* in space. An object's focus intersecting another object's nimbus makes the first object *aware* of the other, enabling subsequent interaction.

The standard formulation of the Spatial Interaction Model defines the focus, nimbus and aura in terms of volumes, but, following the argument in (Bowers, Jää-Aro, Hellström and Carlzon 1999), we choose a formulation that is similar, though not identical, to the one by Greenhalgh (1997). We define the focus and nimbus as functions that take on a value in the interval $[0, 1]$ for any point in

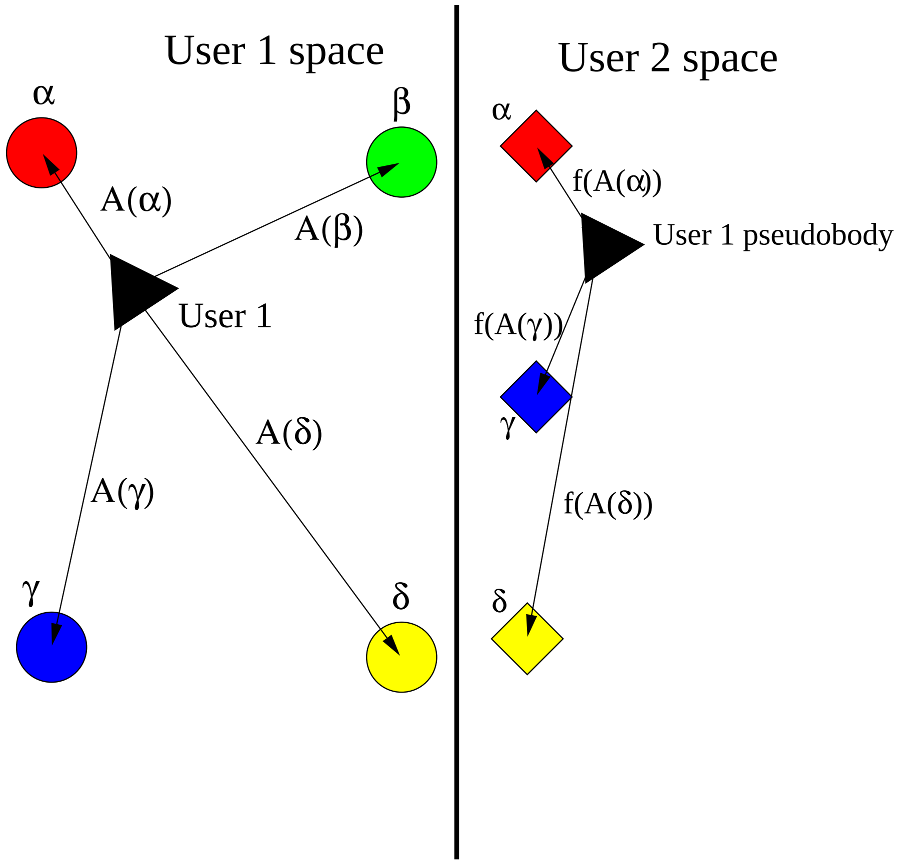

Figure 1.2: User 1, in her subjective view, is aware of the objects α, β, γ and δ, in that decreasing order. To make this information available to User 2, the pseudobody representing User 1 is, in the subjective view of User 2, placed proportionately closer to the objects User 1 is more aware of. As object β is not present in User 2 space, it is ignored in the computation.

3-space. The awareness computation then becomes

$$awareness(a, b) = focus(a, b) \cdot nimbus(b, a).$$

The aura is most suitably interpreted as some simple bounding volume (sphere or box) that completely encloses the points for which the *focus* and *nimbus* functions of an object have values > 0, so for example, if an object has the focus and nimbus functions

$$focus(a,b) = \begin{cases} 1 & \text{if } |position(a) - position(b)| < 4 \\ 0 & \text{otherwise} \end{cases}$$

$$nimbus(a,b) = \begin{cases} 1 & \text{if } |position(a) - position(b)| < 6 \\ 0 & \text{otherwise} \end{cases}$$

then the aura could be a sphere with a radius ≥ 6. In this way one can do a simple collision test for the auræ of two objects before doing a potentially complex focus and nimbus computation. In the given example focus and nimbus are extremely simple, but presumably an "intelligent" awareness function that approximates how a real human apportions interest in the environment would be considerably more complex and depend on a number of factors, but this doesn't affect our basic argument.

In our implementation we have considered *focus* and *nimbus* to be purely geometrical functions, i e $f(position(a) - position(b))$.

The Aether extension to the Spatial Interaction Model (Şandor, Bogdan and Bowers 1997), introduces *time* as an additional coordinate, we could thus allow e g the number of interactions with an object over a period of time to affect the awareness computation. We will however not pursue this issue further here.

For each of the subjective views in which we intend to represent the user we must then find the subset of artefacts for which we have calculated the user's awareness that are present in the target view (if this subset is the empty set we can choose a suitable default location or not represent the user at all in that particular view). Our current method for placing the representation of the user in other users' views is simply to use a weighted sum of the artefacts' positions with the awareness of each artefact acting as the weight. This has the result that users are displayed close to the artefacts of which they are most aware. Likewise we compute the direction of the avatar's gaze—using quaternions to represent angles lets us do averaging over directions—and display the user's pseudobody turned towards those artefacts to which she is mostly turned towards in her own subjective view.

If interaction with objects is represented in some particular manner, this interaction should be visualised in the target view as well. If this interaction is represented by high-level events, they can be translated to the appropriate visualisation in the target view.

1.4.2 Body-Centred Configuration

In an application of interesting complexity there will be set of adjustable features, such as default data set, entry point in the world, enabled user interface features (available interaction hardware, window size, etc). However, in a distributed environment, the application that would need to receive these preference settings may not be allowed to do so, since the application may be running at a node which does not have read access to the file store of that particular user. We might of course simple require all users to place their preference settings at a "well-known place" or even store them at the node where the application runs, if this is constant over time.

However, we propose an alternative approach. Since the users' embodiments are projected into the environment to represent them to themselves and other users, we suggest considering a user's preferences as properties of their embodiment that "represent" them to applications and allows these to inspect and store non-geometric information in the user's embodiment. We refer to this technique as *Body Centred Configuration*. This is conceptually similar to the resource database provided by the X Window System (Scheifler and Gettys 1986) which (possibly remote) applications can inspect to retrieve information about user configuration options. Using Body Centred Configuration, an application is capable of configuring itself to adapt to the user's default preferences as soon as the user enters the world in which the application is resident with no explicit intervention on the part of the user.

1.5 A Subjective Application

To test our ideas for subjective data representation we elected to use a modified version of VR-VIBE. VR-VIBE is a multi-user 3D visualisation of a collection of documents or document references (Benford, Snowdon, Greenhalgh, Ingram, Knox and Brown 1995). Figure 1.3 shows several different visualisations of the same data set. Selecting a document icon causes some summary information to be displayed. If a document is available via the World Wide Web then VR-VIBE can invoke a web browser to display the entire document contents.

We chose VR-VIBE as our test application because we had access to the source code and therefore could adapt it for subjective operation and because it already supports a number of visualisation styles allowing the creation of subjective views that may differ substantially. VR-VIBE is implemented in DIVE.

1.5.1 DIVE

DIVE is a distributed virtual environment platform developed by the Swedish Institute of Computer Science (available at `http://www.sics.se/dive/`). DIVE is based on a notional shared database of objects, these are in fact replicated to all members of the process group that defines a world. Updates to an object at one

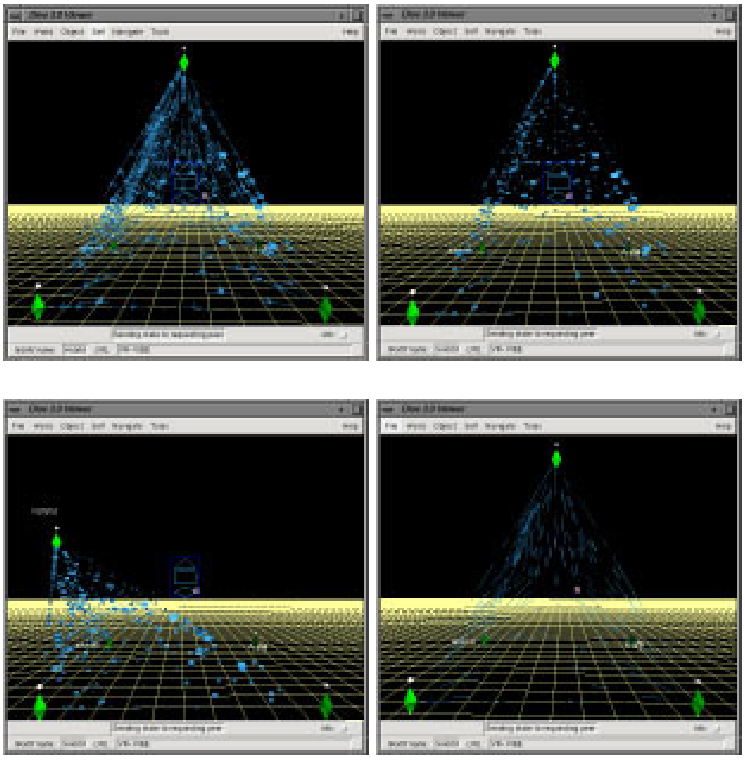

Figure 1.3: Different visualisations of a bibliography database. User-provided keywords (represented as green octahedra) are used to provide a spatial framework for the visualisation. The documents are positioned in the space according to how well they match each of the keywords. Additional information can be given by lines, the length of which indicate the strength of relevance to the keywords.

node are distributed to all other nodes using IP multicast. DIVE also supports partial replication such that not all objects are automatically distributed to all nodes but are kept in a process subgroup that has to be explicitly joined. DIVE provides other useful features that greatly simplified our implementation:

- Support for partitioning a world between several multicast groups. Associating a lightweight group with each user provides an efficient mechanism for implementing subjective (viewer-dependent) views of parts of a DIVE world.

- All objects, including bodies, may contain Tcl code (Ousterhout 1994) that gives the objects behaviour, so that they can react to user interaction, events in the environment or perform application computation. In this manner computation can be distributed over the objects (even though the actual computation normally will take place at a single node and then be distributed to all other member nodes). Objects/processes can send Tcl/Tk code to other objects/processes, providing a mechanism for application-level inter-process communication.

- All objects may in addition use *properties* to hold information about the object which is publicly accessible to other objects and applications. Properties consist of tuples specifying the name, type and value of each property. Applications can augment objects with additional properties as a means of storing application specific information. The property mechanism provides a very powerful way for applications and objects to exchange arbitrary information via the DIVE database.

In addition to this list DIVE has some other features which are generally useful:

- Human participants have an embodiment in the environment. This body can be shaped in any manner, but the DIVE visualizer supports the notion of body parts, such as eyes, feet, head, etc. and if parts of the body are labelled accordingly they will be used for viewpoints, ground reference etc.
- DIVE can import VRML files.
- DIVE object files are passed through a macro-expanding pre-processor before parsing, which allows very rapid and efficient construction of complex objects.

We found the combination of Tcl/Tk and properties to be very powerful when developing applications. In our prototype the entire user interface is written using Tcl/Tk which calls the various application processes to invoke specific operations. Each application process exports globally visible data as DIVE properties which can then be inspected by other objects or processes. This allows each application process to perform only a small number of well-defined functions with Tcl code providing the "glue" required to create a complete system. This modular organisation allows components to be modified or replaced without needing to update the other system components.

1.5.2 Subjective VR-VIBE

In order to support subjective visualisations VR-VIBE needs to be capable of generating a different visualisation for each user. It does this by explicitly separating objective and subjective state information. The objective state contains the content of the document store and other information that is not dependent on the nature of the generated visualisation. There is exactly one copy of the objective state information. The subjective state contains the objects used to create a specific visualisation and any other parameters that are dependent on the nature of the visualisation (configuration options etc.). There is an instance of the subjective state for each subjective view generated by VR-VIBE.

DIVE does not by itself support data protection, but that can be simulated by adding an access list property to each restricted object and let the translator only distribute it to views owned by members of that access list.

1.5.3 Subjective Embodiments using Artefact Centred Coordinates

As noted above, all processes are members of the world multicast group, but in addition may be in a group of their own that we use to contain their subjective view of the world. We cannot place the user embodiments in the world multicast group because, since objects are (can be) placed differently in the subjective views, a user's coordinates are meaningless in any other view than her own. The solution, as described above, is to have different representations of each user in each subjective view. In her own view the user is represented by the normal embodiment mechanism provided by the DIVE visualizer process, in other views the representation of the user is handled by the subjective translator process which displays a pseudobody[2] at a position and orientation which attempts to convey maximal information about the user's activities to the other users.

Figure 1.5 shows the relationship between the standard user embodiments provided by DIVE, the subjective translator and the subjective embodiments created by the subjective translator. The DIVE visualizer process manages the representation of a user in her own view, and the translator process manages the user representation in all other users' views. Figure 1.6 demonstrates what it actually looks like.

1.5.4 Translation functions

Our experiences with different functions for computing awareness and translating position to other spaces have indicated the following:

- The definition set of the *focus* function should not be too large. We first tried using $\mathbb{R}^3$ as the definition set, defining

$$focus(o_1, o_2) = nimbus(o_1, o_2) = \frac{1}{|position(o_1) - position(o_2)|}.$$

[2] At the moment each user gets to define her own representation in the views of others, but we are exploring ways in which to allow users to choose their view of other users as well.

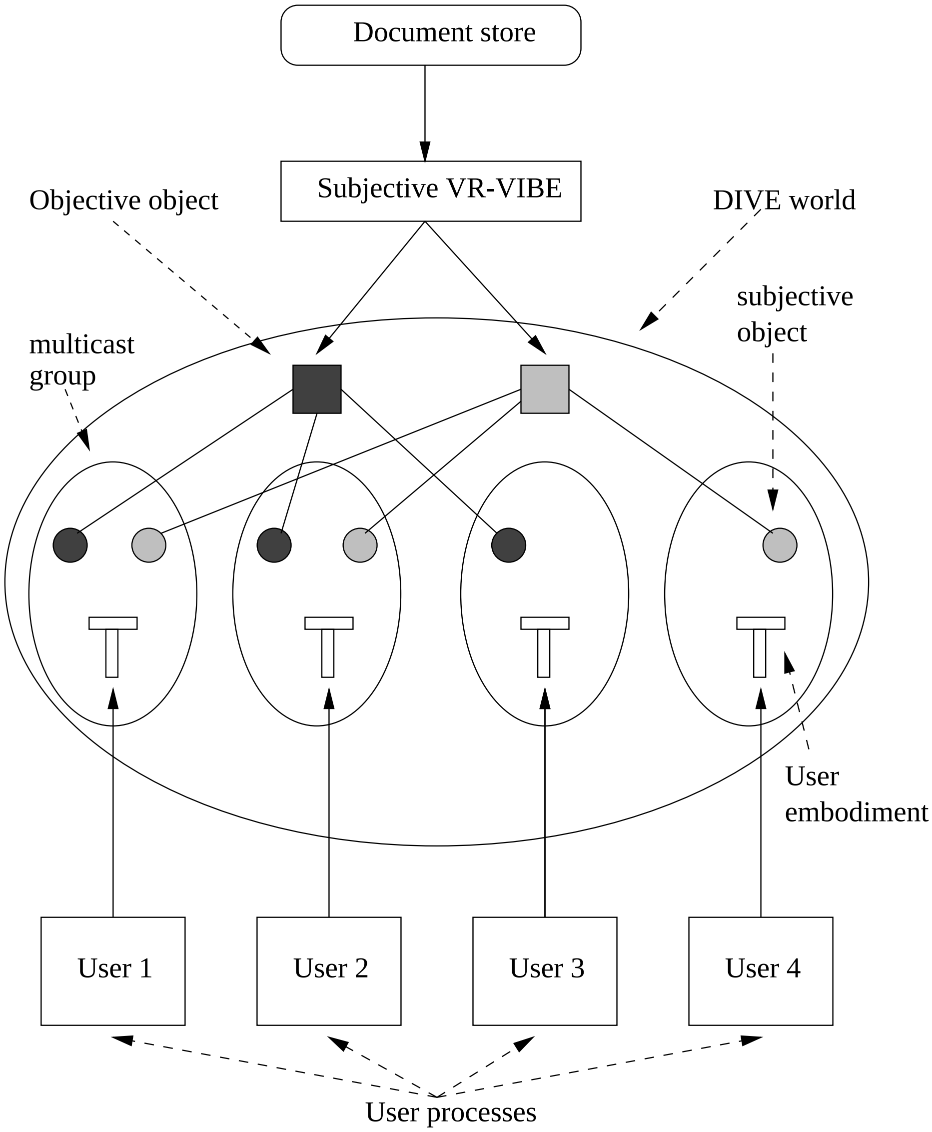

Figure 1.4: The structure of a subjective environment constructed by VR-VIBE. The main DIVE world (which also has a multicast group associated with it) contains only DIVE objects containing objective state information. Each subjective view has its own dedicated multicast group which contains the embodiment of the user "owning" that view and the subjective state information for that view. For simplicity we do not show the subjective representations of other users which must be present in order for users to be aware of one another. User representation is handled by another process, as described in section 1.5.3.

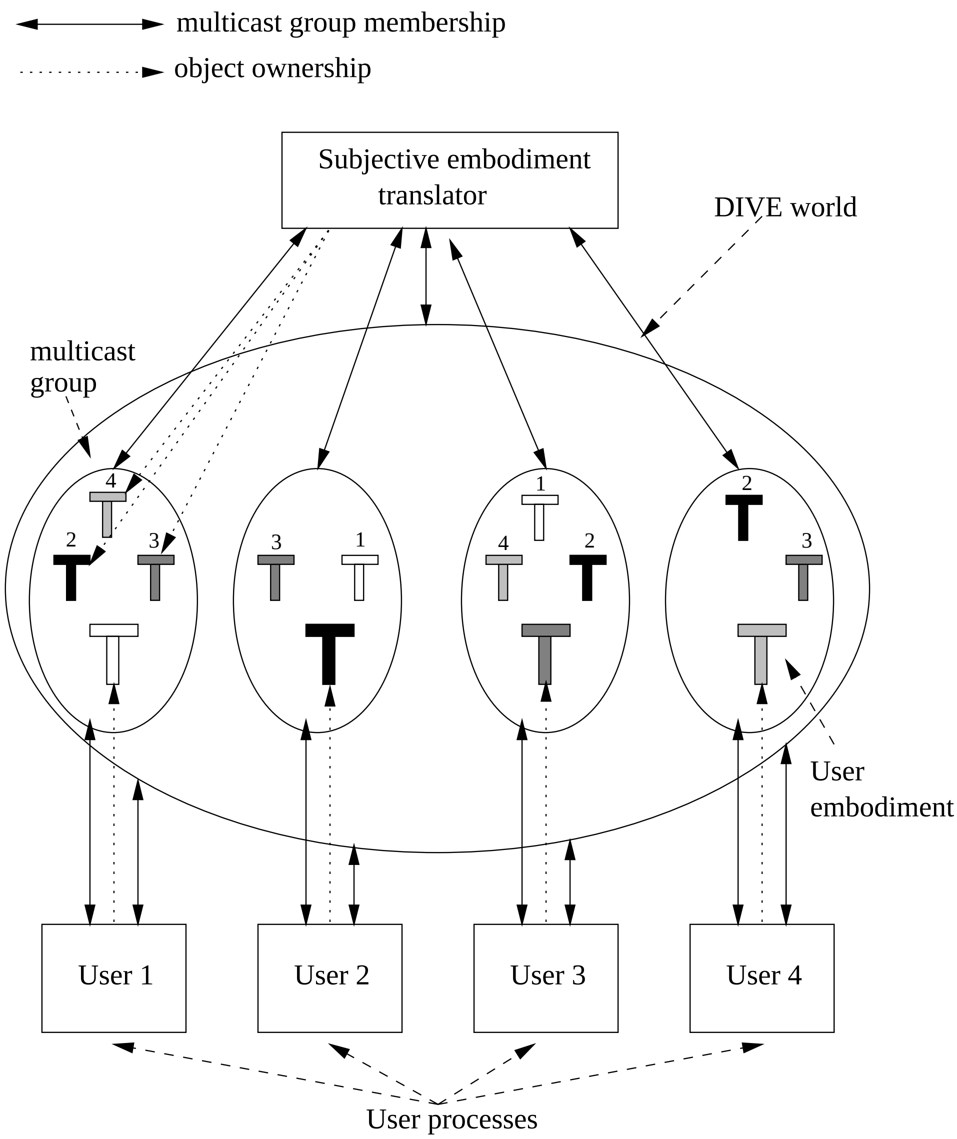

Figure 1.5: Solid lines indicate multicast group membership and dashed lines indicate the process responsible for an object. Object ownership by the subjective translator is only shown for the view owned by User 1.

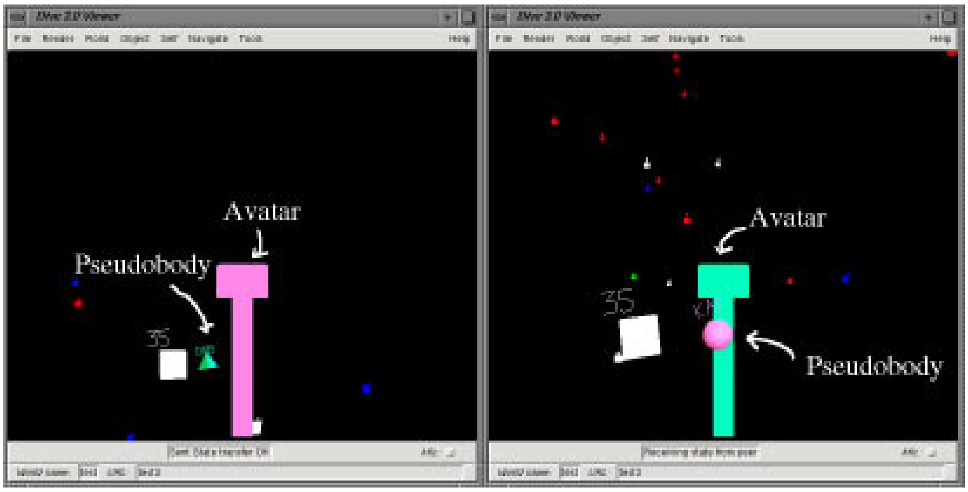

Figure 1.6: A screenshot from a test run with subjective translation. Several hundred objects have been randomly (and differently) placed by two users in their respective subjective views. In the left window we see the T-shaped "blockie" avatar of user `KAI` approaching object #35. The tetrahedral pseudobody of user `DNS` hovers nearby. In the right window we see the converse, the spherical pseudobody of user `KAI` in the subjective view of user `DNS`. We have chosen to have very different shapes for avatars and pseudobodies to make the difference clear, but this is of course not enforced by the system.

This had the undesirable consequence that objects very far away had an inappropriately large effect on the positioning of the pseudobody—the awareness of them might be low, but if they were relatively far away in the target view, they could still pull away the pseudobody from the position that "looks right". Therefore, the *focus* and *nimbus* functions should assume the value 0 at not too large a distance from the object origin, the exact distance depending on the application.

In general the use of an awareness function which does not decrease (or at least stay constant) with increasing distance, will lead to unexpected behaviour in our current implementation.

- If a user moves from one artefact to another, which is nearby in her view, these may be far apart in a target view. The pseudobody may thus suddenly jump from one place to another. As this makes it difficult to follow a pseudobody around we have felt the need to limit the maximum speed of a pseudobody and animate its motion between distant points.

1.6 Discussion

It seems obvious that if users' subjective views diverge beyond a certain point then meaningful collaboration will become difficult or impossible. On the other hand we believe that if users are forced to share the same objective view then collaboration may occasionally become awkward if users have to continually negotiate an agreed common view—this was found to be the case for strictly WYSIWIS 2D shared editors and lead to the development of the Relaxed WYSIWIS concept. This has lead us to believe that a subjective environment that allows users to adjust their subjective views to suit individual needs, but still retain awareness of others and the ability to communicate may be a useful tool.

We would imagine that over the course of a session users' views may converge and diverge depending on the situation. We will therefore need to develop techniques that allow users to switch to a common view or to find information on how divergent their view has become from other users. It may also be useful for the system to monitor the parameters for each user's subjective view and allow users to switch to the currently most popular view.

It is worth noting that our current implementation allows objective applications to run in the same world without modification. If an objective application belongs to the world's multicast group then it will be available to all users in the world. Alternatively, objective applications might belong to a given user's multicast group and thereby be completely private to that user. This would allow users to still use tools such as the DIVE whiteboard whilst also using a subjective visualisation.

1.7 Summary

In this chapter we have argued that in some situations providing subjective or viewer-dependent features in an environment may actually aid collaborative information visualisation by allowing each user to tailor their view while still maintaining contact with other users. We have described a prototype information visualisation based on this ideas. Since our implementation is based on a collection of applications it would be relatively easy to replace some components to provide a different of features (e g to visualise other sorts of abstract information).

Acknowledgements

Many thanks to Lennart Fahlén and the Swedish Institute of Computer Science (SICS) for funding Dave Snowdon as a guest researcher at SICS whilst subjective VR-VIBE was implemented. We also thank Chris Greenhalgh and John Bowers for enlightening discussions, Olof Hagsand, Mårten Stenius and Emmanuel Frécon for DIVE help and the editors for encouraging and insightful comments.

Bibliography

Ackerman, M. S. (ed.): 1996, *CSCW '96: Cooperating Communities*. http://www.acm.org/pubs/contents/proceedings/cscw/240080/index.html.

Benford, S., Bowers, J., Fahlén, L. E., Greenhalgh, C., Mariani, J. and Rodden, T.: 1995, Networked virtual reality and cooperative work, *Presence: Teleoperators and Virtual Environments* **4**(4), 364–386.

Benford, S., Greenhalgh, C., Fahlén, L. E. and Bowers, J.: 1994, Embodiment in collaborative virtual environments, *in* B. Pehrson and E. Skarbäck (eds), *Proceedings of 6th ERCIM Workshops*, ERCIM.

Benford, S., Snowdon, D., Greenhalgh, C., Ingram, R., Knox, I. and Brown, C.: 1995, VR-VIBE: A virtual environment for co-operative information retrieval, *in* F. Post and M. Göbel (eds), *Proceedings Eurographics '95*, Eurographics Association, Blackwell Publishers, pp. 349–360.

Benford, S., Snowdon, D. and Mariani, J.: 1995, Populated information terrains: first steps, *in* R. A. Earnshaw, J. A. Vince and H. Jones (eds), *Virtual Reality Applications*, Academic Press, chapter 3, pp. 27–39.

Bentley, R., Hughes, J. A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D. and Sommerville, I.: 1992, Ethnographically-informed systems design for air traffic control, *in* Turner and Kraut (1992), pp. 123–129. http://www.acm.org/pubs/articles/proceedings/cscw/143457/p123-bentley/p123-bentley.pdf.

Bentley, R., Rodden, T., Sawyer, P. and Sommerville, I.: 1992, An architecture for tailoring cooperative multi-user displays, *in* Turner and Kraut (1992), pp. 187–194. http://www.acm.org/pubs/articles/proceedings/cscw/143457/p187-bentley/p187-bentley.pdf.

Bowers, J.: 1995, The social logic of cyberspace, or, the interactional affordances of virtual brutalism, *in* A. Bullock and J. Mariani (eds), *Esprit COMIC*

Project Deliverable 4.3: Assessment and Refinement of Models of Interaction, chapter 6, pp. 89–145. ftp://ftp.comp.lancs.ac.uk/pub/comic/D4.3.ps.Z.

Bowers, J., Button, G. and Sharrock, W.: 1995, Workflow from within and without: Technology and cooperative work on the print industry shopfloor, *in* H. Marmolin, Y. Sundblad and K. Schmidt (eds), *Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work*, Kluwer Academic Publishers, pp. 51–66.

Bowers, J., Jää-Aro, K.-M., Hellström, S.-O. and Carlzon, M.: 1999, Event management in electronic arenas by visualising participant activity and supporting virtual camera deployment, *in* S. Hirtes, M. Hoch, B. Lintermann, S. J. Norman, J. Bowers, K.-M. Jää-Aro, S.-O. Hellström and M. Carlzon (eds), *Esprit eRENA Project Deliverable D4.3/D4.4: Production Tools for Electronic Arenas: Event Management and Content Production*, chapter 3, pp. 58–79. http://www.nada.kth.se/erena/pdf/D4_3-D4_4.pdf.

Bowers, J., Martin, D. and Wastell, D.: 1998, An ethnography of ambulance control to guide the development of information visualisations for electronic landscapes, *in* Mariani, Rouncefield, O'Brien and Rodden (1998), chapter 2, pp. 35–58. http://bscw.comp.lancs.ac.uk/pub/english.cgi/d64394/Escape-D31.pdf.

Bowers, J., Pycock, J. and O'Brien, J.: 1996, Talk and embodiment in collaborative virtual environments, *CHI '96: Human Factors in Computing Systems: Common Ground*, SIGCHI, pp. 58–65. http://www.acm.org/pubs/articles/proceedings/chi/238386/p58-bowers/p58-bowers.html.

CSC: 1998, *Proceedings of the ACM 1998 conference on Computer supported cooperative work*. http://www.acm.org/pubs/contents/proceedings/cscw/289444/.

Galegher, J., Kraut, R. E. and Egido, C. (eds): 1990, *Intellectual Teamwork—Social and Technological Foundations of Cooperative Work*, Lawrence Erlbaum Associates.

Greenhalgh, C.: 1997, *Large Scale Collaborative Virtual Environments*, PhD thesis, University of Nottingham. ftp://ftp.crg.cs.nott.ac.uk/pub/cmg/cmg-thesis.ps.gz.

Greif, I. and Sarin, S.: 1987, Data sharing in group work, *ACM Transactions on Office Information Systems* **5**(2), 187–211. http://www.acm.org/pubs/articles/journals/tois/1987-5-2/p187-greif/p187-greif.pdf.

Gust, P.: 1988, Shared X: X in a distributed group work environment, *2nd Annual X Conference*.

Heath, C. and Luff, P.: 1991, Collaborative activity and technological design: Task coordination in London Underground control rooms, *in* L. Bannon, M. Robinson and K. Schmidt (eds), *Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, Kluwer Academic Publishers, pp. 65–80.

Hindmarsh, J., Fraser, M., Heath, C. and Benford, S.: 2000, Virtually missing the point: Configuring CVEs for workplace interaction, *in* E. Churchill, D. Snowdon and A. Munro (eds), *Collaborative Virtual Environments*, Springer Verlag.

Hindmarsh, J., Fraser, M., Heath, C., Benford, S. and Greenhalgh, C.: 1998, Fragmented interaction: Establishing mutual orientation in virtual environments, *in* CSC (1998), pp. 217–226. `http://www.acm.org/pubs/articles/proceedings/cscw/289444/p217-hindmarsh/p217-hindmarsh.pdf`.

Huxor, A.: 1998, An Active Worlds interface to BSCW, to enhance chance encounters, *in* D. Snowdon and E. Churchill (eds), *Collaborative Virtual Environments '98*, pp. 87–93.

Isaacs, E. A., Tang, J. C. and Morris, T.: 1996, Piazza: A desktop environment supporting impromptu and planned interactions, *in* Ackerman (1996), pp. 315–324. `http://www.acm.org/pubs/articles/proceedings/cscw/240080/p315-isaacs/p315-isaacs.pdf`.

Kraut, R. E., Fish, R. S., Root, R. W. and Chalfonte, B. L.: 1990, Informal communication in organizations: Form, function and technology, *in* S. Oskamp and S. Spacapan (eds), *People's Reactions to Technology In Factories, Offices, and Aerospace*, The Claremont Symposium on Applied Social Psychology, Sage Publications, pp. 145–199. Reprinted in (Galegher, Kraut and Egido 1990, pp 287–314).

Kraut, R., Egido, C. and Galegher, J.: 1988, Patterns of contact and communication in scientific research collaboration, *Proceedings of the conference on Computer-supported cooperative work*, pp. 1–12. `http://www.acm.org/pubs/contents/proceedings/cscw/62266/`.

Leigh, J., Johnson, A. E., Vasilakis, C. A. and DeFanti, T. A.: 1996, Multi-perspective collaborative design in persistent networked virtual environments, *Proceedings VRAIS '96*. `http://evlweb.eecs.uic.edu/spiff/calvin/calvin.vrais/index.html`.

Lenman, S.: 1999, 3D-, digital environments for social contact in distance work, *Proceedings of Webnet '99 World Conference on the WWW and Internet*.

Mariani, J.: 1998, Visualisation approaches and techniques, *in* Mariani et al. (1998), chapter 1, pp. 9–34. `http://bscw.comp.lancs.ac.uk/pub/english.cgi/d64394/Escape-D31.pdf`.

Mariani, J., Rouncefield, M., O'Brien, J. and Rodden, T. (eds): 1998, *Esprit eSCAPE Project Deliverable D3.1: Visualisation of Structure and Population within Electronic Landscapes*. `http://bscw.comp.lancs.ac.uk/pub/english.cgi/d64394/Escape-D31.pdf`.

Ousterhout, J. K.: 1994, *Tcl and the Tk Toolkit*, Addison-Wesley.

Patterson, J. F., Hill, R. D., Rohall, S. L. and Meeks, W. S.: 1990, Rendezvous: An architecture for synchronous multi-user applications, *Proceedings of the conference on Computer-supported cooperative work*, pp. 317–328. `http://www.acm.org/pubs/articles/proceedings/cscw/99332/p317-patterson/p317-patterson.pdf`.

Şandor, O., Bogdan, C. and Bowers, J.: 1997, Aether: An awareness engine for CSCW, *in* J. A. Hughes, W. Prinz, T. Rodden and K. Schmidt (eds), *Proceedings of the Fifth European Conference on Computer-Supported Cooperative Work*, Kluwer Academic Publishers, pp. 221–236.

Scheifler, R. W. and Gettys, J.: 1986, The X window system, *ACM Transactions on Graphics* **5**(2), 79–109.

Shu, L. and Flowers, W.: 1994, Teledesign: Groupware experiments in three-dimensional computer-aided design, *Collaborative Computing* **1**(1), 1–14.

Smith, G.: 1996, Cooperative virtual environments: lessons from 2D multi-user interfaces, *in* Ackerman (1996), pp. 390–398. `http://www.acm.org/pubs/articles/proceedings/cscw/240080/p390-smith/p390-smith.pdf`.

Smith, G. and O'Brien, J.: 1998, Re-coupling tailored user interfaces, *in* CSC (1998), pp. 237–246. `http://www.acm.org/pubs/articles/proceedings/cscw/289444/p237-smith/p237-smith.pdf`.

Smith, R. B. and Ungar, D.: 1994, A simple and unifying approach to subjective objects, *Technical report*, Sun Microsystems.

Snowdon, D., Greenhalgh, C. and Benford, S.: 1995, What you see is not what I see: Subjectivity in virtual environments, *Framework for Immersive Virtual Environments*, pp. 53–69.

Stefik, M., Bobrow, D. G., Foster, G., Lanning, S. and Tatar, D.: 1987, WYSIWIS revised: Early experiences with multi-user interfaces, *ACM Transactions on Office Information Systems* **5**(2), 147–167. `http://www.acm.org/pubs/articles/journals/tois/1987-5-2/p147-stefik/p147-stefik.pdf`.

Stephenson, N.: 1992, *Snow Crash*, Bantam Books.

Stevens, R. and Papka, M. E.: 1999, ActiveSpaces on the grid: The construction of advanced visualization and interaction environments. To be published in the Proceedings of the PDC Annual Conference 1999.

Turner, J. and Kraut, R. (eds): 1992, *CSCW '92: Sharing perspectives*. `http://www.acm.org/pubs/contents/proceedings/cscw/143457/`.

Ungar, D. and Smith, R.: 1987, Self: The power of simplicity, *OOPSLA '87*, pp. 227–242.

Whittaker, S., Frohlich, D. and Daly-Jones, O.: 1994, Informal workplace communication: What is it like and how might we support it?, *CHI '94: Human Factors in Computing Systems: Celebrating Interdependence*, SIGCHI, pp. 131–137. `http://www.acm.org/pubs/articles/proceedings/chi/191666/p131-whittaker/p131-whittaker.pdf`.