MUDdling Through

Simon M. Kaplan[1,2,3], Geraldine Fitzpatrick[1], Tim Mansfield[2] and William J. Tolone[3]
[1]Department of Computer Science, The University of Queensland
[2]CRC for Distributed Systems Technology, the University of Queensland
[3]Department of Computer Science, University of Illinois at Urbana-Champaign
{simon, ger, timbomb}@dstc.edu.au, tolone@cs.uiuc.edu

ABSTRACT

The wOrlds project is concerned with developing next-generation collaboration frameworks. We strongly believe that real progress in enhancing the usability of collaborative systems hinges on improving our understanding of work, and applying the resulting insights to development of collaborative work support frameworks. We are investigating the thesis that appropriate bases for such an approach can be drawn from existing results in sociology, specifically the work of sociologist Anselm Strauss, and his notion of social worlds. In this paper we motivate and overview wOrlds, the collaborative environment we have built in order to explore our ideas and insights. We then critique wOrlds (and, by implication, the class of systems known as MUDs of which it is a member), and point to future directions for investigation.

1. Introduction

For the past several years we have been investigating collaboration support frameworks that provide support for the informal, cultural aspects of workaday activities as well as the formal, structured aspects of work traditionally associated with workflow systems and other "groupware" tools. A major issue in this investigation is the identification of theoretical approaches or models which can inform systems development and function as "bridges" to the ethnographic investigations of work situations currently popular within the Computer-Supported Co-operative Work (CSCW) community. Recently, we have focused on Anselm Strauss' notion of social world [22] as a theoretical model which might address this issue.

Drawing on our prior experience with developing CSCW systems [14], we have constructed a CSCW support environment called wOrlds (Work Locales and Distributed Social Worlds) [8] to evaluate our theoretical approaches and investigate technologies for support of collaborative work.

In its current form, wOrlds has some similarity to a multi-media Multi-User Domain, or MUD [6]. The multiple *locales* of wOrlds equate to MUD "rooms" in which groups of people can manipulate shared objects and participate in ongoing audio- and video-conferences. These support the informal aspects of work. wOrlds also can be viewed as a media space (see [7], [10]) with sophisticated shared object and navigation facilities due to the pervasive use of audio/video facilities throughout the system. Our initial intention, however, was not to build a MUD (or media space for that matter), but to focus on supporting the activities of social world members through networks of computers.

In this paper, we sketch the origins of the wOrlds project in its predecessor system, *ConversationBuilder*, and briefly critique its failings and our move towards the notion of *social worlds* derived from the work of sociologist Anselm Strauss [22].

We outline our current wOrlds system and the way in which we have interpreted social worlds in the context of computer-based collaboration support. This support is provided by the implementation of *locales* which support the interactions of social world members. We then critique both the implementation of wOrlds and the concepts used to shape its construction, with a view to identifying some significant open problems in the construction of CSCW technologies and to point the way to possible solutions.

In critiquing our experiences with this system, we challenge the interpretation of the spatial metaphor and propose a move away from *space* to *place* and from *boundary* to *centre*, where both place and centre are defined from a social world perspective. We also highlight particular problems and challenges in using existing distributed systems infrastructure to support widely distributed collaborative work. The critique of the existing wOrlds system is used to motivate design directions for its successor, *Orbit*.

Given the obvious overlap between the current implementation of wOrlds and many MUD-based and media-space collaboration systems, we believe this critique, and the theoretical motivations for which we argue, to be relevant to many systems besides our own.

Finally, we end with a discussion of related work and our conclusions.

2. ConversationBuilder background and motivation

This work grows out of our earlier work on flexible workflow support through a system called ConversationBuilder [14]. ConversationBuilder allowed users to describe a wide range of collaborative activities using a protocol specification language loosely based on Speech Acts [27]. The resulting system was used to specify many collaborative activities from asynchronous business processes and software development processes to a complete software development environment.

While ConversationBuilder worked well, it only allowed users to specify the *formal* aspects of collaborative work. Robinson [17] has argued that work takes place on two levels simultaneously - a *formal* level, which emphasizes the manipulation of the artifacts which are an integral part of most work processes, and a *cultural* level, where the informal aspects of work are played out. These levels are inseparable, each shaping and informing the other. Thus, a primary weakness of the ConversationBuilder system was our failure to take account of the double-level aspects of work.

With the wOrlds project we set out to build a system that affords support for work contexts in all their multifaceted richness, both formal and cultural. One of the major issues we had to face was the identification of appropriate theoretical models of work and workgroups. ('Afford' and 'affordance' are terms drawn from Gibson [11], and refer to the facilities an environment provides to facilitate the activities of an animal.)

In ConversationBuilder and other workflow systems, [24], [16], [15] *actions* are the primary building blocks. A workflow system is constructed by specifying the actions that may be performed and (sometimes) the artifacts that are to be manipulated by those actions. Strauss argues convincingly that the actions performed by workers continually vary as workers adapt their activities because of the situated and contingent nature of their work. Thus using actions as a basis for the "codification" of work will be doomed to fail.

Yet, computer programs require that at least some features or assumptions of the system they implement be fixed. We therefore had to identify the key theoretical concepts which we would fix as part of our systems building work. We decided to employ Strauss' concept of *social worlds* as the basis for our investigations. Detailed motivations and overview of Strauss' theories and concepts can be found in [8].

Strauss' social worlds model provides a rich and multifaceted way of understanding the structure and dynamics of groups. In brief, a social world is defined as a group of individuals (or groups), bonded by a common (and possibly implicit) purpose, and bounded by the limits of effective communication. Members of a social world perform actions to accomplish the shared purpose of the world and these actions will continually permute to suit the contingencies of the situation at hand. Social worlds are not necessarily bounded by traditional social or organizational boundaries: their duration is entirely dependent on the task at hand and membership in the social world can range from highly informal and/or transient to highly formal and/or persistent.

Strauss' model of social worlds, we should point out, was developed as an abstract way of making sense of his investigations of workplace situations. It was never designed or intended to be used as a kind of "systems architecture" or "systems requirements" framework. Thus our use of Strauss' ideas is part of an amalgam of technical possibilities, past experiences and searches for models that will break through our blindnesses and, in doing so, point the way to potential new solutions to the problem of computer-supported cooperative work.

So, why use social worlds as a basis for building (or, more appropriately, thinking about building) a CSCW system? In large part the answer is historical: Given our previous work in action-based collaboration support, and given the problems that arose in attempting to support only a part of the continuum of the formal aspects of work (ignoring the informal completely), we were searching for a model that displaced action as the central focus of work, admitted the flexibility and contingency of work, and gave the informal aspects of work equal place with the formal. We believe Strauss' approach does all this, and more. For us, Strauss provides a counterpoint to the obsession with action that pervades a significant part of the CSCW community and points the way to an alternative model.

We do not claim, by any means, that we have yet made full use of his ideas and approach. Indeed we would be the first to agree that our use of Strauss thus far has been more as an inspiration than anything else. But we do believe that there is significant value in this approach and that it can act as a lever to break open intellectual logjams. In Section 5 we explain how insights based on a deeper understanding of the subtleties of social worlds are allowing us to move beyond some of the shortcomings of the current version of wOrlds, and identify directions which potentially can help overcome the limitations of wOrlds and similar systems.

3. The wOrlds solution

Our work on wOrlds has proceeded on two parallel tracks that shape and inform each other: a theoretical track developing the notion of *locales* and a practical track build-

ing our prototype system.

3.1. Locales

We have learned from Strauss that as people work, social worlds continually form and dissolve, as needed. For us the essential question for wOrlds was how we could support *contexts for work* embedded in the computer such that members of social worlds could then use to accomplish their tasks. When groups are working, they need the following:

- The family of artifacts that make up the "formal" layer of their work activities. Examples include program files, medical records, yellow stickies, stripcharts, etc.
- The tools that are used to manipulate these artifacts, such as compilers, editors, debuggers, pens, ECG machines, etc.
- Resources for "effective communication" which grant members of the social world the ability to communicate appropriately to the task at hand.
- Automation of mundane tasks, such as change notifications, where appropriate.
- The ability to navigate, i.e. to seamlessly switch among multiple ongoing tasks, interrelate them as appropriate, and find tasks and people as needed.

The question then becomes: what is it that we can provide through a computer network that allows the construction of contexts of work such that the needs outlined above are met? Our answer is to introduce the notion of *locale*.

A locale for the purposes of wOrlds is a "virtual space" inside the computer system which is intended to be the vehicle for groups to establish shared contexts. Many locales can exist simultaneously, and context switching among them should be relatively easy. Further, each locale should provide the following:

Furnishings. Access to shared objects and tools which can be manipulated by users as necessary. Although the specific objects and tools used to furnish a locale will vary widely from locale to locale, depending on its purpose, every locale will provide audio and video conferencing among all the users "in" the locale at a given time.

Participants. Participants are users who may or may not be present in the locale at any given time, but who have ongoing responsibility, defined by roles, inside the locale.

Visitors. Visitors are users who are in the locale at a given moment, although they may or may not have any particular roles there. Entry to locales can be restricted when required.

Trajectory Schemas or Processes. Processes define the flow of information and control of actions. Processes generally *span* locales, and can be seen as a way both of automating standard aspects of workaday activities where appropriate and useful, and as a way of grouping locales together,

Actions. Actions with user-defined semantics can be invoked at appropriate times, subject to satisfaction of guards and other constraints.

Additionally, locales provide ways of browsing and viewing information, establishing Audio/Video (AV) calls among users and navigating through the collection of locales.

Thus, locales in wOrlds are our current way of "affording" social worlds interactions via the computer, by providing a framework for construction and support of work contexts.

To support the definition of domain specific locales and the representation of trajectory schemas which exist within and across locales, wOrlds provides a specification language called Introspect [26]. We use the term 'trajectory schema' as well as than 'process' or 'workflow' contingent, continually evolving nature of work. Introspect supports run-time modifications to locale definitions and trajectory scheme representations. The environment for constructing and modifying such support is itself a locale within wOrlds.

3.2. The Technological basis for wOrlds

Our goal is to build a system that affords collaboration over the widest possible range of networks and bandwidths, and scales to support tens of thousands of users spread across thousands of locales. Thus a centralized, server-oriented architecture is unlikely to be appropriate. Instead we've been investigating the use of distributed object frameworks as the basis for building wOrlds. Figure 1 gives a schematic overview of our architecture, based on Object Request Broker (ORB) technology [13].

3.3. A Brief wOrlds Tour

Users enter wOrlds by warping into their home locale. *Warping* is the most primitive way within wOrlds of moving from one locale to another. It is similar to barging into a room. Users can think of their home locale as their office. An example of a home locale is presented in Figure 2. The display of a locale is minimally characterized by a locale pane and a tool bar.

The locale pane, which is an object shared by all visitors to a locale (see Figure 2(a)), displays to the user:

- video images of those people who are currently present in the locale. Each locale in wOrlds supports an audio/video conference (provided by standard conferencing tools such as NV and VAT, see Figure 2(c) and (d)) to which users are automatically added and removed as

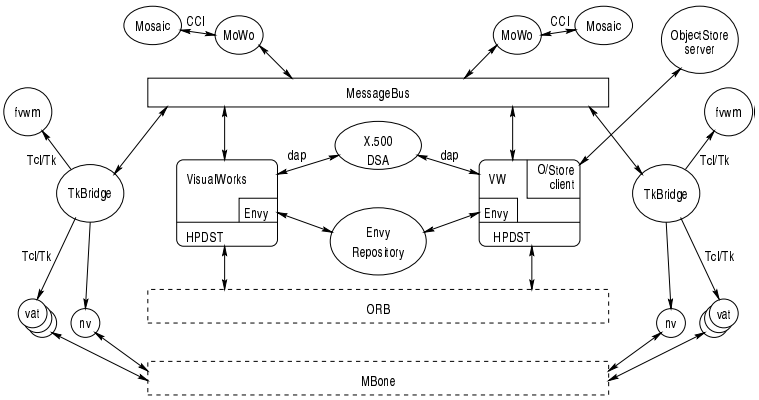
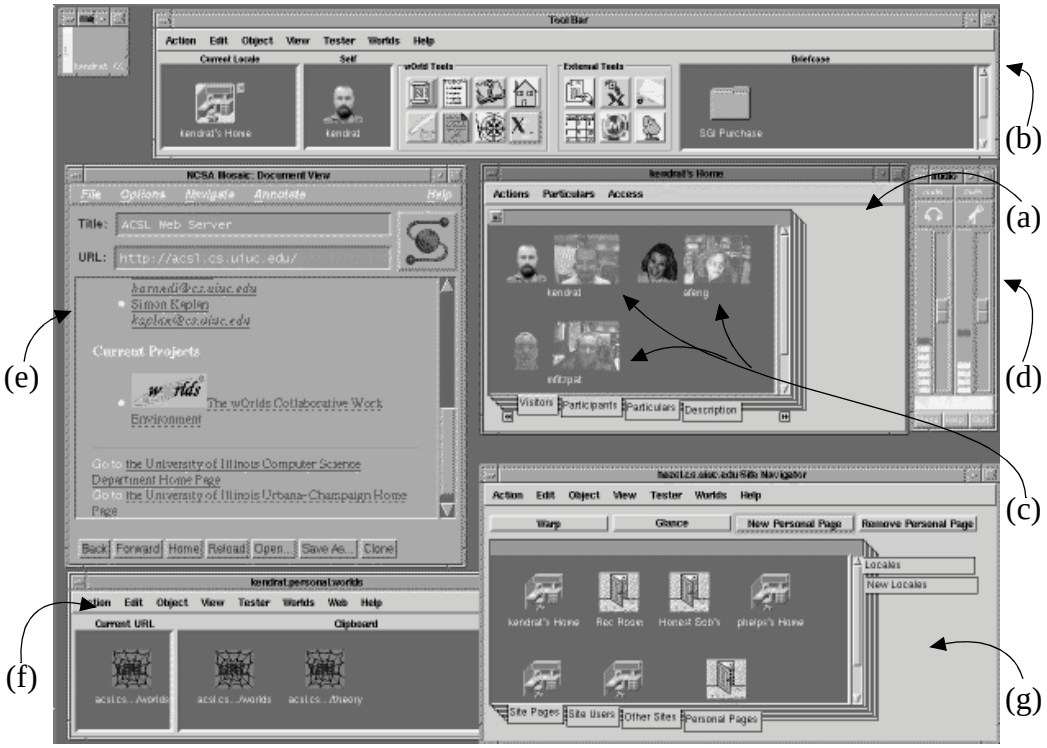


Figure 1. wOrlds architecture



(a) Locale Pane (c) NV (e) Mosaic (g) Site Navigator
(b) Tool Bar (d) VAT (f) Web Tool

Figure 2. Screen-shot of example locale

they enter and leave the locale, respectively.

- the participants in the locale, whether or not they are present.
- the particulars of the locale. This is the family of shared objects relevant to the locale. We distinguish four types of shared objects:
  1. administrative objects necessary to the maintenance of the locale (such as the locale pane objects themselves, or role definitions for participants),
  2. applets, which are small application objects written to furnish the locale (such as an IBIS discussion manager, a shared document annotator or a bug report),
  3. integrated external tools (such as word processors, calendars, or spreadsheets) and
  4. external objects (such as files, URLs, or database objects).
- and, a description which provides the purpose or rationale for the locale's existence.

Each of the above is displayed on a separate page of a notebook widget to conserve screen real estate. Pages of

the notebook may be 'torn off' whenever a user desires to view multiple parts of the locale pane simultaneously.

The tool bar, see Figure 2(b), has four main components:

- the current locale pane which always contains an iconic representation of the user's current locale. Copies of this icon may be created (e.g., via drag-n-drop) and passed around as references to the locale.
- the self pane which always contains an iconic representation of the user. Copies of this icon may be created and passed around as well.
- a collection of tool icons which provide users with a standard set of tools and actions. Examples include: XEmacs, an issue-based discussion applet, a 'warp to home locale' button, a mailer, a network news reader, a Web Tool which opens a web conference using Mosaic 2.5b3 (see Figure 2(e) and (f)), and a navigation tool.
- a drop area called a briefcase which can accommodate any object you want to carry around from locale to locale.

Unlike a locale pane, the toolbar is unique to a particular user, who can extensively tailor both the bar and the bindings of buttons to tools and applets.

As users work in the wOrlds environment, they can move from locale to locale using the wOrlds' Navigator. An instance of this tool can be created at any time by pressing the appropriate button located on the tool bar. The Navigator, see Figure 2(g) has four components.

- Site Pages: The wOrlds universe is partitioned into many sites. Contained within the site pages are those locales registered at the site the user is navigating. From these pages users can warp or glance other locales. *Glancing* is a more polite way of entering a locale. For example, users can glance a locale to see who is currently present, at which time, a temporary audio/video connection is established between the glancer and those present in the locale. Those people present in the locale can then warp the glancer into the locale if they so desire. Our glancing model is closely based on the work of [25].
- Site Users: Contained on this page are icons representing all the users who are registered at the site being navigated. A *call* feature is provided to allow users to establish locale-independent AV conferences.
- Other Sites: This page contains icons representing other sites in the wOrlds universe.
- Personal Pages: Each user has his/her own set of personal pages which contain a 'hot-list' of locales for that user. The locales, represented by icons on these pages, are not necessarily registered at the site being browsed.

The Navigator is not the only means of navigation within wOrlds. Because wOrlds is MIME-compliant, users and locales can be registered with an HTTP server, accessed from the world-wide web or referenced through mail messages, and treated as URLs.

4. Is wOrlds a MUD?

MUDs, or Multi-User Domains, are environments aimed at supporting groups and oriented around a metaphor of a room. MUDs tend to provide multiple rooms, and facilities to move around from room to room. MUDs were initially developed to support multi-user role-playing games (in these MUDs the rooms might represent different parts of a castle or dungeon). Some MUDs have fixed interfaces and furnishings for rooms, while others allow for varying degrees of programmability, allowing for modification of rooms, interconnections, furnishings and behaviours of the virtual artifacts in the system. (such MUDs tend to be called MOOs, after the object-oriented style in which they are programmed). Almost all MUDs have a text-based interface, with scrolling descriptions of rooms and keyboard-based actions.

Certainly to readers who have only ever seen text-based MUDs oriented towards game-playing, wOrlds might not seem very MUD-like. However, there are some ways in which wOrlds is MUD-, or MOO-like. Like many MUDs, wOrlds assumes it is useful to divide the universe into collections of essentially independent "rooms", provides facilities for tailoring these rooms in various ways and provides facilities for navigating among these rooms. Like more advanced MUDs, such as PARC's Jupiter [6], wOrlds is investigating the integration of media space facilities into rooms, such as audio- and video-conferencing.

The issue of whether wOrlds is or is not a MUD is not interesting in and of itself. The fact that wOrlds ended up being MUD-like is an accident of intersecting research trajectories rather than design. But the practical import of this coincidence is that many aspects of our critique of wOrlds, below, are relevant to other CSCW systems, and the research directions we point towards are potentially the basis for, or part of, a manifesto for an entire class of collaboration systems.

5. Critique and future directions

In this section we will critique wOrlds from the perspectives of both theory and technology. Critiques, almost by definition, tend to focus on the negatives but despite the points and issues raised below we consider the wOrlds experiment to have been successful. We have built a useful CSCW system that has been applied in several different domains, we have been able to demonstrate that a non-action-centered approach to CSCW support can indeed

support both formal and cultural aspects of work and we have learned an enormous amount about the subtleties of CSCW support that we simply could never have learned without the hard practical experience of building and working with a system such as wOrlds.

Most importantly, the experience of constructing wOrlds has allowed us to envision its successor system: *Orbit*, in which we hope to answer many of our own criticisms. Where appropriate we will counterpoint our critique of wOrlds with elements of the planned design of Orbit.

5.1. Positive aspects of wOrlds

By their very nature critiques tend to focus on the negative; here we briefly point out some of the many positive features of the wOrlds system.

Multimedia Rooms. wOrlds demonstrated the many benefits of multimedia-based interaction in locales, including the extensive and ubiquitous integration of audio and video.

Shared Objects. wOrlds users tend to make extensive use of the underlying shared-distributed-objects-everywhere framework in the system, which supports the ability to create rich tapestries of objects.

Contextual Action. Users found the ways in which wOrlds, especially the Introspect aspects, supported contextual action a useful way of obtaining guidance and having necessary actions at their fingertips.

Accessing People and Locales. The ability to call people and glance into or warp to locales is a continually-used feature of the system.

Integrating Mail & Web. The seamless integration of email and the web, allowing one to, for example, call someone froma web page or glance a locale from an email message (or vice versa), is a very heavily used feature of the system.

External Object Integration. wOrlds users could integrate external objects (files, etc) with their shared ORB-based objects and make extensive use of this facility.

Navigation Facilities. wOrlds provided a rich navigation tool for finding users and locales which was much relied on by wOrlds users.

5.2. Theoretical issues

One way of thinking about wOrlds, and MUDs more generally, is that they are about setting boundaries around spaces and then populating those spaces. While we have no problem with populating spaces, we believe that the joint focus on setting boundaries and on spaces are misplaced and lead to fundamental problems with collaboration support. The body of this section really critiques wOrlds indirectly, through discussion of several key concepts which are missing from wOrlds and many other MUD-like systems. In many cases we do not have more than the broadest sense of possible solutions to these issues, they are open problems for us at this time.

5.2.1. Idiosyncrasy and intensity. People are players in multiple social worlds at any point in their lives. The combination of personal history and experience, intersecting social world memberships, and personal desires, needs and goals all combine to mean that people tend to have highly idiosyncratic views of the world. While this is well known, collaboration systems tend to take very little account of this, giving only a fixed group view. Thus, our first criticism of wOrlds is that it does not account for the idiosyncrasies of individuals, allowing each user to build his or her individual view of the system, its locales, and the contents of the locales themselves.

Further, while we are all members of different social worlds the *intensity* of that membership is again highly idiosyncratic. In one situation one may be intensely involved, and in another have only a passing interest or be interested only in a particular facet of the activities of members of the social world. Our second criticism of wOrlds, thus, that it does not allow for different individuals to participate in locales at differing degrees of intensity: one is either “in” a locale or one is not.

5.2.2. Spaces and places. Like all MUDs, wOrlds provides a collection of “virtual spaces” in which users “gather” to work together, utilizing whatever resources are available in the space. We believe this emphasis on space is a tactical error on the part of CSCW systems designers. While there are many situations in which modeling the physical in the virtual might have advantages, such as modeling meeting rooms, lectures or conferences, there are many times when trying to simulate the physical in the virtual might be positively debilitating.

The telephone is a fine example of this phenomenon. Telephones do not replace face-to-face communication or provide a perfect substitute for it, instead they provide an alternative means of communication, for which we have developed rich social protocols and which have become a critical part of modern society. But thinking of a telephone as a “space” seems ridiculous and there are many other affordances for communication and collaboration that are not spaces, such as mail, fax machines, mailing lists and radio stations and receivers.

We believe that a shift in focus from the purely spatial to a richer notion of *place* as the basis for supporting collaboration will result in greater power and flexibility in CSCW environments. Places, loosely defined, are collections of affordances with the potential for supporting interaction

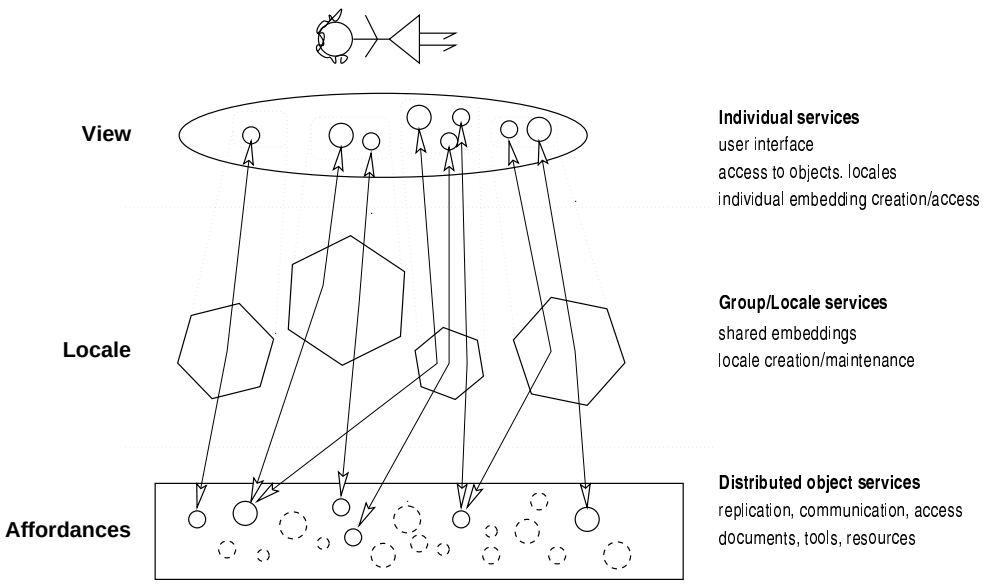


Figure 3. Towards an Orbit Conceptual Architecture

and communication. Thus, places are a superset of spaces, admitting a wider spectrum of possible bases for collaboration.

One immediate practical upshot of such a shift is that the "fundamental thing" we conceive of collaboration systems as providing becomes much more dynamic. In MUDs, for example, "everything is a room" in the end - the fundamental gathering place for collaboration is the MUD room. In wOrlds, it is the locale, and so forth. While graphical representations of rooms will continue to be useful for many situations, in a place-based system users will be able to use any combination of affordances and representations they see fit for the task at hand. Thus one group may choose to have intermittent AV conferences with a whiteboard while another may choose to use a bulletin board and a third could build an entire "situation room", all to deal with the same problem.

5.2.3. Boundaries, permeability and awareness. People employ a multitude of simultaneous foci in their work. wOrlds and other room-based systems provide rigid rooms with strong boundaries. This makes it very difficult to work on "several things at the same time" or keep one's eye on activities in one situation while concentrating on another.

The essential problem here is that while rooms (and locales) have boundaries, social worlds have *centers*. Rigid walls around rooms do not permit users to be sufficiently aware of the activities in other spaces. Our challenge, therefore, is to build systems that allow users to have several simultaneous, personalized foci of differing intensity that support the seamless awareness and permeability among work activities which characterize the real, workaday world.

5.2.4. History and trajectory. Finally, a major weakness with almost every collaboration system we know of is that they do not deal well (if at all) with the issue of providing context through history (what has already happened) and trajectory (what might be a good idea to do next).

To a very limited extent workflow systems *do* provide this in the sense that one can see where one has been (the parts of the workflow that has been completed) and where one might go next (the uncompleted parts of the flow). In [3], Doug Bogia outlines a sophisticated and flexible collaboration model, called Obligations, which provides both for histories and for several different kinds of flexible modifications to workflows. However, no support is provided for capturing the cultural-level history and trajectory of work.

We believe that one of the potential advantages of CSCW support is helping users more quickly to acclimatize to new situations, providing richer notions of history and trajectory is a critical part of this, but at this time we have no ideas beyond Bogia's work as to how to proceed on this. Some possibilities would be to capture the video and audio in a locale, and compress and index this in some way for easy recovery, but the technological problems with storing and indexing vast quantities of (mostly boring) video remain huge.

5.3. Systems implications: abstract view

We do not have, as yet, any clear idea of how to build systems that meet this goal, but the approach we believe should be taken is summarized in Figure 3. In our new system, Orbit, we plan to move from the current one-level

architecture of wOrlds (objects appear in locales, locales are strictly segmented, users can be in only one locale at a time, and all users see the same view of the locale and its objects) to the more sophisticated three-level architecture sketched in Figure 3. The three levels may be summarized as follows:

5.3.1. Distributed object services. At the lowest level, Orbit should provide a collection of services of various kinds that can be used by workers to accomplish their tasks. Examples include shared, distributed objects, AV conferences, resource discovery agents, tools, etc. This level would also contain the infrastructural services, such as object distribution and replication, which form the technological basis for the system.

5.3.2. Group/locale services. In Orbit, locales are the places where social worlds can build shared collections of distributed objects that are germane to the task they are trying to perform, available network bandwidth, etc. A locale should be thought of as a shared "lens" which brings certain of the facilities at the lower level together. This level will also contain the services for editing and maintaining locale definitions and processes.

5.3.3. Individual services. Where the distributed services level allows the definition of particular services, objects, etc., and the group/locale level allows the definition of shared group views over subsets of the information at the affordance level, shared processes, etc., the individual services level allows users to define the ways in which they will view and interact with the many locales in which they participate. Unlike wOrlds, which imposed strict boundaries among locales, in Orbit the user will be able to hold information from multiple locales in focus simultaneously, with differing degrees of intensity, and switch easily from one to another.

Naturally the question of how exactly to build a system like Orbit remains open. We will report on our progress in future papers.

5.4. Systems implications: technological view

The issues raised here are those which have presented the wOrlds implementation team with *fundamental* difficulties or obstacles, as opposed to problems which are, in one way or another, related to particular implementation choices.

5.4.1. Distributed objects. The ability seamlessly to manipulate local and distributed objects is fundamental to the construction of any complex CSCW system. We chose to investigate ORB technology as the basis for the distributed object model in wOrlds. The choice seemed sound - ORBS are built to a standard model, that support the interoperability of systems, languages and machines, and they are the vehicle of choice for many distributed systems researchers and developers. However, we have identified several significant shortcomings with ORB-based distributed object systems which greatly hinder our ability to build a truly scalable collaboration support environment.

Internet instability and ORB reliability. The Internet provides an extremely unstable networking framework, while ORBs are architected on the assumption that interaction among the various servers providing access to distributed objects is reliable and stable. This creates a fundamental mismatch that makes it extremely difficult to deploy ORB technology over anything other than local area networks (or dedicated wide-area networks with considerable bandwidth). Since the Internet is not likely to become any more stable in the near future, ORB services that support reliable access to objects is absolutely essential.

Replication. One way that the reliability-of-access issue can be mitigated is to replicate information in many sites, systems can then concentrate on accessing local data, or have several alternate paths to data, while the underlying replication service takes care of duplicating objects around the network as required. While this sounds very attractive, ORBs support no replication services, and indeed no general replication solutions exist. We believe that the solution to this problem lies not in developing the perfect replication algorithm as every situation could potentially require different replication strategies, but rather to build a replication framework and associated specification environment (rather like Introspect) which allows the development and evolution of many replication strategies.

5.4.2. Adaptability and quality of service. wOrlds should be capable of adapting to changing resource availabilities and different quality-of-service (QoS) demands. wOrlds is built almost exclusively from off-the-shelf components, which provide no facilities for automatically adapting to changing QoS demands, or adapting to changing resources. The entire area of adaptable interfaces which can deal with changes of these kinds remains open, but solutions are critically required in our domain. Obviously any solutions to these problems are going to be closely related to solutions to the problems outlined in Section 5.4.1.

6. Related work

As we pointed out above, the work most similar to our own is the development of MUDs, such as MediaMOO [5]. The major similarity is that MUDs, like wOrlds, reflect the

view that the appropriate role of the computer is to provide a setting where users can interact as freely as possible, albeit with text-based interfaces. MUDs also partition the space of interaction. Rooms in MUDs have similar qualities to wOrlds locales: sharing is encapsulated by the boundaries of the room, rooms typically have a purpose and furnishings. Curtis and Nichols' Jupiter adds video-conferencing to the basic MUD providing comparable affordance for informal communication. A key difference is that MUDs are generally implemented on a central server, whereas wOrlds is distributed.

Virtual reality teleconferencing systems such as MASSIVE [12] and DIVE [1] are obviously similar to wOrlds in supporting partitions of the space of interaction and informal, multi- participant, multi-site conferencing. They focus primarily on enabling mixed-media conferencing between multiple participants using heterogeneous interfaces (either text or graphics) and supporting informal intuitions of interaction in a graphically-rendered, non-video system. These systems provide only extremely limited support for distributed shared objects and integrated shared tools, and make no attempt to model formal work activities.

wOrlds also owes a debt to the development of media spaces specifically at PARC, EuroPARC and Sun, but it extends the basic media space concept with shared objects, locales and many other features. Shared document systems such as GroupDesk [9] and BSCW [2] provide useful abstractions for cooperative work based around documents, DIVA [20] adds a notion of rooms and shared audio-video conferencing.

Although we have focused on Strauss' social worlds model as the basis for our investigations, this work does not exist in isolation. A significant portion of the theoretical work in the CSCW community is, by and large, in fundamental alignment with Strauss' approach, even if the social worlds of the researchers may not overlap. For examples, see [4], [18], [19], [21], [23]).

7. Conclusions

The success of the wOrlds project notwithstanding, for us the real value of this project has been the complex of difficult issues which the project has opened up. On the theoretical front it has exposed us to the subtleties of social worlds as a way of understanding group interactions and raised many issues concerning appropriate support for social world interactions through the computer. Building wOrlds has led to the uncovering of a range of issues concerning distributed systems infrastructure. We believe that the successful future of CSCW systems depends in no small part on finding solutions to these problems.

8. Acknowledgments

This work has been supported by the National Science Foundation under grants CDA-9401124, CCR-9108931 and CCR-9007195, the Defense Advanced Research Projects Agency under grant F30603-94-C-0161, the US Army Corps of Engineers Research Laboratory, Fujitsu Open Systems Solutions Inc., Intel, Bull, Sun Microsystems, Hewlett-Packard, Digital Equipment and the Department of Computer Science of the University of Queensland. The work reported in this paper has also been funded in part by a Co-operative Research Centre Program through the Department of the Prime Minister and Cabinet of Australia. Geraldine Fitzpatrick has received additional support through an Australian Postgraduate Research Award and an Australian Telecommunications and Electronics Research Board scholarship.

The views and conclusions expressed in this document are those of the authors and should not be interpreted as representing the official policies, expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

We would like to acknowledge Andrew Berry, Doug Bogia, Alan Carroll, Donald Cook, Annette Feng, Mark Fitzpatrick, Mark Kendrat, Ted Phelps, Helen Purchase, Chuck Thompson and Yun Yang for their assistance in shaping this project, discussing ideas, and/or working on aspects of the implementation.

Online information about the wOrlds project, systems availability and references to the technology referred to in this paper can be obtained from:

http://www.dstc.edu.au/TU/wOrlds/

9. References

[1] Benford, S. and L. Fahlen (1993). A Spatial Model of Interaction in Large Virtual Environments. Third European Conference on Computer-Supported Cooperative Work (ECSCW 93), Milan, Italy, Kluwer Academic Publishers.

[2] Bentley, R. and P. Dourish (1995). Medium Versus Mechanism: Supporting Collaboration through Customization. Fourth European Conference on Computer-Supported Cooperative Work, Stockholm, Sweden, Kluwer Academic Publishers.

[3] Bogia, D. P. (1995). Obligations: Flexible Support for Dynamic Workflows. Computer Science Department,. Urbana, IL, USA, University of Illinois.

[4] Bowers, J., G. Button, et al. (1995). Workflow from Within and Without: Technology and Cooperative work on the Print Industry Shopfloor. Fourth European Conference on Computer-Supported Cooperative Work (ECSCW '95), Stockholm, Sweden, Kluwer Academic Publishers.

[5] Bruckman, A. and M. Resnick (1993). Virtual Professional Community: Results from the MediaMOO Project. Third International Conference on Cyberspace, Austin, TX.

[6] Curtis, P. and D. A. Nichols (1994). MUDs Grow Up: Social Virtual Reality in the Real World. 1994 IEEE Computer Conference, IEEE Press.

[7] Dourish, P. (1993). Culture and Control in a Media Space. Third European Conference on Computer-Supported Cooperative Work (ECSCW 93), Milan, Italy, Kluwer Academic Publishers.

[8] Fitzpatrick, G., W. J. Tolone, et al. (1995). Work, Locales and Distributed Social Worlds. Fourth European Conference on Computer-Supported Cooperative Work, Stockholm, Sweden, Kluwer Academic Publishers.

[9] Fuchs, L., U. Pankoke-Babatz, et al. (1995). Supporting Cooperative Awareness with Local Event Mechanisms. Fourth European Conference on Computer-Supported Cooperative Work, Stockholm, Sweden, Kluwer Academic Publishers.

[10] Gaver, W. W., G. Smets, et al. (1995). A Virtual Window on Media Space. Conference on Human Factors in Information Systems (CHI '95), Denver, Co, ACM Press.

[11] Gibson, M. (1979). The Ecological Approach to Visual Perception. New York, Houghton-Mifflin.

[12] Greenhalgh, C. and S. Benford (1995). Virtual Reality TeleConferencing: Implementation and Experience. Fourth European Conference on Computer-Supported Cooperative Work, Stockholm, Sweden, Kluwer Academic Publishers.

[13] Group, O. M. (1995). The Common Object Request Broker: Architecture and Specification Revision 2.0, Object Management Group, Inc.

[14] Kaplan, S. M., W. J. Tolone, et al. (1992). Flexible, Active Support for Collaborative Work with ConversationBuilder. ACM Conference on Computer-Supported Cooperative Work (CSCW 92), Toronto, Canada, ACM Press.

[15] Kogan, D. (1993). Design and Implementation of CB Lite. ACM Conference on Organizational Computing Systems (COOCS 93), Milpitas, CA, ACM Press.

[16] Medina-Mora, R., T. Winograd, et al. (1992). The Action Workflow Approach to Workflow Management Technology. ACM Conference on Computer-Supported Cooperative Work (CSCW 92), Toronto, Canada, ACM Press.

[17] Robinson, M. (1991). “Double-Level Languages and Cooperative Working.” AI and Society 5: 34-60.

[18] Robinson, M. (1993). Design for Unanticipated Use. Third European Conference on Computer-Supported Cooperative Work (ECSCW 93), Milan, Italy, Kluwer Academic Publishers.

[19] Schmidt, K. (1994). The Organization of Cooperative Work: Beyond the 'Leviathan' Conception of the Organization of Cooperative Work. ACM Conference on Computer-Supported Cooperative Work (CSCW '94), Chapel Hill, NC, ACM Press.

[20] Sohlenkamp, M. and G. Chwelos (1994). Integrating Communication, Cooperation and Awareness: The DIVA Office Environment. ACM Conference on Computer-Supported Cooperative Work (CSCW '94), Chapel Hill, NC, ACM Press.

[21] Starr, S. L. and K. Ruhleder (1994). Steps Towards and Ecology of Infrastructure: Complex Problems in Design and Access for Large-Scale Collaborative Systems. ACM Conference on Computer-Supported Cooperative Work (CSCW '94), Chapel Hill, NC, ACM Press.

[22] Strauss, A. (1993). Continual Permutations of Action. New York, Adeline De Gruyter.

[23] Suchman, L. (1993). Do Categories Have Politics: The Language/Action Perspective Reconsidered. Third European Conference on Computer-Supported Cooperative Work (ECSCW 93), Milan, Italy, Kluwer Academic Publishers.

[24] Swenson, K. (1993). Visual Support for Reengineering Work Processes. ACM Conference on Organizational Computing Systems (COOCS 93), Milpitas, CA, ACM Press.

[25] Tang, J., E. Isaacs, et al. (1994). Supporting Distributed Groups with a Montage of Lightweight Interactions. ACM Conference on Computer-Supported Cooperative Work (CSCW '94), Chapel Hill, NC, ACM Press.

[26] Tolone, W. J., S. M. Kaplan, et al. (1995). Specifying Dynamic Support for Collaborative Work within wOrlds. ACM Conference on Organizational Computing Systems (COCS '95), Milpitas, CA, ACM Press.

[27] Winograd, T. and F. Flores (1986). Understanding Computers and Cognition. Reading, New York, Addison-Wesley.