



Designing our town: MOOsburg

JOHN M. CARROLL, MARY BETH ROSSON, PHILIP ISENHOUR, CRAIG GANOE, DAN DUNLAP,
JAMES FOGARTY, WENDY SCHAFER AND CHRISTINA VAN METRE

*Center for Human-Computer Interaction and Department of Computer Science,
Virginia Tech, Blacksburg, VA 24061-0106 USA*

MOOsburg is a community-oriented multi-user domain. It was created to enrich the Blacksburg Electronic Village (BEV) by providing real-time, situated interaction and a place-based model for community information. Three versions of MOOsburg have been developed: a classic text-based MOO, a MOO extended to drive a Web-browser, and a Java-based system. The most recent version of MOOsburg is fundamentally different from classic MOOs, supporting distributed system development and management and a direct manipulation approach to navigation. We are currently developing a variety of community-oriented applications, including a virtual science fair and a dispersed natural history museum.

© 2001 Academic Press

1. Vision

Multi-user domains (MUDs) and MUDs object-oriented (MOOs) offer an interesting combination of synchronous and asynchronous communication mechanisms. Users in a MOO can chat with one another, but the database that underlies the MOO is persistent: users can create, modify and manipulate objects, changing the state of the MOO for subsequent users. For example, users can leave messages posted on bulletin board objects. MOOs are fundamentally spatial—the content is organized into “rooms”, and users navigate the information structure with directional commands (for example, “go north”). This evokes an experience of immersion and co-presence with other participants (Benford, Brown, Reynard & Greenhalgh, 1996). These environments are most often used for fantasy-oriented entertainment and informal social activity (Curtis, 1992; Cherny, 1995); they have also been used for professional meetings (Bruckman & Resnick, 1995; Glusman & Prilusky, 1996), as learning environments (Bruckman, 1998; Haynes & Holmevik, 1998) and as general navigation tools (Diebeger, 1996; Kies, Amento, Mellott & Struble, 1996). Our MOOsburg project is studying the creation and use of a MOO in the context of community and home activities.

MOOsburg models the geography of the town of Blacksburg, Virginia; its intended users are the residents of the town and surrounding area. Thus, MOOsburg is not merely spatial; it is *place*-based. It is not fantasy- or recreation-oriented, it is *community*-oriented. The project seeks to enhance community development by supporting better access to local information and to new kinds of collaborations among residents. Our goal is to support construction and end-user programming activities that

will enrich the community's options for cooperation, commitment and concerted action (Kies *et al.*, 1996).

MOOsburg has been developed in the context of the Blacksburg Electronic Village (BEV; <http://www.bev.net>), an advanced community network in southwest Virginia (Carroll & Rosson, 1996). More than 90% of the Blacksburg population, over 30 000 people, have network access. Public-access kiosks are available at the public library and in some business establishments. All of the county's public schools have high-speed network access, and a variety of network-based projects are underway (Koenemann, Carroll, Shaffer, Rosson & Abrams, 1999). Over 150 community groups and more than 400 local businesses maintain Web sites (>75%). There are many unique community-oriented initiatives, such as a senior citizen's on-line nostalgia archive (Carroll, Rosson, Van Metre, Kengeri & Darshani, 1999). The Town of Blacksburg makes extensive use of the BEV, providing on-line forms for surveys, house check requests and email to town officials, as well as on-line town chats and electronic dissemination of schedules and other documents.

Community networking is quite unlike workplace computing. The intended user community for a system like MOOsburg is not a coherent organization; its key organizing rubric is that users' homes are co-located. In terms of activity, communities are organized into pockets of relative coherence, but each pocket of activity is fairly independent of the others.

There is great diversity among people in a community. Unlike a work organization, MOOsburg's intended users span all phases of life (children, teenagers, adults, the elderly), personal roles (parents, siblings, friends) and occupations (teachers, shop owners, police). Groups in a community often share a high-level vision or sense of purpose, but they rarely coordinate their agendas at the operational level. At the same time, many dependencies exist due to shared access to community resources such as buildings or other infrastructure, or to the time and expertise of community participants. It is common for public buildings such as schools or libraries to be used by many community groups with only slightly overlapping interests and goals.

A community network must run on a great variety of platform infrastructures, both hardware and software. A work organization often mandates a standard platform and environment; this is impossible in a community network. It is particularly important that community systems provide acceptable user interfaces and performance on fairly low-end infrastructure, since they will often be used from home.

The motivation and reward structure for community network use is fundamentally different than workplace computing, relying almost exclusively on intrinsic motivation and individual initiative-taking. Most community residents are not paid for participating, thus participation must be its own reward. Innovative development within these systems usually occurs within a single, semi-autonomous pocket of the user community, initially serving local and even idiosyncratic interests and needs. A key concern is the collection and integration of these efforts such that the community at large can benefit.

Our vision of MOOsburg is that it can help to *integrate* the BEV, to provide a more coherent community network experience. Many people visiting the BEV wonder "where" it is, since it is the union of otherwise unrelated community-oriented software

and information. MOOsburg provides a spatial view of these many elements; for example, library services are “at” the library, school activities are “at” the schools. We also hope to make the BEV more *interactive*. The core of the BEV is a distributed hypertext of community information: lots of reading material. MOOsburg emphasizes collective local action in support of shared goals.

In the balance of the paper we give a brief history of the MOOsburg project, then describe its architecture, user interface and example MOO activities. Throughout we consider MOOsburg design and usage issues from the perspective of the special needs of community and home computing settings.

2. Early MOOsburg

The first MOOsburg was created as a course project in Fall 1995 by Kies *et al.* (1996). It was modeled on Jay’s House MOO (<http://jhm.ccs.neu.edu:7043/>), itself derived from LambdaMOO (Curtis, 1992), and the enCORE database (<http://lingua.utdallas.edu/encore/>). The goals of the project were to increase collaboration within the BEV and to present an alternate, spatial view of the community network. At the time, the BEV included email lists, newsgroups and Web pages, but no synchronous communication channels; the user’s view of the BEV was essentially a hierarchical file structure.

In the first few months, MOOsburg attracted several hundred users, and a variety of new collaborative community activities. From the start, programming privileges were available to all users—removing the typical sharp distinction MOOs make between wizards, who can create new objects, and ordinary users, who can merely manipulate objects already created. The MOOsburg designers did, of course, reserve certain extraordinary management powers, such as the power to remove user accounts after persistent misbehavior (“toading”). In fact, MOOsburg had very little such misbehavior.

There was no master plan for the development of MOOsburg. Because no organization “owns” the community network, it grew through distributed initiatives. Early MOOsburg users started up activities that attracted the cooperation and participation of others, inspiring further initiatives. For example, members of the Science Fiction and Fantasy club had a regular meeting on-line in a MOOsburg pub. Residents of several Blacksburg neighborhoods built their own homes in the MOO—so much so that a community steering committee was formed to manage MOOsburg real estate. Users created novel objects and behaviors, such as a MOO Cowbot that roamed the MOO lowing at people, and a self-serve souvlaki machine for the local Greek restaurant. Former residents of Blacksburg logged in from places as far away as Australia to stay in touch. Use of the MOO was not as skewed toward undergraduate males as has typically been reported (e.g. Curtis, 1992): 18% of MOOsburg users were female, and 43% were non-college students.

In Spring of 1997, Craig Struble created a Web interface for MOOsburg, based on the two successful educational MOOs, BioMOO (<http://bioinformatics.weizmann.ac.il/BioMOO/>) and Diversity University (<http://moo.du.org/>), and on the CupOmud Java client for real-time communication (<http://www.du.org/java/CupOmud>). MOOsburg was presented in a framed multi-paneWeb user interface, but retained the normal real-time

functionality of a text-only MOO. This had important consequences for the diverse computing platforms used throughout the community—the support of Java by popular Web browsers meant that residents could use the MOO without installing any special software. Users were able to access existing BEV Web pages, but now via the MOO location to which they refer; they could meet and interact with other users as they visited these places.

MOOsburg's new graphical user interface was elaborated over the next year. An important contribution by Cara Struble was an interactive map that provided spatial orienting information and hyperlink navigation (see Section 4.4). We also began to associate images with locations in the MOO. For example, one could “walk” down Main Street (that is, repeatedly move north or south) and see characteristic views of the center of Blacksburg. We explored the concept of “tours”—a visit to the historical district, illustrated with antique photographs and images; a real estate tour with houses currently for sale pictured and described at corresponding MOO locations.

The graphical MOO was attractive and engaging to a broader range of community groups. The Town wanted to run their bimonthly forum using the MOO instead of a chat. They were particularly interested in the MOO slide projector object that they saw as a convenient display for maps and plans under discussion by committees and residents. The public library worked with us on a book review forum. They were also interested in developing a collaborative story-writing tool for the children's reading room in their MOO site. While such objects could be built in a traditional MOO, residents were more motivated to participate when they were shown the graphical, Web-based MOO (<http://moosburg.cs.vt.edu>).

The MOOsburg users' interest in maps, plans and other graphical objects is interesting, given the low-tech implementation of many community computing projects (e.g. Berkeley Community Memory (Farrington & Pine, 1996), the Cleveland Free Net (Beamish, 1995), Santa Monica PEN (Rogers, Collins-Jarvis & Schmitz, 1994). Simple text-based community networks provide access to useful resources, but history has shown that such access may have the unintentional side effect of directing attention to resources *outside* the community. For example, Big Sky Telegraph supported teachers in rural Montana, linking 1- and 2-room schools with regional libraries, and providing computer support for the literary and artistic projects of Native Americans (Uncapher, 1997). The project was implemented on obsolete computer equipment refurbished by a local women's resource center. It connected a remote and quite dispersed community to the world; for example students now had access via electronic bulletin boards to professors at MIT. Ironically, one of its most visible impacts was to contribute to the regional outflow of talented young people (Rheingold, 1993).

The lesson we drew from these earlier projects is the importance of community network services that encourage interaction with *local* features and issues, and that promote togetherness and collaboration within the community. This has led to a two-sided approach to development of MOOsburg. We view the project as an experiment in supporting diverse community constituencies with a state-of-the-art collaborative environment. But we also have closely involved community groups in the project, so that they can directly represent their activities, their needs and their interests, and so that our infrastructure development is guided by realistic usage scenarios.

3. Moosburg as a community networking infrastructure

Two earlier generations of the MOOsburg architecture were built on traditional MOO server software (Jay's House MOO; <http://jhm.moo.mud.org:7043>). However, the current MOOsburg has a substantially different architecture. Given our analysis of the community and home-usage environments, we knew that it would be critical to make MOO access as simple and universal as possible. We needed to provide a familiar and engaging user interface and to provide community-oriented activities that would provide their own intrinsic motivation for use. Finally, the architecture needed to provide robust mechanisms for arbitrary user-defined extensions that are developed and hosted on servers distributed throughout the community. Our approach in this has been to preserve the most useful aspects of traditional MOOs while extending the client-side user interface and the server-side components to take advantage of emerging internet technologies.

As with traditional text-based MOOs, the MOOsburg software architecture provides access to a database of manipulable objects created by its user community. Some of these objects represent *places* such as rooms and buildings, while others represent *characters* or other *items* that populate the MOO spaces. Also as in traditional MOOs, end-user authoring of objects in the database is a basic feature of the environment.

MOOsburg preserves other fundamental concepts found in traditional MOOs. In particular, synchronous text-based communication between users visiting the same place is pervasive. There is also a seamless integration of synchronous and asynchronous collaborative activities: a user can see the effects of other users' actions in real time, and can also see the preserved effects of past actions.

However, the MOOsburg architecture reflects two significant departures from traditional MOOs. The first is the use of *Java object replication*, instead of streams of text, for communication between client and server. The second is support for well-defined *hierarchies* of places within the spatial database maintained by MOOsburg.

3.1. CLIENT-SERVER COMMUNICATION

Traditional MOOs rely exclusively on text for user interaction. Modern MOO client software, such as CupOmod (<http://www.du.org/java/CupOmod/>), hides arcane textual commands and provides more graphical user interfaces for displaying and interacting with MOO objects. However, these newer interfaces still have an underlying commitment to text that limits the flexibility and sophistication of the interaction techniques. Perhaps the most significant issue is the single "stream" of text used for all input and output. Users move from place to place, communicate with each other and manipulate objects in the MOO by entering commands. Similarly, output describing the content and inhabitants of the current location, incoming communication from other users and system-generated messages describing other users' actions all appear in the same (often quickly scrolling) stream of text. Expert MOO users adapt to this style of interaction, but novice users quickly become overwhelmed. We felt that this command-oriented user interface was a critical usability issue for a community system intended to serve diverse user populations.

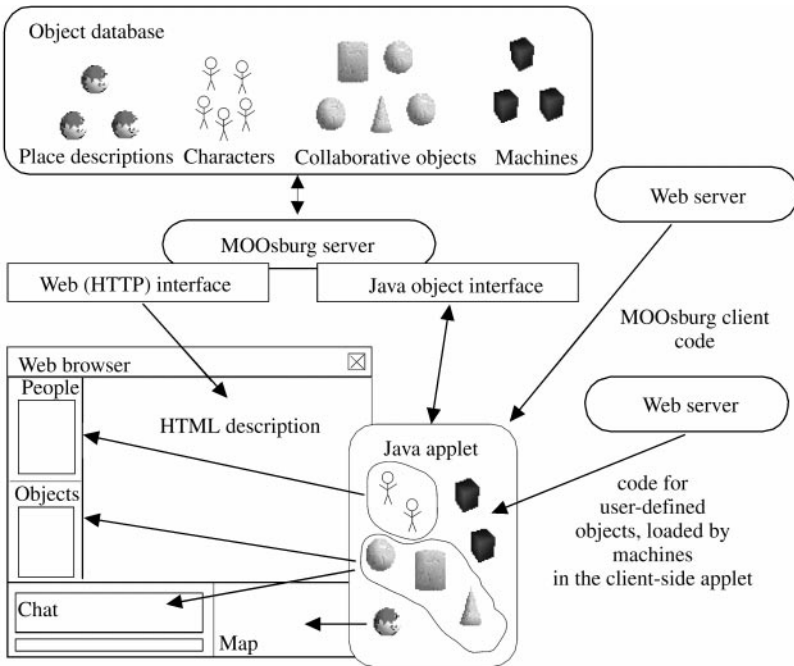


FIGURE 1. The MOOsburg software architecture. Each MOOsburg server provides access to a database of replicated objects, allowing the client browser to retrieve HTML versions of place descriptions and allowing the MOOsburg client applet to retrieve replicas of objects representing characters, places, and interactive items located at each place. Compiled Java code, both for the MOOsburg client and for any user-defined objects, is downloaded from web servers.

MOOsburg employs a very different approach, relying on Java object replication to provide each user's client program with a lightweight, replicated, Java object for each interesting entity (place, person or thing) encountered in the environment (see Figure 1). A generic object replication package called Content Object Replication Kit (CORK; Isenhour, Rosson & Carroll, 2000b) ensures that, where permissions allow, changes made to any replicated object by any user are reflected in all other object replicas.

MOOsburg's object replication scheme has important consequences for the user interface techniques we can support. Because individual replicated objects are independent of each other, different user interface components can be provided for different objects. For example, the users and objects at each location are displayed as a scrolling, selectable, list pane. The current chat conversation is presented in a composite pane that includes the message history, an input field and a send button.

In general, there is a clear distinction between where and what a user is and can do, and how the place and its contents are presented or manipulated. As the user moves around in MOOsburg, the client software retrieves the objects that contain information about places and the objects they contain. User interface widgets then render this information and allow the user to interact with the objects. The general object replication scheme ensures that changes made by any user are reflected in the views seen by other users in the same location.

MOOsburg's object replication scheme also enables an enhanced awareness of people and objects at each location. Users of traditional MOOs request a textual description of the other inhabitants of a place, and receive automatic textual updates as people or objects enter or exit. In contrast, co-located MOOsburg users and objects are always listed and updated. Of course, information about inhabitants and activity at a given location can also be used to create more elaborate visualizations. For example, a map could examine the contents at each location, and add visual details that show the number of people or objects present, or of movement into and out of each location.

A related issue is the object database server. In a traditional MOO, all MOO content exists in a central database. Aside from potential limitations on scale and robustness, this centralized model precludes transparent movement of objects between different MOO spaces on different servers. This is very different from, for example, the World Wide Web, where a page hosted on one server can present images from different servers, and contain hyperlinks to pages on other servers. Because we assume that different community groups and individuals will be developing and maintaining their own "part" of MOOsburg, it is important to support a more distributed model.

In MOOsburg, this is accomplished by giving each MOO object a unique CORK identifier. These object identifiers are conceptually similar to URLs—they specify the address of the server holding the object, and the identity of the object on that server. Thus a room in MOOsburg might contain objects from multiple servers; a subspace could be stored on a different server than its parent space. Initially, the MOOsburg research team will develop and host the top-level space that models the Blacksburg geography, but we expect that community groups or individuals will host buildings or other content on their own servers.

3.2. SPATIAL DATABASE

A second novel feature of MOOsburg is the structure of its spatial database. Traditional MOOs support construction of a *directed graph* of rooms, with very few restrictions. Containment structures (e.g. rooms within a building) may be defined by convention, but are not enforced by the MOO software. A room's position in the space is determined solely by the rooms to which it is *adjacent* [see Haynes & Holmevik (1998) for more information on room construction in a traditional MOO]. Except again by convention, there is no concept of absolute location, and only a crude concept of distance. This ambiguity allows structures to be created within a MOO that could not exist in the physical world. For instance, a user can "go east" and then "go north", and end up in a different location than asking to "go north" and then "go east".

While such "impossible" structures might be experienced as fun or intriguing, they limit the options for graphical navigation. One of the earlier versions of MOOsburg included a JavaScript compass that could generate and send commands like "go north". However, since open-ended MOO structures may not map to a Cartesian coordinate system, it is difficult to provide more of a graphical overview (Shneiderman, 1999), and in particular to reliably render a two-dimensional map of the MOO space.

The current MOOsburg has addressed this issue with a containment hierarchy of locations, and by assigning Cartesian coordinates to each location. In our model, a place that contains other places is a *space*; and any location that a user can navigate to is

a *landmark*. Users may navigate “to” a landmark (i.e. position themselves next to it), or—if the landmark is itself a space with a substructure—they may move “into” the landmark. Streets and buildings are typical spaces; street corners and rooms are typical landmarks within these spaces. A room in a building that contains its own subspace (e.g. a gym with a spatial array of science fair exhibits) acts as both a landmark within the building, and a space containing its own landmarks. This model ensures that a map can be rendered for each space, showing the space’s substructure and the position of landmarks within. The maps visualize the space and enable navigation from landmark to landmark, and into or out of subspaces (Section 4.4).

As with rooms in a traditional MOO, landmarks in MOOsburg can be occupied by users and objects. Where permissions allow, users can create, destroy and move objects from landmark to landmark. Objects, landmarks and users have actions (“verbs” in traditional MOO terminology) that can be invoked on them. For example, most objects understand “take” and “drop”, so that users can move them from place to place. Most objects also respond to “rename”, enabling users to change an object’s name. Landmarks typically support actions such as description editing and subspace creation. Actions for users might include “follow”, “private chat”, “email” and so on.

Actions on users, objects or landmarks can be made available to all users or can be restricted—to an individual or a specified group of users. This allows users to populate a space with objects that are freely available to everyone, or with objects that can only be taken away, destroyed or otherwise manipulated by authorized users. It also gives users who create new spaces control over the addition of new landmarks to their space. For example, members of the MOOsburg team can add arbitrary landmarks to the top-level map, but “public” users can only add markers indicating where they would like to have their own real estate within this model of Blacksburg. Once they create landmarks at the top level, they can create new subspaces, and then can control the objects and landmarks contained within these spaces.

3.3. AUTHORING

A fundamental feature of the MOO paradigm is end-user authoring. Indeed, this was a key attraction of the MOO paradigm for community computing, because it sets up the distributed initiative necessary for the network to grow and thrive (Carroll & Rosson, 1996). Like a traditional MOO, MOOsburg supports several levels of authoring. The simplest is *manipulation of existing objects*. A landmark can contain whiteboards, notebooks or other editable objects. Because these objects are replicated and shared through CORK, they support both synchronous and asynchronous collaboration: changes made to a notebook or whiteboard are visible in real time to others currently viewing or editing the object, and are also preserved for users who encounter the object at a later time.

Users can also *create new spaces, landmarks and objects*. For example, a user might add a new library room to the library building, and put notebooks within it. By default, newly created landmarks contain a text chat object, as well as a toolbox that allows users to instantiate other generally useful objects. Users can also populate landmarks with objects brought from other rooms (i.e. using the “take” and “drop” actions described earlier).

Finally, more advanced users can *implement new kinds of objects*. For example, a book object might be created to present on-line reference materials. The mechanism for building and deploying user-defined MOO objects is similar to creating Java applets: developers implement a new kind of MOO object in Java, and put the compiled object code on the Web. They then create and configure a *machine* object in the MOO; this connects their code to the MOO so that instances of the new kind of object can be created in MOOsburg (see Figure 1). Each machine has a URL for downloading the code, a name for the object that this machine creates and the object's Java class name. The developer maps the methods defined in the object class to actions that will be offered to MOO users. As users create or manipulate these user-defined objects, the code that defines object behavior is downloaded as needed.

The machine mechanism provides a simple means for integrating diverse types of objects into the MOO—any Java class, stored at an arbitrary location, can be instantiated and operated through a machine placed at a MOO location. If these user-defined extensions make use of CORK for object replication, they will also support synchronous and asynchronous collaboration.

To provide a more intermediate level of authoring, a generic scriptable object is being developed that will allow users to define new objects with a scripting language such as JavaScript or Python. This will support the style of end-user programming found in traditional MOOs, where users are able to introduce interesting, active objects using a simple language and tools that are built into the environment.

4. MOOsburg as access to community places and activities

The current user interface to MOOsburg was motivated by two major concerns. First, we wanted a user interface that would be comfortable and engaging to a diverse community audience familiar with modern windowing environments, and in particular, with Web browsers as an interaction paradigm for navigating and interacting with large information spaces. Second, we wanted to address specific problems that we had observed for our early Web-based prototypes.

The earlier prototypes reflected a sometimes awkward combination of a traditional MOO and a graphical user interface to MOO locations, objects and users. For example, the “chat” was simply the standard MOO teletype log. This means that everything that happened in terms of location and object interaction was reflected in the log, and then repeated (sometimes) in the graphical user interface. The result was redundancy and inconsistency in whether and how information was available in the graphical view of the MOO. There were multiple ways to navigate, by typing text commands (e.g. “go north”), using the JavaScript compass or following links out of the Web page for the current location. It was also very easy to “leave” the MOO space without even realizing it, because a location in the MOO could have links to arbitrary Web pages. If a user did this, they were forced to use standard Web browsing interaction (e.g. the “Back” button) to return to the MOO, then pick up again with the MOO-specific commands.

4.1. GENERAL LAYOUT

A prime goal of the MOOsburg user interface is to ensure users' *awareness* of location, other users and opportunities for action. A secondary goal is to promote consistency

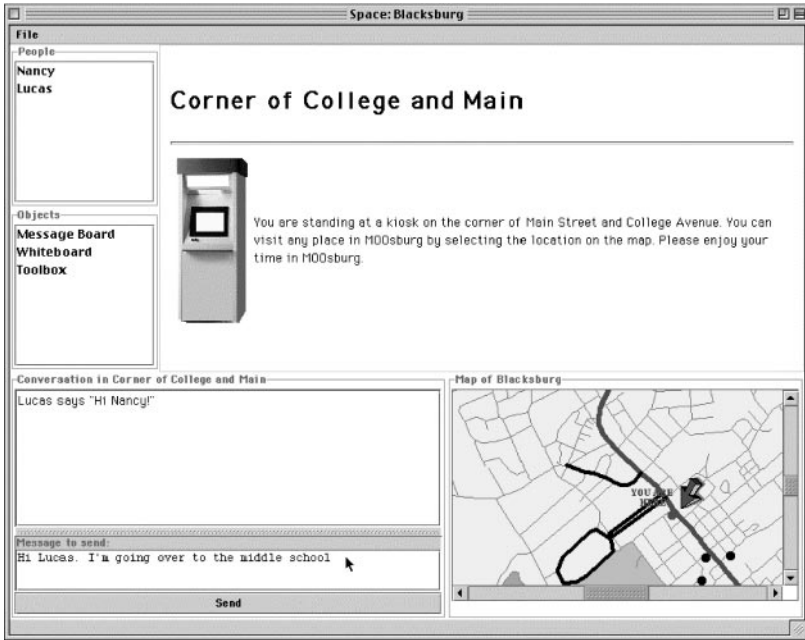


FIGURE 2. MOOsburg: the user (Nancy) has entered at the default location. The location depiction is in the upper right pane, users are listed in upper left, objects for interaction in the middle left, the chat pane in the lower left and the map in the lower right. The open circle in the map signals an embedded subspace (the middle school).

throughout the environment, so that users feel they are moving around and interacting within a coherent space. These two goals motivated the standard view exemplified in Figure 2. In the figure, a user has come to the default MOOsburg landmark, a corner in downtown Blacksburg.

The view in the figure has five basic components: a visual depiction of the user's current location (upper right); a scrollable map showing where the user is located along with other landmarks (lower right); other users currently at the same location (upper left); objects available for interaction (middle left); and a chat pane for sending messages to other users at this location (lower left). (An alternative user interface client for MOOsburg incorporating graphical "avatar" display of people and objects at a location is described in Carroll, Rosson, Isenhour, Van Metre, Schafer & Ganoe, in press.)

In addition to a consistent overall layout, the visual depiction of each landmark includes a visual "signature" that identifies it as part of the MOOsburg database (Mullet & Sano, 1995): the title of the landmark at the top, followed by a bold horizontal line and the room description. The material displayed in this frame is generated and rendered with HTML, using pluggable (server-side) renderer objects. This allows consistently formatted renderings of depictions for all landmarks of a given type (e.g. an entry foyer), but at the same time allows considerable flexibility in the content of any specific depiction. For example, by default the landmark name is shown at the top of the pane,

but the remainder is composed of arbitrary HTML code. Thus, any media type supported by Web browsers can be included.

The map available at each location in the MOO provides a single and direct mechanism for navigation. Users no longer have to know reserved words or room names. They also no longer need to navigate from room to room to get to a destination. The map always available in the bottom right of the interface informs the user of his or her current location, and displays other locations to which he or she can navigate.

Landmarks are indicated as spots on the map. A filled spot means that the user can only go “to” the spot; an open circle means that the user can also go “into” that location (i.e. that there is a subspace and a new map defined for it). To support browsing and feedthrough, we display a location’s name at the bottom of the map when the user positions the mouse over a spot. A single click on a spot will bring the user to that location; the map is updated with an arrow showing “You are here”, and the standard MOOsburg depiction of that landmark is shown in the location description frame (e.g. typically containing information that one would find on a “Welcome page” of a Web site). A double click on an open circle will move the user directly into a subspace; the map is replaced with a map of the new space, and the user is positioned at a pre-defined entry point (e.g. an entry foyer).

The People list indicates who else is present at the current landmark, encouraging informal interaction among visitors. Group chat takes place in the frame at the bottom left of the screen and is similar to standard Internet-based chat tools. Text is typed into the “Message to send” input field and sent explicitly with the “Send” button or implicitly by pressing “Enter” on the keyboard. This eliminates the need for the “Say” command used in conventional MOOs, minimizing the keystrokes needed for informal conversation.

By default, messages sent at a landmark will be seen by anyone else also located there. However, one-to-one chat can be initiated by selecting a user’s name from the People list and double clicking. Again, this is a simplification to standard MOO functionality, wherein private chatting was accomplished through the special “Whisper” command. This point-and-click technique for initiating one-to-one chat increases the feeling of a direct link among users, as well as simplifying the request itself.

4.2. OBJECT INTERACTION AND SHARING

As for chat, interactive objects “stored” at a landmark are manipulated through a point-and-click interface—a pop-up menu that lists all actions available for the selected object. An important design decision was to open any such object (whether a whiteboard, a slide projector, a notebook, etc.), in a separate window. This means that users can open and interact with multiple objects at the same time. Although this increases the complexity of window management, it supports multi-threaded activities, for example simultaneous interaction with a whiteboard (for informal drawing) and a notebook (for editing and archiving text). The decision of whether to automatically close an object’s interactive view when a user leaves a landmark is left to the object’s designer. Again, we have opted for flexibility, allowing for scenarios in which users want to compare and contrast objects associated with different locations.

An important specific consequence of opening objects in separate windows is that there is now a clear distinction between what is “in MOOsburg” vs. what is “on the Web”. Any URLs accessed at a MOOsburg landmark are displayed in a secondary window, a simple Web browser. Recall that these objects are also automatically shared, with the result that users can now navigate the Web together, work together on a shared whiteboard and so on.

4.3. SHARED WHITEBOARD

One generally useful object available through the MOOsburg toolbox is a shared, persistent whiteboard. Our whiteboard supports synchronous and asynchronous image-based communication and collaboration, a functionality that is beyond the capability of traditional text-based MOOs. The whiteboard is an important complement to the pervasive chat, supporting more analog forms of communication (sketching, photos, etc.). Because whiteboard content is persistent, this tool should help to integrate the synchronous and asynchronous collaborative activities that we expect to see in community interactions.

In designing our whiteboard, we have taken an approach that is fundamentally different from other whiteboards built to support informal collaboration. For example, the whiteboard in TeamRooms is a more inclusive spatial workspace; it supports informal communication, but also allows users to import and interact with other applets (Roseman & Greenberg, 1996). Our whiteboard is simpler, a informal communication tool within the overall spatial context provided by MOOsburg. It provides a drawing context that maps naturally into a metaphor of a whiteboard on the wall of a room, a page in a virtual notebook or a portable sketchpad.

A relaxed WYSIWIS interface supports stroke-based drawing over a plain background or any user-specified background image. Freehand strokes and a variety of structured strokes are supported. Users can move or delete any stroke on the whiteboard and can use telepointers to gesture at parts of the drawing surface. A user can change the size of the drawing area, and strokes are repositioned appropriately. Because changes are persistent, any user opening a whiteboard finds it in the same state it was in when last used. Users can export the contents of a whiteboard into a JPEG encoded image file, so that the results of collaboration may be moved into other contexts. A user can also clone a whiteboard, initializing a fully independent copy of the whiteboard's current state.

While the whiteboard will inevitably be used to play tic-tac-toe, it was designed to serve many purposes. For example, community residents discussing an image might load it into the background and then use the whiteboard to gesture and annotate the image. Users might create sketches collaboratively, perhaps deciding how to arrange a MOOsburg subspace, what the interface for a new MOOsburg object should look like, or where to put the playground equipment in a new Blacksburg city park (see Figure 3). It could even function as a graffiti wall, with visitors to a MOOsburg location stopping to make personal additions to community graffiti. The design of the whiteboard does not limit it to any particular usage.

As a ubiquitous tool in MOOsburg, the whiteboard will support the integration of the BEV and MOOsburg. Regardless of their location in MOOsburg, users will have access to a whiteboard to support their activities. Consider, for example, two users in the

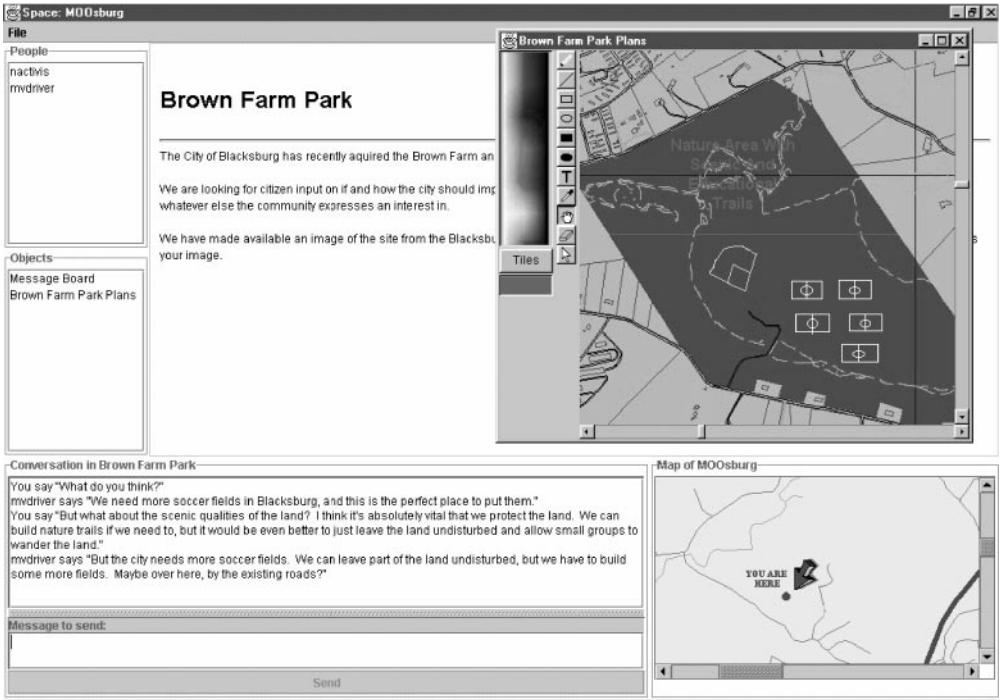


FIGURE 3. A shared whiteboard (the smaller window on top of the MOOsburg frame) is being used to plan construction of a new city park. A map of the park has been loaded in as background and users are using chat and annotation to discuss their plans. Note that the whiteboard has been renamed to better indicate its persistent function.

MOOsburg location for Burruss Hall on the Virginia Tech campus. Using the Virginia Tech website, they have each found a campus map and want to discuss it. They can open a whiteboard, load the map image as whiteboard background, and then discuss and annotate it. We hope that interactions like this will promote development of a stronger sense of community in the BEV, as users can interact in ways not supported by standard Web pages.

Because collaborative opportunities like the one we describe will be common in MOOsburg, it is important that users are always able to access a whiteboard and that support for image-based collaboration is consistent throughout MOOsburg. To this end, whiteboards are available through the toolbox in every MOOsburg location. Developers may also create new kinds of whiteboards. For example, one user group has created a collaborative easel—the same drawing surface, but users can flip forward or backward to get to other pages.

Several improvements to the whiteboard are in progress. Strokes are now only manipulated individually, but we are extending this to offer simple grouping functionality. Improvements in the support of images on the whiteboard are being pursued. An advanced undo and redo system is also being explored. Improved support for user awareness is under investigation. Possible additions include a radar pane, improved

telepointer identification, and the ability to playback the history of changes on a whiteboard. Finally, we are exploring mechanisms that will promote extensibility of the tool. Possible options include providing a set of methods that can be defined to change the behavior of the whiteboard. For example, we might allow designers to design a collaborative chess game by imposing an 8×8 grid onto the workspace, and snapping object movements to be in alignment.

4.4. MAP-BASED NAVIGATION

A signature of a traditional MOO is compass-based navigation. Users enter directional commands to move step by step through a connected graph. This “maze metaphor” can be fun when the overarching task for users is exploration, but it is not consistent with more goal-directed tasks, such as “going to the Middle School for a virtual Parent-Teacher meeting”. The whole point of using the virtual space is to avoid walking down to the school! Thus, we support direct access to locations via a map. A secondary consequence is that residents will interact with a familiar layout of their town, reinforcing the sense of shared community.

The map in MOOsburg is used to render the database’s containment hierarchy. At the top level is a street map of the town of Blacksburg (see Figure 1). It indicates landmarks at this highest level—for example, buildings or street corners. Subsequent levels are defined by the subspaces created for landmarks. Typically, a building landmark presents its subspace as a floorplan; and if the building has multiple floors, stair widgets provide access to different floors.

An important issue for map-based navigation, especially for large spaces, is screen real estate. We are committed to the consistent layout that positions the map in a small pane in the lower right. Thus, we are experimenting with fisheye maps that can be used to view and navigate large geographic spaces (Furnas, 1986). Fisheye views use focus + context techniques to present the details needed for local interaction, but also include a compressed view of the overall structure. In MOOsburg, we display a user’s current location as the focus, with the rest of the town (or other subspace) as context.

We have developed a prototype that uses different mathematical projections to provide a focus + context map view. Figure 4 shows the Arc Tan and the Parabolic $x^{1.5}$ projections, which produced the smoothest and most interpretable views. The maps show an area of about nine square miles and indicate the roads that exist within that area (based on map data acquired from the Town of Blacksburg). A major landmark in town, the Virginia Tech Drillfield, is at the center of the map and is the focus. The Drillfield and roads nearby appear much as they would in a standard flat-view map. Roads more distant provide the context, conveying the overall layout of the town. A single mouse click changes the focus of the map; a change in focus takes place through smooth animation.

The slider widget provides direct control of zooming with smooth display updating, under various zoom options. The zoom supports a focus + context view in that one can easily zoom out to see more context and zoom back in without changing the focus. In contrast, a standard flat-view map might require many scrollbar adjustments to obtain similar context information. The prototype uses two layers of data—one for roads and another for buildings. This allows the buildings to be rendered independently at finer

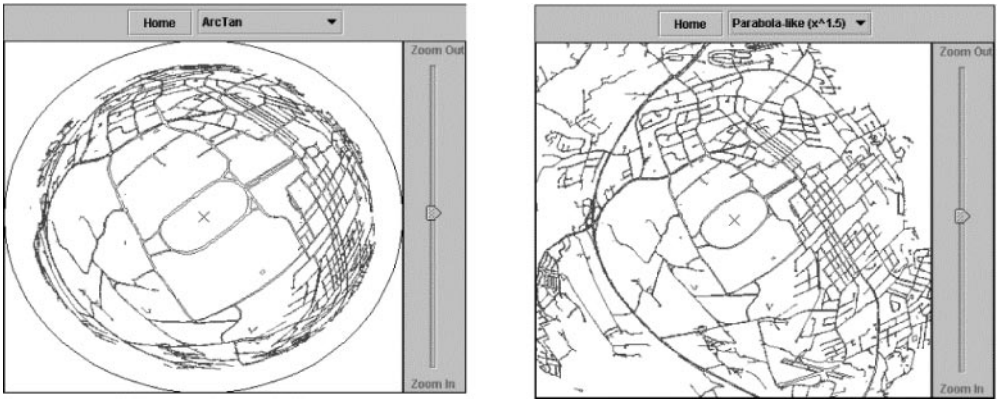


FIGURE 4. An experimental map interface to MOOsburg showing the Arc Tan and Parabolic projections used to create a fisheye visualization of the town streets.

zoom levels. Layering data provides many new possibilities for map visualizations. For example, layers could be interactive, allowing users to turn on and off different types of information

We have done some exploratory user evaluation of navigation using the prototype and paper maps. Many of the users had initial difficulties with the focus + context views, making comments such as “I’m buffaloed” and “This is really distorted.” However, we found that users were about equally able to navigate with a focus + context view and with a zoomable flat view, that is, a standard map enhanced to support zooming to see context. Zooming appeared to be critical: users were more successful navigating with a focus + context view than with a scroll-only flat-view map. We also found that major roads and frequently visited places were the most important landmarks for map navigation. For example, users were aware of and able to use the location of major grocery stores in town.

The map prototype has also helped to raise further depiction issues and possibilities. Buildings are currently rendered in a single color, with their size and shape relative to the physical world. But in the physical world a building may contain multiple independent locations that should be displayed separately. For example, both a restaurant and the store located above it should be displayed and accessible in a community map. The map could also be used to visualize activity in MOOsburg. For example, we could indicate where other people are or where they have recently been, and possibly what they are doing, on the highest level map. Such a graphical people-finder might facilitate opportunistic interaction among users.

5. Growing MOOsburg

The MOOsburg effort faces two immediate challenges. First, we need to make MOOsburg salient and interesting to the community. We need to differentiate MOOsburg from the static HTML pages that are currently still standard in the BEV; we need to convey the utility of the new infrastructure; and we need to stimulate people’s imaginations with

respect to new applications. Second, we need to provide models and tools to ease the process of developing MOOsburg sites. The BEV became an effective community networking infrastructure, not because a few pages of links were created by the BEV management group, but because many groups and individuals throughout the community were motivated and able to create and post their own information (Carroll & Rosson, 1996).

Our current strategy for addressing these challenges has three facets. First, we are directly facilitating high-visibility community events. During the past year, we have focused on a Virtual Science Fair application, a MOO implementation of the traditional spring exhibition of student science projects occurring in high school gymnasiums throughout North America. Our goal is to produce a major community event, raising awareness of the project, and giving a wide variety of community members some basic experience with MOOsburg.

Second, we are working with community groups to develop model sites. In these collaborations, we play relatively more of a support role, helping groups to exploit the MOOsburg infrastructure in order to realize their own goals. In some cases, these example activities are merely imported; for example, a Web-forum for community history developed with local senior citizens is an object in the Senior's Center (Carroll *et al.*, 1999). But we are also exploring new MOOsburg tools through each of these collaborations. These sites will provide a collection of models that other community members could use in developing their own sites in the MOO.

Third, we are assessing and refining support for programming extensions to MOOsburg: to enable creative use of MOOsburg, we need to facilitate construction of new kinds of sites and MOO objects by end-users. Making programming accessible to end-users is a long-standing challenge in human-computer interaction (Nardi, 1993); MOOsburg brings this challenge right into the home, as well as raising the general problems of distributed software development, deployment and maintenance.

Of course, these three facets of MOOsburg evolution are not independent. For example, investigating new tools and techniques for software extensions to MOOsburg is typically coordinated with work on model sites. And all three lines of work are helping to guide the further development of the MOOsburg infrastructure. New subspaces add new requirements for navigation that must be integrated with the ongoing research on map-based interaction. New types of objects add requirements for the kinds of behaviors that must be mapped into the MOOsburg point-and-click interface.

5.1. INCREASING MOOSBURG'S VISIBILITY

One way to introduce MOOsburg to the town is through carefully staged "events" that will take place in the MOO. We have identified a Virtual Science Fair (VSF) as one such event, partly because it builds on related work, aimed at supporting collaborative projects in K-12 science education (Koenemann *et al.*, 1999; Isenhour *et al.*, 2000a). Science projects are diverse and can be created by a wide range of student participants; there is a history of mentoring activities in the community (Gibson, Neale, Carroll & Van Metre, 1999). We are leveraging the growing interest in community involvement in science education, building a virtual analogue of current science fairs that removes some of the time and location constraints of physical fairs.

The VSF is being developed as part of Christina Van Metre's dissertation research. Science fairs typically exhibit projects that students have developed outside of class, over a period of weeks or months. The exhibition usually occurs at the school during a single evening toward the end of the school year. We believe that this traditional activity could be enhanced if the projects involved community members more actively, if the exhibition of the projects could be carried out over a longer period of time, and more conveniently accessed by community members. In our VSF concept, students use the Virtual School to collaborate with one another and with community members to carry out projects, which they exhibit at the VSF. The exhibit process itself becomes an extended collaborative activity, since the virtual fair is not constrained to be a one-time event.

Prior to prototyping the VSF, we carried out brainstorming sessions with community esidents, school teachers and high school students, who helped us to develop requirements for the browsing, mentoring and judging of hypothetical science projects. Ideas generated included calendars, automated email reminders, message boards, judging templates, annotation capabilities and subscribable interest lists. It was important to these potential users that they be able to use the VSF without downloading additional software; students and community members wanted to work together from home during after-school hours, parents wanted to check in on the progress of their children's activities. These users also envisioned the VSF as an archive of projects that might inspire ideas for new projects, or provide other information related to project interests.

The initial version of the VSF in MOOsburg uses a gymnasium floorplan as the orienting spatial metaphor (see Figure 5). The fair is accessed via the Blacksburg Middle School, and includes the default chat, message board and toolbox objects. It also includes a calendar object that visitors can use to find out when students will be present to discuss their projects. The exhibits are accessed by clicking on their representations on the floorplan. They open within individual Web browsers on a set of multimedia pages authored by the students. Visitors can open as many projects as they wish, engage other visitors in discussion about the projects or leave messages for subsequent visitors or for students.

Our work on the VSF has also focused on how the content will be developed in the Virtual School (Koenemann *et al.*, 1999; Isenhour, Carroll, Neale, Rosson & Dunlap, 2000a). The Virtual School integrates text chat, audio and video conferencing, email, Web forums, a shared whiteboard and a collaborative notebook. It hides low-level details of connecting conference participants, for example, providing single-click launching of video conferencing sessions. It coordinates synchronous and asynchronous interactions through automatic persistence to ensure that chat or whiteboard contents are retained when a conference ends. It supports collaborative awareness through buddy lists and a notice board of important system events. Student activities in the Virtual School include planning research projects, gathering resources (images, URLs, text), running simulation experiments, sharing experiments and data and writing reports.

Contributions to the VSF will occur as students publish their work. These "public views" of students' projects will be automatically indexed, appearing in the appropriate section of the VSF (e.g. as part of the biology board at the middle school fair). At the same time, students can establish links from other locations in the MOO. For example,

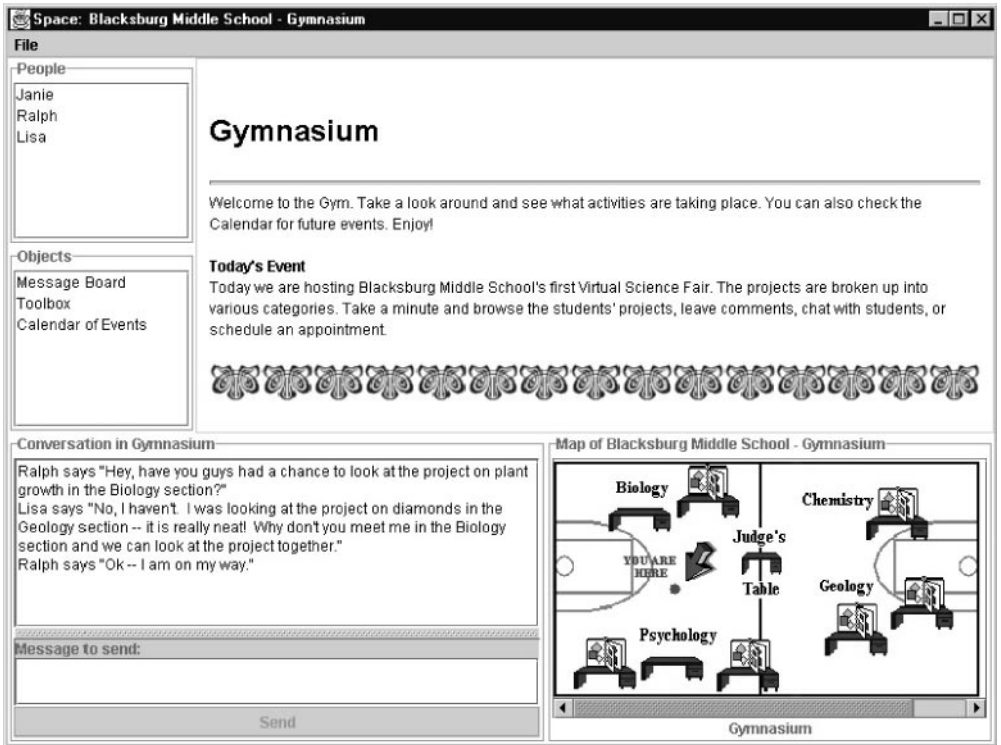


FIGURE 5. An invitation to a Virtual Science Fair, introduced (as Today's Event) upon entering the Blacksburg Middle School gymnasium.

a high school student conducting a survey on a proposed zoning change might drop a copy of her report in a MOOsburg boardroom set up to discuss the issue. Finally, MOOsburg users visiting the VSF will be able to add their own comments or reactions to exhibits.

By providing these types of capabilities, we hope to foster an environment in which informal mentoring and community involvement can take place. We believe that MOOsburg addresses many of the constraints that plague these interactions in the real world—time constraints, geographic location, convenience of interactions, visibility of results and effective evaluation of project focus and structure (Gibson *et al.*, 1999). Mentoring is a process that is reciprocally beneficial to both students and mentors. The more focused use of mentors in Virtual School projects have been very successful in the past. We hope to carry over these types of positive activities and relationships and to encourage more community involvement.

The main elements of the VSF are provided by the MOOsburg infrastructure, for example, the concept of subspaces and maps, and the shared Web browser. Any teacher who has asked her students to construct Web-based project reports could use the system we have described here, to promote sharing and interaction with the community. Full integration of the Virtual School with the VSF will require a further linking step, so that

Web material created in the former can be automatically “wrapped” and indexed as a MOO object.

5.2. PARTICIPATORY DEVELOPMENT OF MODEL SITES

MOOsburg application development is fundamentally community- and end-user oriented. Ultimately, this development must be self-sustaining, but at the moment, we are bootstrapping MOOsburg development with participatory design methods (Greenbaum & Kyng, 1991; Chin, Rosson & Carroll, 1997). We have solicited input from a wide variety of groups, and are working closely with groups who have expressed the greatest interest and ideas. The resulting user community is very different from business or workgroup populations. While some of these groups maintain fairly consistent and visible membership, others are made up of quite transient and nebulous associations.

The diversity of the user communities and the transient character of many of their constituencies demands diverse but focused applications and environments. For example, Dan Dunlap has been working with a local natural history museum to build virtual museum services in MOOsburg. The intent is to serve all possible user groups: very basic qualitative and visual information for elementary school students, but also detailed technical information for college students and researchers. While both groups may use taxonomic indices to access data, they are likely to need rather different formats and interaction options. Because users will use such an application for short intervals, it is also critical that the user interfaces leverage knowledge of common user interaction styles.

In the museum project, a major goal has been to use MOOsburg activities as an enhancement to the “real” museum. This design goal, coupled with a desire to not merely copy what is already available on the web, has led to a new paradigm for interacting with museum specimens—MOOsburg will support the co-discovery and interaction of museum visitors who have shared interest in museum exhibits, most of which can be physically touched at the museum. Rather than confining interactions with museum specimens to a “virtual museum”, we are designing a specimen database that can be carried around in the MOO, and can be used to “plant” specimen instances in different spaces, along with tools that help visitors find and join interest-based groups. Thus, what was originally an on-line database confined to one network location will become an object used to populate the MOO with things that may help users find one another and collaborate on science investigations.

Other model sites are being developed in collaboration with the BEV Seniors and the League of Women Voters. The members of these community organizations are more stable than the visitors to a museum, but still have specialized needs. Moreover, such groups do not combine into a coherent whole except through their shared pursuit of specific goals and interests. We are working with the groups to develop subspaces and tools that fit the groups’ special needs. For instance, while the verbal and mental abilities of seniors are mature and sophisticated, their physical abilities and awareness can be limited. Meeting and forum rooms designed by and for seniors can include tools specific to these needs, for example, a virtual magnifying lens to enlarge text and simplify navigation.

One of MOOsburg's key goals is to integrate community information and activities within a single shared infrastructure. Thus, we have assumed from the start that groups will seek to integrate existing systems that are already a part of their practice. The distributed architecture described earlier was designed explicitly to support such efforts. A good example is the museum specimen database described earlier. Another is the ongoing Save Our Streams project, which collects and archives water quality data from physically distributed stream sites. We are working with the organizers of this project to create monitoring stations in MOOsburg, where participants at one site record, edit and analyse data from their stream, while also interacting with participants at other stream sites.

The diversity of the user communities implies diverse goals and motivations for the users of MOOsburg. However, unlike business use contexts, the motivation for using MOOsburg must be largely intrinsic. There will be no external salary or reward system to facilitate use. Rather, we are leveraging local innovations, interests and existing programs and resources to drive MOO development and use. Students may wish to add water quality data to stream monitoring stations in the MOO. Members of the League of Women Voters may wish to create, organize and conduct on-line meetings or forums with political candidates. Teachers may wish to create interactive displays for student work. The motivational strategy is to blur the distinction between users and developers by providing user communities the ability to create, modify, equip and populate virtual spaces for their unique purposes and needs.

5.3. EXTENSIONS TO THE MOOSBURG INFRASTRUCTURE

A crucial enabler for the ongoing development of MOOsburg will be end-user programming of new sorts of MOO content. Ultimately, community groups need to be able to create and manage their own specialized MOOsburg places and objects. This empowers them with respect to using technology to achieve their own goals. More importantly, it makes MOOsburg feasible: it is difficult to imagine how any community could fund a central development team for a community MOO. But this in turn demands that we provide support for end-user programming activities and for integrating independent development efforts.

We have begun to explore support for more advanced development in MOOsburg through a set of course projects in a graduate course in computer-supported cooperative work at Virginia Tech (<http://courses.cs.vt.edu/~cs5734>). Six MOOsburg activities were prototyped: an event publicity system, a software engineering room, a Virginia Tech Off-Campus Housing office and fair, advertisement and connectivity for *ad hoc* sports activities, and a children's collaborative story writing at the public library (see Table 1).

Development of a sample collaborative activity for MOOsburg was mandatory for this graduate course; problem domain was open but the project had two required elements. First, it was necessary for project groups to identify appropriate stakeholders for their selected problem context, and to perform requirements analysis and cooperative design activities with these stakeholders. Secondly, each group was required to create at least one new type of object within the MOO (i.e. a new Java class deployed via the machine mechanism).

TABLE 1

Six MOOsburg activities developed as course projects in a graduate class

Project	Description	Features
Event planner	Support for scheduling and advertising events in MOOsburg	Bulletin boards to post events Event flyers that can be taken and carried by the user Support to teleport to the location of an event or the location of its' coordinator
Housing fair	Virtual version of Virginia Tech Off Campus Housing Fair	Booths for realty offices with Web page link objects Q&A board maintained by realtors
Sports calendar	Schedule <i>ad hoc</i> sports activities	HTML content and discussion area for each sports activity location Java calendar tool allows you to list sports events and sign up for events
Story writing	Childrens' collaborative story writing activity	Open-ended picture book Each page includes a user-selected image, writing area and discussion area Authors can publish story as HTML
Real-time auction	On-line synchronous and asynchronous auctions	HTML content for browsing and submission of auction items Collaborative object for synchronous bidding activities
Software engineering Room	A room supporting software development in MOOsburg	Link objects to MOOsburg developer web pages Link object to BSCW server for threaded discussion and document sharing Wrapper object for JCVS to access MOOsburg source files in CVS Extended whiteboard to multi-page

Developers taking on these more ambitious levels of authoring in MOOsburg have many options for extending the generic infrastructure. Table 2 summarizes the range of options explored by the six projects. MOOsburg directly supports the adding of new spaces and objects to the MOO. These student teams were all familiar with HTML, making it easy for them to add rich content that enhanced the spaces that they developed. The distributed publishing and machine mechanism supported the development of simple custom objects, and the CORK replication toolkit simplified the creation of new object types that supported collaboration as well.

The virtual housing fair provides an example of how a useful community activity can quickly be built. Modeling after the real-life Virginia Tech Off Campus Housing Fair, MOOsburg was used to construct a room with separate booth spaces for realtors that will attend. Each booth is populated with its own message board and slide viewer (tools

TABLE 2

End-user programming features explored by the six course projects

	Event planner	Housing fair	Sports calendar	Story writing	Real-time auction	Software room
GUI for creating spaces	X	X	X	X	X	X
Non-interactive HTML content		X	X		X	X
Interactive HTML content			X		X	X
Custom Java object(s) [†]	X	X	X	X	X	X
Object replication scheme (CORK)	X	X		X		X
Reuse of existing MOOsburg objects				X		X
Reuse of existing external software						X

[†] This was a project requirement.

already existing in MOOsburg) for use by the realtors. Additionally, this project group developed a customized, moderated question and answer board object that was placed in each booth; this object now can be reused by other MOOsburg site developers.

The software engineering room offers three examples of how reuse of existing code can bring new resources into MOOsburg. Taking advantage of accessibility to HTML, this project team built an object that created an external link with a BSCW server (an existing Web-based collaborative system; Bentley, Appelt, Busbach, Hinrichs, Kerr, Sikkell, Trevor & Woetzel, 1997). Utilizing JCVS, an existing Java class library providing a graphical front-end to the Concurrent Versions System (CVS), the group created another object that would allow developers to access source files within CVS. Finally, this team extended the collaborative whiteboard, creating a subclass that supports multiple pages. By taking advantage of reuse, this project group rapidly composed a useful set of tools that supported collaborative software development.

The story writing team produced the most complex example of a new collaborative activity. This group's StoryBook object allows children to collaborate synchronously, to write new stories. The pages in the book contain a user-selectable picture and an associated text area. A persistent chat log for each StoryBook is provided for children to discuss the development of the book's content. Awareness mechanisms are provided so a child can see what other children are currently working on in the book, as well as who has contributed to the content thus far.

The MOOsburg architecture greatly simplified the development and integration of StoryBook. New books are created through a “machine” object in the MOO. These book objects contain references to two shared list objects that track the picture on each page and each page’s text content; these shared objects are replicated and shared via the CORK toolkit (Isenhour *et al.*, 2000b). Two other shared lists track current authors and past contributors to the book. This second type of list represents a new replicated data structure added to MOOsburg during the development of StoryBook. The chat discussion object, which already existed in MOOsburg, was extended to provide a custom interface supporting the children’s discussion.

The student projects exposed important issues concerning user-constructed MOOsburg extensions. The machine mechanism worked well as a technique for creating and operating new kinds of objects. However it led to rooms that listed two new objects, a “machine” that can be used to make new instances of a new class, and one or more instances of the class itself. This is somewhat disconcerting for end-users of the new space, and it led us to wonder whether MOOsburg should have a different interface for room developers than for room users. We were also struck by the variety in the user interfaces developed across the six projects. Although the initial landmark depictions reflected a consistent graphical design (landmark name with a bold horizontal bar, followed by an HTML description page), the custom objects that the teams created varied a great deal in look and feel. It is not clear that this is a problem; indeed given the diversity of the MOOsburg user population, it may be a necessity. However, we will be interested to see if a “MOOsburg interface genre” develops over time (Erickson, 2000).

These student projects represent just a small sample of the diverse software applications that MOOsburg can support. The event publicity system and software engineering room developed general, reusable objects that are now available to all MOOsburg users. The Virginia Tech Off-Campus Housing office and fair, along with the children’s collaborative story writing at the Blacksburg Public Library, provide virtual counterparts to current real-life activities that take place in the Blacksburg community. Allowing community members to plan local *ad hoc* sports activities or hold a real-time auction on-line, represent collaborative activities that are only possible in a virtual environment like MOOsburg.

We are very interested in the enhancements to the MOOsburg infrastructure that will enable MOO users with diverse needs, like the student groups, to contribute to the system. In this sense, the student projects are an important source of requirements for end-user authoring support. Some of the key areas we are now exploring include distributed software development and installation; simplified programming of collaboration features; security requirements; and increased options for multi-user awareness.

6. Conclusions

Community computing seeks to enhance participation in community life at a time in history when traditional communities appear to be eroding (Bellah, Madsen, Sullivan, Swindler, Tipton, 1985; Putnam, 1996). It investigates and facilitates the development of local *social capital*—the trust, social interactions and norms of mutual reciprocity

throughout a community (Coleman, 1990). The MOOsburg project is attempting to bring new technological approaches and applications to community computing.

This is not to impugn the pervasive and exciting vision of what McLuhan (1964) anticipated as the “global village”. We all participate in a global village of international collaboration, electronic commerce, telework and so forth, mediated by the Internet. But we live in physical communities, next door to someone, down the street from someone else. We believe that a critical complementary element, one often missing in contemporary visions of information technology and society, is an appreciation of the importance of local commitment, initiative and cooperation (Carroll and Rosson, 1999). To achieve a global village we must incorporate, codify and celebrate the unique nature of every region, town and neighborhood. Environments like MOOsburg may help networking technology play a more proactive role in allowing people to come to know one another throughout the global village.

Community computing raises a variety of distinctive and difficult challenges. Community-oriented systems must address diverse user needs and interests, and assume widely varying levels and types of sophistication, motivation and participation. They must manage a great range and volume of community of information, and enable an assortment of activities. These systems must accommodate a range of computer and networking platforms; it is easy to exclude people without intending to do so. Sustainable community computing projects must ultimately rely on local resources, often exclusively on volunteer effort. Finally, community-oriented systems must be extensible; technology will surely continue to evolve, but a community cannot reimplement its applications and data with every ripple of innovation.

The design themes being explored in MOOsburg provide an approach to these challenges. Better integrating the community’s information with physical community places, and introducing greater interactivity relative to the static HTML paradigm can reduce the effort and expertise required for fruitful participation. Enabling more creative contributions and control of the technology by community members can enhance the feasibility of end-user development and maintenance of MOOsburg. Relying on standard Web-based software and open-system components can increase the chance that MOOsburg will provide an extensible infrastructure for further innovation.

MOOsburg supplements face-to-face opportunities for cooperation in a community. It is not our intention, nor do we expect, to diminish the frequency or the satisfaction of encounters among neighbors as they move around the physical community. Rather, MOOsburg can enable encounters that might otherwise not have taken place conveniently or at all. It can enhance the feasibility and richness of casual discussion with a neighbor whose work and/or social schedules make face-to-face encounters difficult. It can ease the process of finding and connecting with individuals or organizations for shared activities. And participation for whatever reason ineluctably causes incidental exposure to community information and cultural norms. An emergent effect of such interactions throughout a community is to increase local social capital.

The significant challenges that face us now, as discussed in Section 5, pertain to adoption, scaling and sustainability. We are working to attract interest and creative participation in MOOsburg from throughout the community, to increase the accessibility and the reward of building and using MOOsburg, and to better understand and

support MOOsburg as a community infrastructure. The challenge beyond is evaluation: We need to track who participates, what they use MOOsburg for, and the possible causes and effects of unequal participation throughout the community. We need to investigate effects on local business activities and opportunities, public decision-making and participation in local government, home life, community health and well-being. The ultimate question is whether and how the social capital of the Blacksburg community increases. These questions will occupy us for some time to come.

One feature of the MOOsburg project that has become more salient through time is the extent to which the project is really about community learning. We acknowledged this in the title of our 1998 proposal to the Hitachi Foundation—"Facilitating the community as a learning community". But it has come to appear even more important as we go on. The MOOsburg project, like the BEV project it extends, is a community education initiative. The most significant requirement for success, and the most significant outcome as well, is community learning about cooperation, communication, information and technology. But the cultivation of human resources throughout the community is not a discrete objective that can be attained tout court, rather, it is a dynamic nexus of technology development and human development.

In the small, this learning process is necessarily continual, both for us and for our community collaborators. The reason for this is that information and communications technology is itself continually changing. Some of the software that MOOsburg, as described in Sections 3–5, depends upon did not even exist when the project began in 1995. All of us have had to learn together about new possibilities for human–computer interaction and information sharing.

In the large, this learning can be seen as a developmental process in which community domain experts attain more direct control over their technology infrastructures through long-term participation in system design and development. We have recently analysed one such long-term design collaboration involving six public school teachers (Carroll, Chin, Rosson and Neale, 2000). The collaboration with the teachers is perhaps exemplary: through the course of 5 years our role diminished from being initiators to being facilitators, while the teachers' role expanded from being informants and users to being designers of new activities and coaches for other teachers. An appealing trajectory for our MOOsburg collaborations would be to move them along a gradient from projects we directly facilitate, to projects others initiate and we support, to projects we indirectly support by providing tools and components.

A briefer version of this paper was presented at the International Network Conference 2000, 3–6 July, Plymouth, UK, and published in the proceedings of that conference (Carroll, Rosson, Isenhour, Van Metre, Schafer & Ganoë, 2000).

Thanks to Jonathan Kies, Brian Amento, Michael Mellott and Craig Struble for implementing the original MOOsburg software and activities, and investigating the use of MOOsburg, in 1995. Thanks in particular to Craig Struble for continuing to design and develop MOOsburg during 1996–1998. Thanks also to Cara Struble for developing the original concept and implementation for map-based orientation and navigation in Fall 1998. Thanks finally to Dennis Neale, member of the current MOOsburg team, for suggestions.

The MOOsburg project is partially supported by the Hitachi Foundation and the Office of Naval Research. The Virtual School project was partially supported by the National Science Foundation (REC-9554206). Ganoë, Schafer and Van Metre were supported by National Science Foundation Graduate Research Traineeships (DGE-9553458).

References

- BEAMISH, A. (1995). *Communities on-line: community-based computer networks*. Masters Thesis, Department of Urban Studies and Planning, Massachusetts Institute of Technology.
- BELLAH, R., MADSEN, R., SULLIVAN, W., SWINDLER, A. & TIPTON, S. (1985). *Habits of the Heart: Individualism and Commitment in American Life*. Berkeley, CA: University of California Press.
- BENFORD, S., BROWN, C., REYNARD, G. & GREENHALGH, C. (1996). Shared spaces: transportation, artificiality, and spatiality. *Proceedings of CSCW '96*, pp. 77–86. New York: ACM.
- BENTLEY, R., APPELT, W., BUSBACH, U., HINRICH, E., KERR, D., SIKKEL, S., TREVOR, J. & WOETZEL, G. (1997). Basic support for cooperative work on the World Wide Web. *International Journal of Human-Computer Studies*, **46**, 111–136.
- BRUCKMAN, A. (1998). Community support for constructionist learning. *Computer-Supported Cooperative Work*, **7**, 47–86.
- BRUCKMAN, A. & RESNICK, M. (1995). Spring. The MediaMOO project: constructionism and professional community. *Convergence*, **1**, 94–109.
- CARROLL, J. M., CHIN, G., ROSSON, M. B. & NEALE, D. C. (2000). The development of cooperation: five years of participatory design in the virtual school. *DIS'2000: Designing Interactive Systems*, Brooklyn, New York, August 17–19.
- CARROLL, J. M. & ROSSON, M. B. (1996). Developing the Blacksburg Electronic Village. *Communications of the ACM*, **39**, 69–74.
- CARROLL, J. M. & ROSSON, M. B. (1999). The neighborhood school in the global village. *IEEE Technology and Society*, **17**, 4–9, 44.
- CARROLL, J. M., ROSSON, M. B., ISENHOUR, P. L., VAN METRE, C., SCHAFFER, W. A. & GANOE, C. H. (2000). MOOsburg: supplementing a real community with a virtual community. In S. FURNELL, Ed. *Proceedings of the 2nd International Network Conference: INC 2000*, p. 307–316. Plymouth, UK, July 3–6.
- CARROLL, J. M., ROSSON, M. B., ISENHOUR, P. L., VAN METRE, C., SCHAFFER, W. A. & GANOE, C. H. (in press). MOOsburg: Multi-user domain support for a community network. *Internet Research*.
- CARROLL, J. M., ROSSON, M. B., VAN METRE, C. A., KENGERI, R. & DARSHANI, M. (1999). Blacksburg nostalgia: a community history archive. In M. A. SASSE & C. JOHNSON, Eds. *Proceedings of 7th IFIP Conference on Human-Computer Interaction INTERACT 99*, pp. 637–647. Edinburgh, August 30–September 3. Amsterdam: IOS Press/International Federation for Information Processing (IFIP).
- CHERNY, L. (1995). *Conversational registers in a MOO*. Ph.D. Dissertation, Department of Linguistics, Stanford University, Palo Alto, CA.
- CHIN, G., ROSSON, M. B. & CARROLL, J. M. (1997). Participatory analysis: shared development of requirements from scenarios. In *Proceedings of Human Factors in Computing Systems, CHI97 Conference*, pp. 162–169. New York: ACM.
- COLEMAN, J. S. (1990). *The Foundations of Social Theory*. Cambridge: Harvard University Press.
- CURTIS, P. (1992). Mudding: social phenomena in text-based virtual realities. *Proceedings of the 1992 Conference on the Directions and Implications of Advanced Computing*, Berkeley, CA.
- DIEBERGER, A. (1996). Browsing the WWW by interacting with a textual virtual environment. *Proceedings of Hypertext'96*, pp. 170–179. New York: ACM.
- ERICKSON, T. (2000). Making sense of computer-mediated communication (CMC): conversations as genres, CMC systems as genre ecologies. In *Proceedings of HICSS 2000*, Maui, HI, January 2000. Available at http://www.pliant.org/personal/Tom_Erickson/genresEcologies.html.
- FARRINGTON, C. & PINE, E. (1996). Community memory: a case study in community communication. In P. AGRE & D. SCHULER, Eds. *Reinventing Technology, Rediscovering Community: Critical Explorations of Computing as a Social Practice*. Norwood, NJ: Ablex Publishing Company.
- FURNAS, G. W. (1986). Generalized fisheye views. *Proceedings of CHI'86*, pp. 16–23. New York: ACM.

- GIBSON, S., NEALE, D. C., CARROLL, J. M. & VAN METRE, C. A. (1999). Mentoring in a school environment. In *Proceedings of CSCL'99: Computer Supported Cooperative Learning*, pp. 182–188. Mahwah, NJ: Lawrence Erlbaum.
- GLUSMAN, G. & PRILUSKY, J. (1996). Real-time, online conferencing for Biologists. *Conference on Bioinformatics and Structure*, Jerusalem, November 17–21.
- GREENBAUM, J. & KYNG, M. Eds. (1991). *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Erlbaum.
- HAYNES, C. & HOLMEVIK, J. R. Eds. (1998). *High Wired: On the Design, Use, and Theory of Educational MOOs*. Ann Arbor, MI: University of Michigan Press.
- ISENHOUR, P. L., CARROLL, J. M., NEALE, D. C., ROSSON, M. B. & DUNLAP, D. R. (2000a). The Virtual School: an integrated collaborative environment for the classroom. *Educational Technology and Society*, **3**, 74–86.
- ISENHOUR, P. L., ROSSON, M. B. & CARROLL, J. M. (2000b, in press). Supporting interactive collaboration on the Web with CORK. *Interacting with Computers*.
- KIES, J. K., AMENTO, B. S., MELLOTT, M. E. & STRUBLE, C. A. (1996). *MOOsburg: experiences with a community-based MOO*. Technical Report, Center for Human-Computer Interaction, Virginia Tech, Blacksburg, VA.
- KOENEMANN, J., CARROLL, J. M., SHAFFER, C. A., ROSSON, M. B. & ABRAMS, M. (1999). Designing collaborative applications for classroom use: the LiNC Project. In A. DRUIN, Ed. *The Design of Children's Technology*, pp. 99–123. San Francisco: Morgan-Kaufmann.
- MC LUHAN, M. (1964). *Understanding Media: The Extensions of Man*. New York: McGraw-Hill.
- MULLET, K. & SANO, D. (1995). *Designing Visual Interfaces: Communication-oriented Techniques*. Englewood Cliffs, NJ: SunSoft Press (Prentice-Hall).
- NARDI, B. A. (1993). *A Small Matter of Programming: Perspectives on End User Computing*. Cambridge, MA: MIT Press.
- NEALE, D. C. & CARROLL, J. M. (1999). Multi-faceted evaluation for complex, distributed activities. In *Proceedings of CSCL'99: Computer Supported Cooperative Learning*, pp. 425–433. Mahwah, NJ: Lawrence Erlbaum.
- PUTNAM, R. D. (1996). The strange disappearance of civic America. *The American Prospect*, **24**, 34–49.
- RHEINGOLD, H. (1993). *The Virtual Community: Homesteading on the Electronic Frontier*. Reading, MA: Addison-Wesley.
- ROGERS, E. M., COLLINS-JARVIS, L. & SCHMITZ, J. (1994). The PEN Project in Santa Monica: Interactive communication, equality, and political action. *Journal of the American Society for Information Science*, **45**, 401–410.
- ROSEMAN, M. & GREENBERG, S. (1996). TeamRooms: network places for collaboration. *Proceedings of ACM CSCW'96 Conference on Computer Supported Cooperative Work*, Boston, MA, November 16–20.
- SCHULER, D. (1996). *New Community Networks: Wired for Change*. Reading, MA: Addison-Wesley.
- SHNEIDERMAN, B. (1999). *Designing the User Interface*. New York: Addison-Wesley.
- UNCAPHER, W. (1997). New communities/new communication: Big Sky Telegraph and its community. In M. SMITH & P. KOLLOCK, Eds. *Communities in Cyberspace*. California: University of California Press.