[In Proceedings of the Third International Conference on Computer Support for Collaborative Learning, December 1999, pp. 518-526.]

From MOO to MEOW: Domesticating technology for online communities

Patricia Schank, Jamie Fenton, Mark Schlager, Judi Fusco

SRI International, Center for Technology in Learning

Abstract: TAPPED IN is an on-line teacher professional development (TPD) research testbed designed to meet the needs of a large and diverse community of education professionals. The MOO technology that supports the testbed has demonstrated sustainability, usability, desirability, and utility across a wide-rage of activities and users. However, we are quickly coming up against technology scaling and integration issues as our community grows and demands new collaborative capabilities emerging on the Internet. Informed by our experience with TAPPED IN and reviews of related work, we are developing a new online "community-ware" technology called MEOW (Multi-user Educational Online Workspace) which can scale to handle large virtual communities. Under development in Java and related frameworks, MEOW introduces the concepts of persons, places, and things in a way intended to bind together many existing and emerging Internet services (e.g., web, email, ftp, search, recommendation) that are useful to a virtual community. Our goal is for MEOW to become a flexible, powerful, yet inexpensive platform for all forms of educational research and practice on the Internet. We invite others to join TAPPED IN and help us design the social spaces and technology to address the implications of online communities.

Keywords: multi-user environment, community-ware, education community, scalability

Introduction

This paper traces the design evolution of an on-line teacher professional development (TPD) research testbed called TAPPED IN (Schlager & Schank, 1997; Schlager, Fusco, & Schank, 1998; www.tappedin.org). Our partners include nationally-recognized education organizations, preservice and master's degree programs, state and local education agencies, and scores of small groups. Our research goals over the past three years have been to (a) investigate the resources and technological support that TPD efforts need to conduct professionally valuable on-line activities; (b) develop innovative TPD models that integrate face-to-face and asynchronous interaction with collaborative synchronous on-line activities; and (c) identify social, motivational, and technological factors that contribute to (or obstruct) the success and evolution of our on-line TPD community concept.

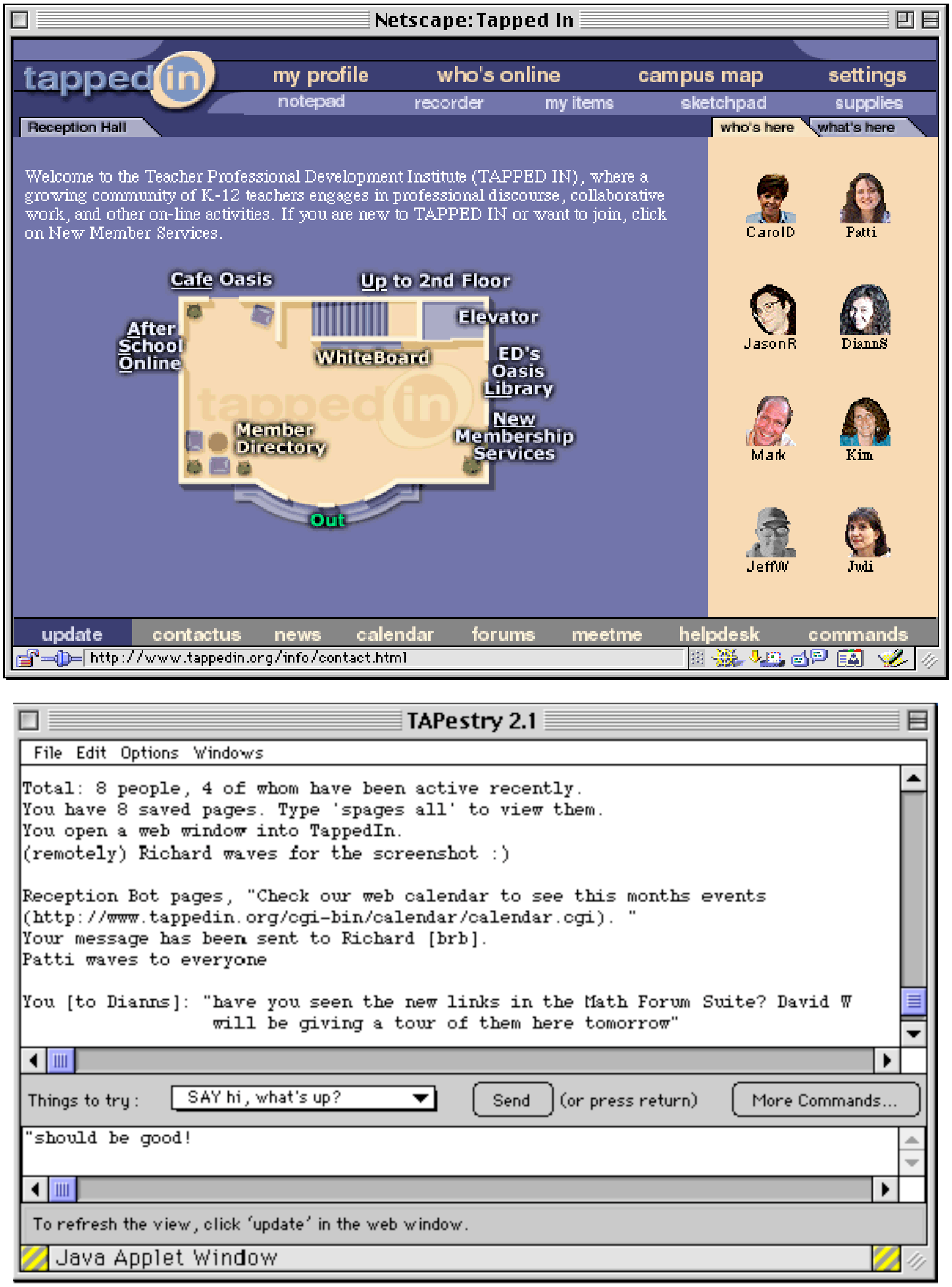
Our research design is based on the theoretical conjecture that self-sustaining on-line professional development communities require (a) the participation of multiple organizations offering a variety of high-quality activities in association with their own mission, (b) community-wide activities through which individual members can "learn the ropes" of the community and its culture, contribute to, and take ownership of the community, and (c) a persistent shared environment that supports a natural flow of communication (e.g., from real-time conferencing to email and discussion boards) and the creation and manipulation of discourse-support artifacts. Our methodology can be described as a "design experiment," interweaving research in the design and implementation of TPD activities and technical capabilities that we conjecture will help establish and sustain on-line TPD communities.

The online venue that supports the testbed is implemented on a platform-independent, Web-based, real-time environment designed to meet the needs of a large and diverse community of education professionals (see Figure 1). Activities occur in virtual rooms that provide a basic yet powerful set of synchronous and asynchronous communication mechanisms and persistent support tools (e.g., whiteboards, notes, tape recorders, and shared Web viewers). On any given day, TAPPED IN members can attend online meetings and other activities hosted by partner organizations, conduct their own activities, take online courses, bring their students online, experiment with new ways to teach, or expand their circle of colleagues by participating in community-wide events. We continuously collect data on our member activity, with their consent (e.g., objects they access, rooms they visit, when they log in and out, what commands they use; but not the content of what they say or do). More immediately, members of the community provide us with feedback on the design of both our technical and social support infrastructure on a daily basis. The combination of formal data collection and informal feedback has helped guide a process of continuous improvement of both our technology and user support services.

In this paper we focus on how our technology has evolved and how we anticipate it continuing to evolve over the next several years as computational power, bandwidth, and user demand continue to increase.

Technology design

The TAPPED IN concept is not tied to a particular technology platform, but rather is a set of capabilities that are central to collaborative work and professional development (e.g., synchronous and asynchronous interaction, persistence, awareness; Sproull & Kiesler, 1991; Kollock, 1998; Renyi, 1996; Lave & Wenger, 1991) and necessary for practical reasons (e.g., cross-platform, low bandwidth, and web accessibility; Schlager & Schank, 1997). We constructed TAPPED IN from a descendant of the LambdaMOO core (Curtis, 1992) because it was (and still is, in many ways) the most appropriate technology available to satisfy this set of design constraints.



Welcome to the Teacher Professional Development Institute (TAPPED IN), where a growing community of K-12 teachers engages in professional discourse, collaborative work, and other on-line activities. If you are new to TAPPED IN or want to join, click on New Member Services.

http://www.tappedin.org/info/contact.html

```
Total: 8 people, 4 of whom have been active recently.
You have 8 saved pages. Type 'spages all' to view them.
You open a web window into TappedIn.
(remotely) Richard waves for the screenshot :)

Reception Bot pages, "Check our web calendar to see this months events
(http://www.tappedin.org/cgi-bin/calendar/calendar.cgi). "
Your message has been sent to Richard [brb].
Patti waves to everyone

You [to Dianns]: "have you seen the new links in the Math Forum Suite? David W
                  will be giving a tour of them here tomorrow"
```

```
"should be good!
```

To refresh the view, click 'update' in the web window.

Figure 1: TAPPED IN user interface.

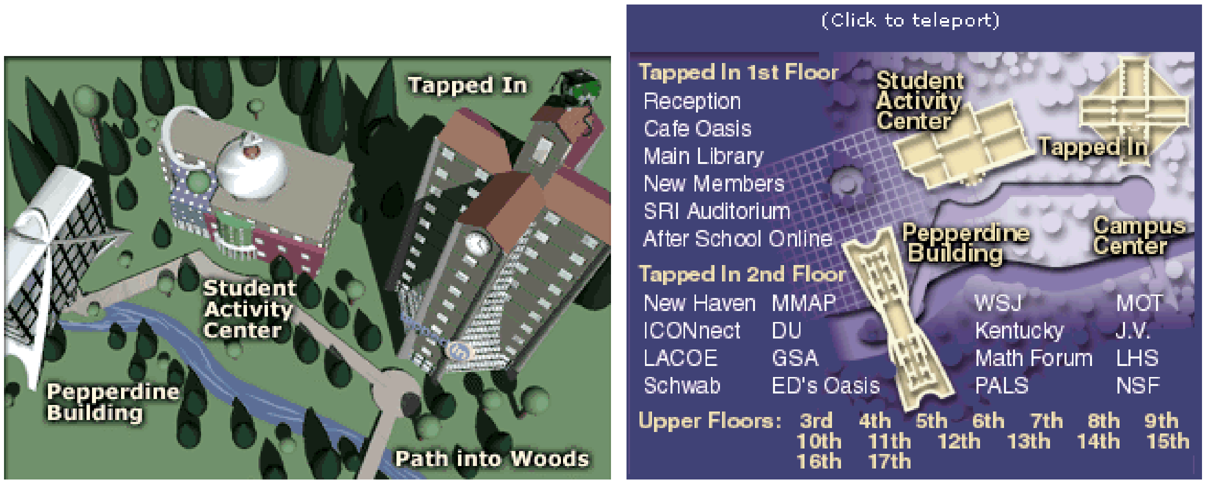


Figure 2: TAPPED IN campus view and teleport map.

MOOs have effectively supported professional and educational communities for several years (Bruckman & Resnick, 1993; Haynes & Holmevik, 1997). Such use has demonstrated the sustainability, usability, desirability, and utility of MOO-based multi-user environments across a wide-rage of activities, users, and age groups. Many concepts implemented in the technology have been emulated by developers of next-generation multi-user virtual environments like TeamWave (www.teamwave.com; Roseman & Greenburg, 1996), Electric Communities, www.communities.com), PlaceWare (www.placeware.com), and Microsoft's Virtual Worlds Group (http://www.research.microsoft.com/vwg; Vellon, Marple, Mitchell, & Drucker, 1998). We have also found the MOO environment to be a very useful research platform. For example, because of the dynamic nature of the MOO programming language, it is easy to fix problems and add new features in response to user requests while the system is running. We were also able to easily integrate mechanisms to automatically collect data on member activity.

Our interface design differs significantly from those of other MOOs. We specifically designed the environment to resemble a conference center, to evoke a professional atmosphere and encourage the kinds of discourse one would find at a conference facility or institute, and reinforced this metaphor with our graphical layout. The vast majority of MOOs only support a text interface with no room drawings, and the few that have a web interface (e.g., Diversity University, LinguaMOO) graphically represent only people and objects, not rooms. As shown in Figure 1, the TAPPED IN web interface includes a room map, which we have found helps orient our users and gives novices a sense of immersion in an online "place." We design room graphics for each of our partners based upon their specifications, and offer a clickable "teleport" map for the entire campus (see Figure 2). However, a room map takes up screen real estate and competes with other information (people, objects, rooms, commands, conversation, etc.) for the user's attention. We realized that the interface needed to be highly usable and help users selectively attend to their current focus in this sea of information (e.g., Nielson, 1995; Shneiderman, 1998).

With these constraints in mind, we worked with an interaction designer in early 1999 to create the current (new) interface that allows users to selectively reveal objects and people in the current room via tabs when needed. (See http://www.tappedin.org/info/designhistory.html for a summary of the design history, which involved over 40 design iterations.) Additional information (like who's online, supplies available, the campus map, detailed description of a person or object) replaces the room graphic when requested (i.e., clicked on). A click on the room tab restores the room map whenever needed. We also employ a separate window for text interaction, because users generally spend most of their time conversing. They often want to enlarge the text window and make it their main focus for an extended period (e.g., to view the history of the conversation with minimal scrolling). However useful, multi-window designs can also be confusing to new users (windows can get hidden), so we plan to develop an all-in-one interface alternative.

Informal feedback from our users suggest that we met our design goals to provide (a) one- or two-click access to most information, (b) a professional environment that is contextualized to each group's work, (c) compact windows that fit on even the smallest monitors, and (d) a layout that progressively reveals information to help the user focus her attention.

Designing for growth

Early on, the addition of a web interface (served by the MOO) and the automatic data collection mechanisms made it so that our MOO couldn't support more than about 50 concurrent users without noticeable performance lag. This didn't overly concern us at the time, since we thought that when we reached 4 or 5 partner organizations and 500 members, we would have hit "critical mass" enough to sustain the community with an average of 20 or more concurrent users logged in at a time on weekdays. We also assumed that by the time we outgrew the MOO, scalable commercial environments would exist, since so many groups were attempting to develop them. We were wrong on both counts!

Our membership and list of partners has grown faster than we expected, primarily by word of mouth. As of September 1999, we have 14 partners and about 6000 members. Our growth rate since September 1998 has been approximately 200 new members each month, double that of the previous year. As membership has grown, our login rate has remained steady: approximately 15% of the membership logs in per month, typically for six one-hour sessions. Members log in every day of the week and almost around the clock, from around the world. Logins are relatively equally divided Monday through Friday and shrink by about two-thirds on weekends. Approximately half of our members describe themselves as K-12 teachers; the balance is composed of relatively equal proportions of researchers, university faculty and graduate students, staff developers, school support and administration staff, and preservice teachers. Hundreds of guests "drop in" each month, having seen TAPPED IN events posted on other websites. Many join after visiting and further spread the word about TAPPED IN, accounting for most of the increase in our growth.

Even with this greater-than-expected growth, we feel we haven't reached critical mass. We are seeing the benefits that member interaction in a large and diverse community can provide, and still see a need for more activities and resources (including more members) to keep our members actively engaged and meet their needs. Our technology has also evolved to support (and attract) this fast growing and diverse community. One near-term problem we face is that MOOs are single-threaded, non-distributed environments, which means that they cannot scale up to handle arbitrarily large virtual communities. Adding the web interface and automatic data collection mechanisms consumed even more processing time so that our environment couldn't handle as much traffic as standard text-only MOOs, which usually reach their performance limit at 200 concurrent users. Some developers have "linked" MOOs together (e.g., for special events like large conferences), but such efforts have only temporarily connected a few MOOs or have allowed cross-MOO communication from only one room in each MOO—users can not "bring objects with them" across MOOs, since MOOs do not have built-in mechanisms to support distributed objects. To continue to provide reasonable performance with our increase in activity and additional process-consuming features, we've implemented code optimizations and moved to a new, faster Sun server with a gigabyte of RAM. We estimate that such optimizations have at least tripled the number of simultaneous users we can support to around 200 before the system exhibits unacceptable lag.

At present, the number of concurrent users tends to fluctuate between 10 and 100, so we haven't yet seen significant performance loss. However, given our current growth rate, we expect to experience occasional problems (e.g., increased "lag") within the next year, and serious problems within a couple of years if we stay with the current MOO platform. We are also beginning work with the Kentucky Department of Education and the Los Angeles County Office of Education which, if successful, will mean scaling up over time to a large proportion of over 100,000 teachers in these two regions. Clearly, we need a scalable technology solution to be able to support such growth.

Another factor that has greatly affected our growth is the evolution of Java. Early on (spring 1997) we created a Java applet so that users could login to TAPPED IN from any Java-aware browser, and also project web pages to others in the same room. Prior to the Java applet, users had to download a freeware MOO client application for their platform, but these clients do not support the sharing of web pages. The applet, called TAPestry, became more of a "marketing tool" during this early period, since most of our users didn't have Java-aware browsers, or if they did, Java ran unbearably slow on their older machines. As the Java virtual machines implemented in browsers got better (e.g., with version 3 and 4 browsers) and educators got newer computers, TAPestry use increased and we added more features to it (like a whiteboard and command help) based on user feedback. Now well over half of TAPPED IN logins are through TAPestry, and virtually all guests connect this way. We recently released an application version of TAPestry that allows local logging (not allowed by applets due to Java security), offers an improved whiteboard that supports images and text, and lets frequent users bypass the applet download time for faster access.

Criteria for effective online community technology

Designing and evaluating technology to support online community is challenging. Informed by our experience with TAPPED IN and other research on online communities (e.g., Smith & Kollock, 1998; Sproull & Kiesler, 1991), we identified several criteria to guide design and to help select between candidate environments:

Persistence. People, places, and objects do not disappear between sessions—they are saved to disk, most likely to a database, and are available to new sessions. Individuals must have persistent identities in order to establish reputations, trust, and responsibility.

Transparency. The system is easy to use and understand (e.g., provides a familiar spatial metaphor, offers clear commands and a well-designed user interface, supports awareness of other users). The system is unified; identities transfer across services (e.g., discussion boards, mailing lists, recommendation services), providing transparent access to each service.

Security. The system is reliable and safe, and privacy of sensitive information is maintained. Users can be given access to objects in a controlled manner without compromising functionality.

Extensibility. Improvements or new services are easily implemented (e.g., custom interfaces and tools can be added in a modular way; some new features and fixes can be implemented while the system is running via a dynamic scripting language).

Scalability. The system can support large numbers (e.g., millions) of concurrent users while maintaining acceptable performance, and allows movement of people and objects across servers.

Versatility. A diverse population of individuals and organizations can be served (e.g., a variety of services and world objects can be provided based on user needs).

Openness. Users and other developers can improve the system (i.e., it's not a proprietary solution, but rather is open source or has a published API based on standard protocols).

Accessibility. Cross-platform clients and multiple interface options are available or can be added without too much effort (e.g., ranging from low-bandwidth text solutions to custom clients with support for multiple media types).

Internationalization: The system should be adaptable to different languages or regions, conforming to local requirements and customs without engineering changes.

With these criteria in mind, we have investigated scores of environments advertised as "community-ware" (e.g., DiGiano & Schank, 1998), and have been fairly discouraged by the current options. Most do not support persistent places, identities, and objects; many only support the Windows platform (at least half of our members are Mac users), and almost all require that complex client software be installed on the users machine. We believe these to be significant barriers to acceptance. For example, web-based chat-solution providers (e.g., Internet Factory, Ichat) are limited to only basic discussion support and don't create the sense of a world inhabited with objects and behaviors; thus, they do not foster the kinds of social norms that we believe are central to successful communities of practice and group collaboration. More advanced custom-client based multi-user environments such as The Palace, Microsoft's Vworlds, and those based on VRML require a custom client installation (which often only works on one platform), and do not provide lower-bandwidth options (e.g., text or 2D interfaces) for users with slow internet connections or older computers—a common situation for many educators. Moreover, none of the alternatives adequately address scalability or distributed objects. Nor do they offer a world-building programming language, a compelling feature of MOOs by which one can dynamically add new functionality to the system.

In summary, MOO technology has enabled us to provide a single, low-cost, highly supportive venue for scores of large and small TPD projects around the country to begin experimenting with on-line activities within our community-based model. However, scaling and integration issues spurred us to investigate alternative platforms, and eventually design our own technology guided by the criteria identified above.

MEOW: A new community-ware technology

MEOW (Multi-user Educational Online Workspace) is a software technology we are developing, using Java and related frameworks, for constructing Internet based educational communities. In this section, we describe the existing and proposed features of MEOW. A simple version of MEOW that demonstrates some of the elements described is undergoing beta testing.

Design concepts

MEOW introduces the concepts of persons, places, and things in a way intended to bind together many additional services (e.g., web, email, ftp, search, collaborative filtering/recommendation services) furnished to a virtual community. Inspired by MOO technology, MEOW tracks the motion of people and things from place to place. For example, MEOW will be able to track which web pages a teacher is viewing so that, if appropriate, students can follow along on a web tour. While TAPPED IN is our initial target "customer," our goal is for MEOW to become a powerful yet inexpensive platform for all forms of educational research and practice on the Internet.

MEOW will enable users to maintain identity, hold discussions, possess and exchange objects, and travel between places (which may reside on separate servers), carrying objects with them. The technology is being designed to scale up to handle arbitrarily large virtual communities where educational activities can be conducted. It will include support for live discussions, presentations, and asynchronous discussion. A large-scale virtual community will require users to bring new types of objects across system boundaries; MEOW includes as part of its architecture the appropriate security policies that make mobile objects possible.

Conventional web browsers can serve as the client-side interface to the environment, enabling less-sophisticated users to participate since minimal client setup is required. MEOW allows this by separating the representation of "model" and "view." The model of the world is intended to be abstract and independent of any individual's view of it. This allows different users with different capabilities to be able to participate. For example, some users with very simple computers will only be able to read textual descriptions of world state changes. This is important for educational communities, as the equipment available in many institutions is not state-of-the-art. Others may be able to view our world as an animated three-dimensional virtual reality. A software subsystem (called a "narrative generator") will be responsible for converting changes in the state of affairs into user perceived phenomena (e.g., a text view, browser view, or custom 3-D client view).

MEOW is designed to operate as a distributed cluster of servers. This allows an educational community to grow to an arbitrary size. Moreover, a citizen of one virtual institution can visit another, and bring objects along with them. MEOW will allow new types of world objects and object behaviors to be added to the environment dynamically, as it runs (as the MOO scripting language allows in LambdaMOO environments).

Current implementation

A simple version of MEOW is currently undergoing beta testing with an external community. This preliminary version does not fully demonstrate all of the elements described above, but this process is necessary for us to discover the idiosyncrasies and coping strategies necessary for a thin client multi-user environment. MEOW is written to run inside of a Java servlet runner, an environment designed to efficiently run Java programs on the server computer on behalf of client computers. This makes it easy to quickly field web-based requests for updates and state changes. Current features of the MEOW prototype include: Web-based chat room functionality; avatar, place and communication objects; old and recent-model browser interfaces (with an optional Java applet to receive multicasts); a user command parser; and a user authentication database.

The next phase

The next phase of MEOW will involve implementing basic, persistent, ownable discourse-support objects (e.g., notes, documents, presentations), getting multiple servers talking with each other (much of the mechanism necessary for this is already present in the Java environment as provided by JavaSoft), offering 2D and 3D spatial interfaces, and integrating a secure scripting language.

Several near-term issues will need to be addressed before this next phase can be completed. For example, what is the best way to build the world object database? The state of world objects must be stored persistently. Existing technology for interfacing Java with relational databases could be used, but would have poor performance. Object databases might be faster, depending on the complexity of the database schema, but don't yet have a cross-vendor standard programming API. We want to stay away from technology that locks the developer into one proprietary solution. Another alternative may be to develop a simple scheme based on human readable text representations (e.g., in XML) of object state. As the interface between the rest of the system and the persistent object database would be carefully defined, it should be possible to adopt other persistence schemes easily.

The "server coordination framework" includes interfaces to other services in the server cluster, as well as to other MEOW servers on the Internet. This means MEOW must be able to talk to other services, like mail and Usenet, using standard Internet protocols (e.g., SMTP, NNTP). MEOW servers also need be made to talk to each other through a secure communications protocol. A user will be able to travel from server to server bringing objects with her, facilitated by arranging for each user to have a "home base." When a user arrives at a new region, the transfer of her identity would be negotiated between the origin computer, the home base computer, and the destination. Objects that are not immediately visible (e.g., ones in a user's "pocket") would be opportunistically instantiated as they are used—provided that they are permissible in the new region. For example, requests to bring out an object that is not certified as "classroom-safe" (e.g., non-safe art, violent games) could be rejected automatically by the region, if so specified in the "terms of service" set by the region manager. By default, the home base computer will be the ultimate repository of a user's possessions and the ultimate authority for their whereabouts.

In order to provide world objects, we need to develop a world object support library. This library will contain the basic pieces out of which collections of components are made. Included are facilities for assemblages of components to register themselves, locate each other, store and load themselves from persistent memory, and send messages to each other. We envision two levels of world objects: basic objects that would be built into the foundation of MEOW, and advanced world objects that might be part of advanced "world object kits" released separately from MEOW. The basic objects include fundamental building blocks that any virtual world would require (e.g., people, places, things). Advanced kits include objects targeted for a particular kind of community (e.g., assessment tools for an educational community). Different kinds of communities (e.g., an online mall vs. a single-parent support group) might employ different kits of objects. Many advanced objects might be useful to more than one community, and might be included in more than one released world object kit.

Another compelling feature of MOOs is the ease with which functionality can be extended while the system is running. We expect to provide a similar feature in MEOW through a secure world building/scripting language. Although MEOW will be written primarily in Java, Java has limitations as a world building language; for example, it's not very interactive, and relies upon a conservative security model that can prevent users from accessing functionality that might be safe. Two good candidates for scripting in a Java-based system are Jpython (www.jpython.org) and E (www.erights.org), since they are both Java-based and support simple and effective interoperability with Java code. Of these, E seems superior due to its advanced security features. Security is vital in any community; in the context of education, for example, security is critical for payment of tuition, purchasing of supplies, grading, testing, and maintaining privacy of sensitive information. E is a lightweight, easy-to-use scripting language that extends work on "capability security" to provide effective support for managing interactions between distributed components and give users access to objects in a controlled manner that more conservative models disallow. (See the ERights website, www.erights.org, for more information.)

Discussion

The design of MEOW is informed by our experience developing and supporting TAPPED IN and reviews of related work. MEOW attempts to address the scalability problems of current MOO technology, and to create a sense of integration from a diverse collection of services and protocols. We are currently testing an early MEOW prototype, and plan to publish the source and/or API's so that other developers can help us in the process. Areas where future innovations will take place in MEOW development include the scripting language and its security features, the persistent object database scheme, and integration of knowledge management functionality. This latter feature will help users create a structured group memory and communicate learning results to a wider Internet community.

Whether or not MEOW succeeds as a future community-ware platform, or we migrate to another commercial or open source platform that supports the features we desire, we will need to address the implications of online education communities of practice for both TAPPED IN and public education if we (and other similar projects) succeed. For example, will communities compete rather than cooperate? What do we do about disruptive users? Will the ability to engage in TPD activities on-line cause school districts to reallocate TPD budgets and how? Will teacher's unions denounce the on-line communities as competing with their interests or as a subtle way to get teachers to put in more hours without being paid? Will these developments affect movements such as home schooling, charter schools, or commercial models for K-12 education?

We also need to understand the level of technical vs. human support needed in an online community. We have found that the labor required to support a rapidly growing community and maintain quality interactions is quite intense. Community leaders (staff and volunteers) greet and mentor new members, and also provide answers, resources, continuity, group identity, and social information filtering. Their effort is invaluable, but no "mere mortal" can do all things for all people at all times of the day. We look forward to the time when technology, like intelligent agents, can supplement the support humans now give. However, many social issues arise when such technology is considered. For example, what kind of human support shouldn't be replaced by technology? How will members feel about interacting with agents, and can they turn them on and off? How should a user be notified that they are interacting with an agent, not another human? What privacy issues will emerge if agents monitor conversations for the sake of providing recommendations and help?

There are several places where we wish agents could augment and possibly replace some human support. For example, a proactive agent could anticipate needs and suggest resources (activities, people, etc.) based on a member's interests. These preferences could be specified in a profile or culled from activities and conversations in which they have engaged. For new users, it would be especially valuable if an agent could watch the commands issued and the mistakes made, and suggest tips for more efficient use of the technology. Another type of agent could help in the preparation and facilitation of online meetings, based on guidelines already developed by experienced members of the community. Before a meeting, an agent could help the speaker prepare an agenda, suggest resources, and provide tips on pacing and meeting management. During the meeting, the agent could moderate discussions, suggest resources based on questions that are raised, assist speakers in presenting an agenda, monitor the pace and encourage participation when appropriate, manage discussion threads by suggesting irrelevant side conversations be whispered or taken to another room, and field technology questions. After the meeting, the agent could summarize the discussion and publish both a transcript and the summary.

We realize that the future technologies we've described in this paper won't be easy to implement, but we think the related social and policy issues will likely be even more difficult to address. We invite those who study such issues to become part of the community, and help us design and structure social spaces that address these issues.

Acknowledgments

We thank Richard Godard, Linda Polin, Hulda Nystrom, BJ Berquist, Terrie Gray, Courtney Glazer, and countless other volunteers for their ongoing help in making TAPPED IN a success. We are also grateful to Larry Hamel for his technology advice and thoughtful comments on earlier drafts of this paper; Nick Kledzik for consulting with us on Java development and ensuring that TAPestry works on the Macintosh; Martin Fong, Richard Godard, and Rob Kinninmont for their invaluable input and feedback on the design of MEOW; and Aaron Becker, our interaction designer, for helping us create an attractive and useable interface. The work reported in this paper was funded in part through NSF Grant REC-9725528 and by internal funding from SRI International.

Bibliography

Bruckman, A., & Resnick, M. (1993, May). Virtual professional community: Results from the MediaMOO project. Presented at the Third International Conference on Cyberspace, Austin, TX.

Curtis, P. (1992). Mudding: Social phenomena in text-based virtual realities. *Proceedings of the Conference on Directions and Implications of Advanced Computing,* 48-68. Berkeley, CA:

Computer Professionals for Social Responsibility.
ftp://ftp.lambda.moo.mud.org/pub/MOO/papers/DIAC92.txt

DiGiano, C., & Schank, P (1997). Virtual places in perspective: A review of telecollaboration tools. Technical report. SRI International, Center for Technology in Learning. http://insider.ctl.sri.com/ctl/whitepapers/public/telecollab.html

Haynes, C., & Holmevik, J. R. (Eds.) (1997). *High wired: On the design, use, and theory of educational MOOs*. Ann Arbor, MI: Univ. of Michigan Press.

Kollock, P. (1998) Social dilemmas: The anatomy of cooperation. *Annual Review of Sociology*, 24:183-214.

Lave, J. & Wenger, E. (1991). Situated learning: Legitimate peripheral participation. Cambridge, UK: Cambridge University Press.

Nielson, J. (1995). Usability Engineering. AP Professional, Boston, MA.

Renyi, J. (1996). *Teachers taking charge of their learning: Transforming professional development for student success*. National Foundation for the Improvement of Education, http://www.nfie.org/takechar.htm

Roseman, M. & Greenburg, S. (1996) TeamRooms: network places for collaboration. *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work*, 325-333. New York, NY: Association for Computing Machinery.

Schlager, M., Fusco, J., & Schank, P. (1998). Conceptual cornerstones for an on-line community of education professionals. IEEE Technology and Society, Special Issue on Computers in the Classroom: The Internet in K-12, 17 (4), 15-21.

Schlager, M., & Schank, P. (1997). TAPPED IN: A new on-line community concept for the next generation of Internet technology. In R. Hall, N. Miyake & N. Enyedy (Eds.), Proceedings of the Second International Conference on Computer Support for Collaborative Learning, pp. 231-240. Hillsdale, NJ: Erlbaum.

Shneiderman, B. (1998). Designing the User Interface. Addison-Wesley: Reading, MA.

Smith, M., and Kollock, P. (Eds.). Forthcoming (1998). Communities in Cyberspace. London: Routledge.

Sproull, L., & Kiesler, S. (1991). Connections: New ways of working in the networked organization. Cambridge, MA: MIT Press.

Vellon, M., Marple, K., Mitchell, D., & Drucker, S. (1998). The architecture of a distributed virtual worlds system. Technical Report. Microsoft Corporation, Virtual Worlds Research Group. http://research.microsoft.com/vwg/papers/oousenix.htm

Authors' addresses

Patricia Schank (patricia.schank@sri.com)

SRI International, Center for Technology in Learning; 333 Ravenswood Ave.; Menlo Park, CA 94025. Tel. (650) 859-3934. Fax (650) 859-3673.

Jamie Fenton (jfenton@unix.sri.com)

SRI International, Center for Technology in Learning; 333 Ravenswood Ave.; Menlo Park, CA 94025. Tel. (650) 859-2905. Fax (650) 859-3673.

Mark Schlager (mark.schlager@sri.com)

SRI International, Center for Technology in Learning; 333 Ravenswood Ave.; Menlo Park, CA 94025. Tel. (650) 859-2881. Fax (650) 859-3673.

Judith Fusco (judith.fusco@sri.com)

SRI International, Center for Technology in Learning; 333 Ravenswood Ave.; Menlo Park, CA 94025. Tel. (650) 859-6207. Fax (650) 859-3673.