Remote Exploratoriums: Combining Network Media and Design Environments

James Ambach, Corrina Perrone, Alexander Repenning
James Ambach is the corresponding author

Center for LifeLong Learning and Design,
Department of Computer Science,
University of Colorado, Boulder.
{ambach, corrina, ralex}@cs.colorado.edu
Phone: (303) 492-1503
Fax: (303) 492-2844

http://www.cs.colorado.edu/~ambach/Home.html
http://www.cs.colorado.edu/~corrina/Home.html
http://www.cs.colorado.edu/~ralex/Home.html

Abstract–In an educational context, World Wide Web clients such as NCSA Mosaic® are of limited value because they put learners in the role of information absorbers. Drawing on a museum analogy, learners using Mosaic can be perceived, similar to museum visitors, as passive observers of exhibits. Despite the richness of exhibits in terms of the amount of information as well as the use of multimedia, activity is restricted to navigation through real (museum) or hyper (Mosaic) spaces. More effective learning would include constructive activities that are more engaging than just browsing information spaces. Distance education could be improved by augmenting *network media* with *design environments* to create *remote exploratoriums* that encourage highly interactive, engaged learning experiences. In contrast to classical museums, exploratoriums, such as the one in San Francisco and several children's museums, feature hands-on learning exhibits that are not only observed but can be actively manipulated. The Agentsheets Remote Exploratorium is a mechanism to facilitate the easy exchange and distribution of interactive educational exhibits through the Internet. Agentsheets is a programming substrate to create design environments. In this paper, we discuss the implications of combining a network medium and a design environment in order to support distance education.

INTRODUCTION

The World Wide Web (WWW) and its client applications, such as NCSA Mosaic®, provide access to rich information repositories that are similar to museums. In museums, people are presented with different exhibits and are allowed to browse through them viewing information at will. However, interaction within museums, or while browsing WWW pages, is limited in the sense that text, pictures and even videos can only be passively observed. Especially in an educational context, activity restricted to the navigation through real or hyper information spaces does not engage people in active learning experiences. Exploratoriums, such as the one in San Francisco, provide more engaging learning approaches by featuring hands-on experiences through interactive exhibits that are not only observed by learners but are also actively manipulated. This paper outlines our experiences in creating a remote exploratorium by combining network media (WWW servers and the NCSA Mosaic client application) with design environments [1] to support engaged, remote learning.

Network media have the ability to communicate many rich information types such as text, pictures, sounds and videos through global computer networks like the Internet. The World Wide Web is a system developed to allow the easy exchange of multimedia information over the Internet. Although the WWW makes this information readily available, it does not provide the user with constructive opportunities to apply any new knowledge gained from the information.

Design environments, on the other hand, are software applications that allow users to build artifacts. For example, a word processor can be seen as a design environment because through its use, documents are created. In the spirit of constructionist learning [2] design environments provide a means for learners to actively apply knowledge through the construction of personally meaningful artifacts. Agentsheets [3] is a programming substrate for creating visual design environments that can include user-directed simulations, models and construction kits. Network media can be used in conjunction with design environments by providing educational scaffolding in the form of instructional information as well as related topics. The result is an exploratorium-like learning environment that can be accessed over a network.

This paper describes our experiences creating the Agentsheets Remote Exploratorium. First, we discuss our motivations. We briefly describe Agentsheets and some of the design environments that have been created with it, and then we describe how we combined Agentsheets with Mosaic. We have included a scenario describing the use of the Agentsheets Remote Exploratorium, and we end with a discussion of the issues concerning the development and use of remote exploratoriums to support distance learning.

FROM MUSEUMS TO EXPLORATORIUMS

Increasingly, network applications and architectures are changing to allow the distribution of multimedia information. Although the addition of color images, sounds and videos to text provides new communication capabilities, the information that can be distributed is still presented in a passive, instructionist method that emphasizes knowledge acquisition but does not provide a means for knowledge application. The essential problem with merely providing this kind of information is that the knowledge representation, however rich it may be, lacks any sort of interactivity that would engage a learner to participate within that representation [4]. Interactive and dynamic information sources, such as simulations, models and other design environments, assist learners in the construction of their own intellectual structures and allow the personalization and ownership of new knowledge by providing a means to apply that knowledge to a task relevant to the learner [5]. An ideal way to support distributed communities of learners is to augment current networking applications to allow the distribution of design environments with multimedia information.

The impetus to create the Agentsheets Remote Exploratorium arose from the need to support a growing and increasingly remote Agentsheets user community. As this community grows and becomes more distributed, it can no longer be supported by direct contact. Networking tools need to be developed to support the community by providing ready access to Agentsheets and its applications.

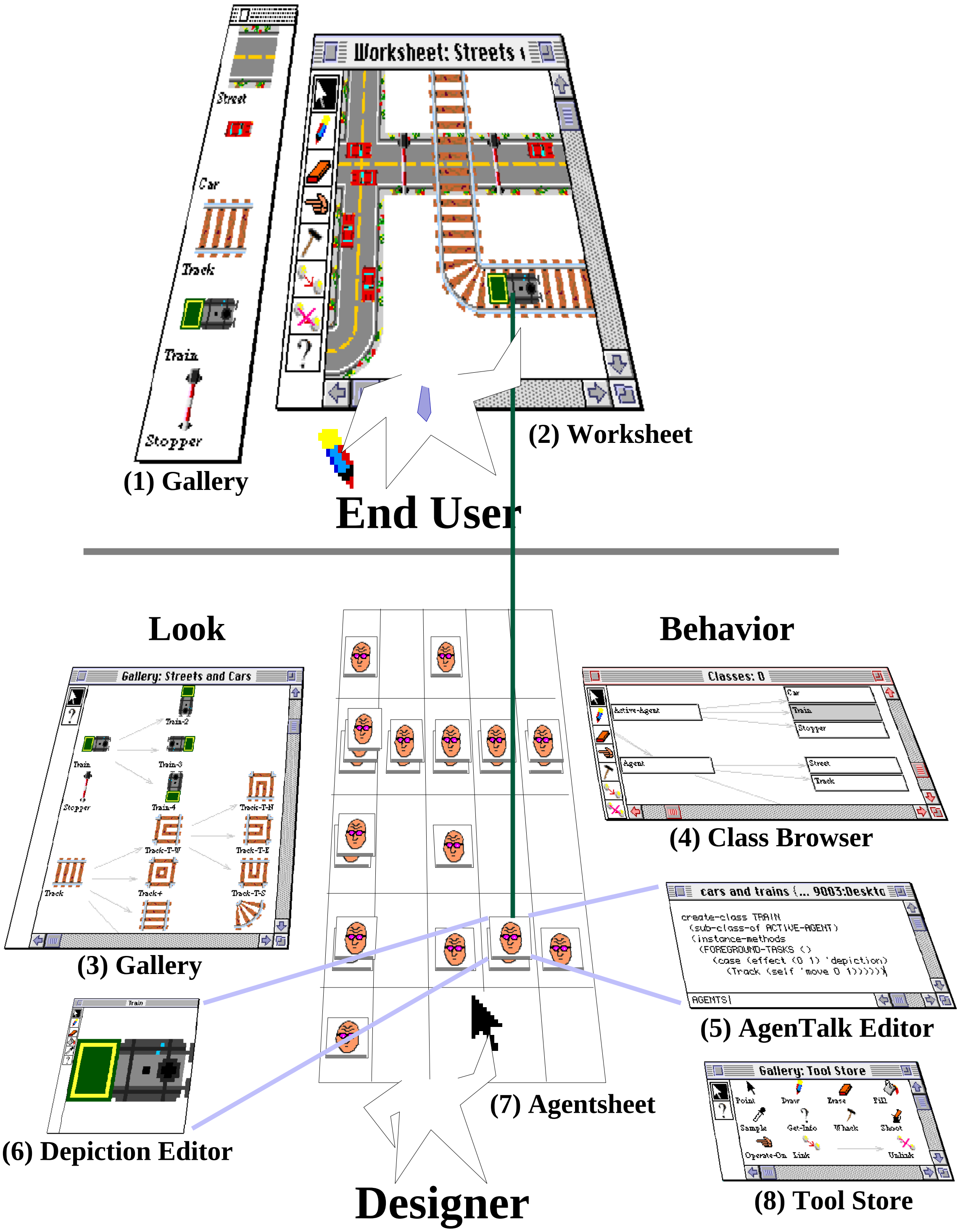
Our choice for an underlying technology to support this distribution was the Internet, given that most people interested in Agentsheets had access. However, although the use of a network is necessary for providing global communication, we found it insufficient for supporting a community. Merely providing access to the software with no additional information is not enough to entice and support interested users. Instead, we wanted to create a place where users could access and interact with the Agentsheets applications while at the same time browse pertinent information. Our solution was to combine Agentsheets with the WWW and Mosaic to create the Agentsheets Remote Exploratorium. We realized that the creation of this environment had potential impact beyond the Agentsheets community to distributed communities of learners in general.

AGENTSHEETS: A TOOL FOR CREATING DESIGN ENVIRONMENTS

Agentsheets is a software application for creating design environments. Agentsheets and Agentsheets applications run on the Apple Macintosh®. In the past four years, Agentsheets has been used to create more than fifty educational and industrial applications such as construction kits, interactive simulation environments, visual programming languages, and other design environments. Similar to spreadsheets and cellular automata [6], Agentsheets uses a visual formalism [7] based on a grid structure. A grid location in Agentsheets may contain any number of stacked autonomous computational units called agents (Figure 1 (7)).

Agentsheets provides different tools for end-users and designers. End-users compose programs by selecting components in the gallery (1) and arranging them in a worksheet (2). Designers view worksheets (2) as agentsheets (7), i.e., agents organized in a grid. They define the look of agents using the Depiction Editor

(6) and store the depictions in Galleries (3). Designers define agent behavior with the AgenTalk editor (5) by reusing existing Agentsheets classes found in the Class Browser (4), and create or subscribe to tools located in the Tool Store (8). Tools enable end-users to interact with agents even when an agent is doing something. This interaction style extends direct manipulation schemes by allowing end users to more flexibly interact with applications involving large numbers of autonomous agents. This interaction style is called participatory theater [8].



```
create-class TRAIN
  (sub-class-of ACTIVE-AGENT)
  (instance-methods
    (FOREGROUND-TASKS ()
      (case (effect (0 1) 'depiction)
        (Track (self 'move 0 1))))))
```

```
AGENTS|
```

Figure 1. End-users and Designers

The Agentsheets substrate has been used to create a number of different applications in domains as varied as voice dialog systems [9] computer network design and interactive drawing applications to name a few. These applications range in complexity from industrial systems involving thousands of lines of code to simple simulations aimed at children that only involve a few lines of code. Increasingly, Agentsheets has been used to create a number of interesting educational design environments like the ones described below.

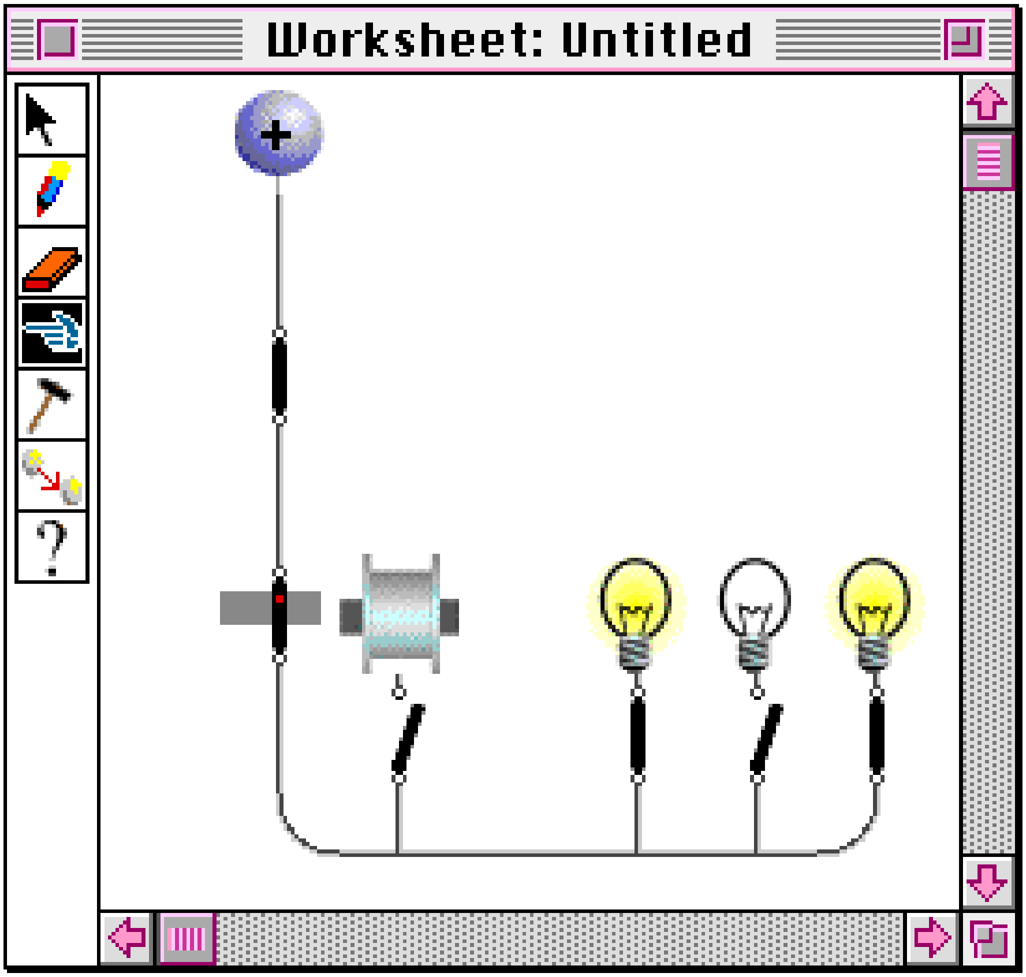


Figure 2. Electric World

Electric World

Electric World (Figure 2) is a design environment that allows the construction and simulation of electrical circuits that are made up of light bulbs, switches, wires, power sources and solenoids. Learners can construct different circuits, turn the power on, and observe what happens. For instance, if a light bulb receives power, it will turn on. Furthermore, switches are active and can be switched on and off. Electric World was developed to provide learners a means to interact and learn about electrical circuits without being exposed to the risk of shock.

Melting Ice

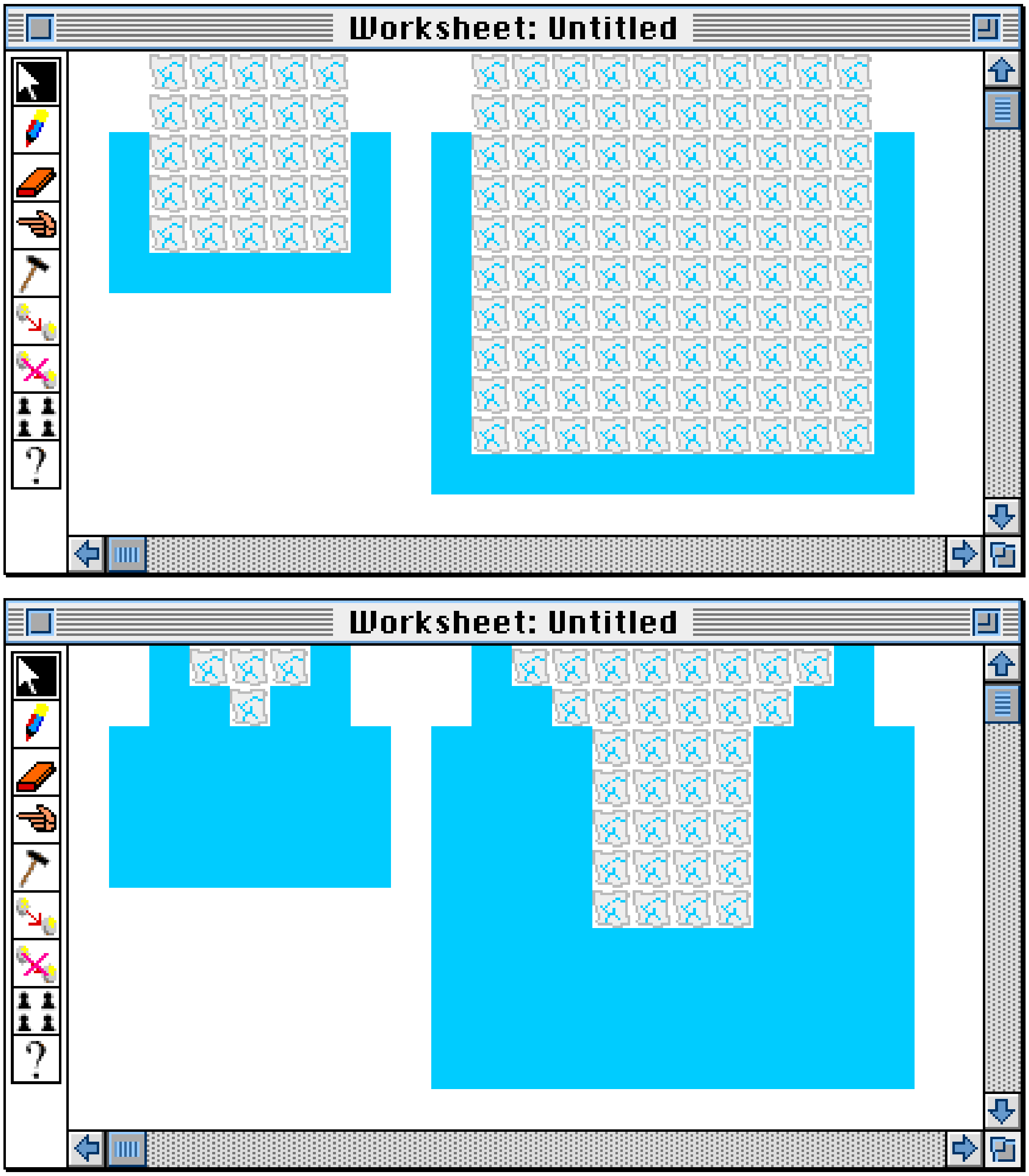
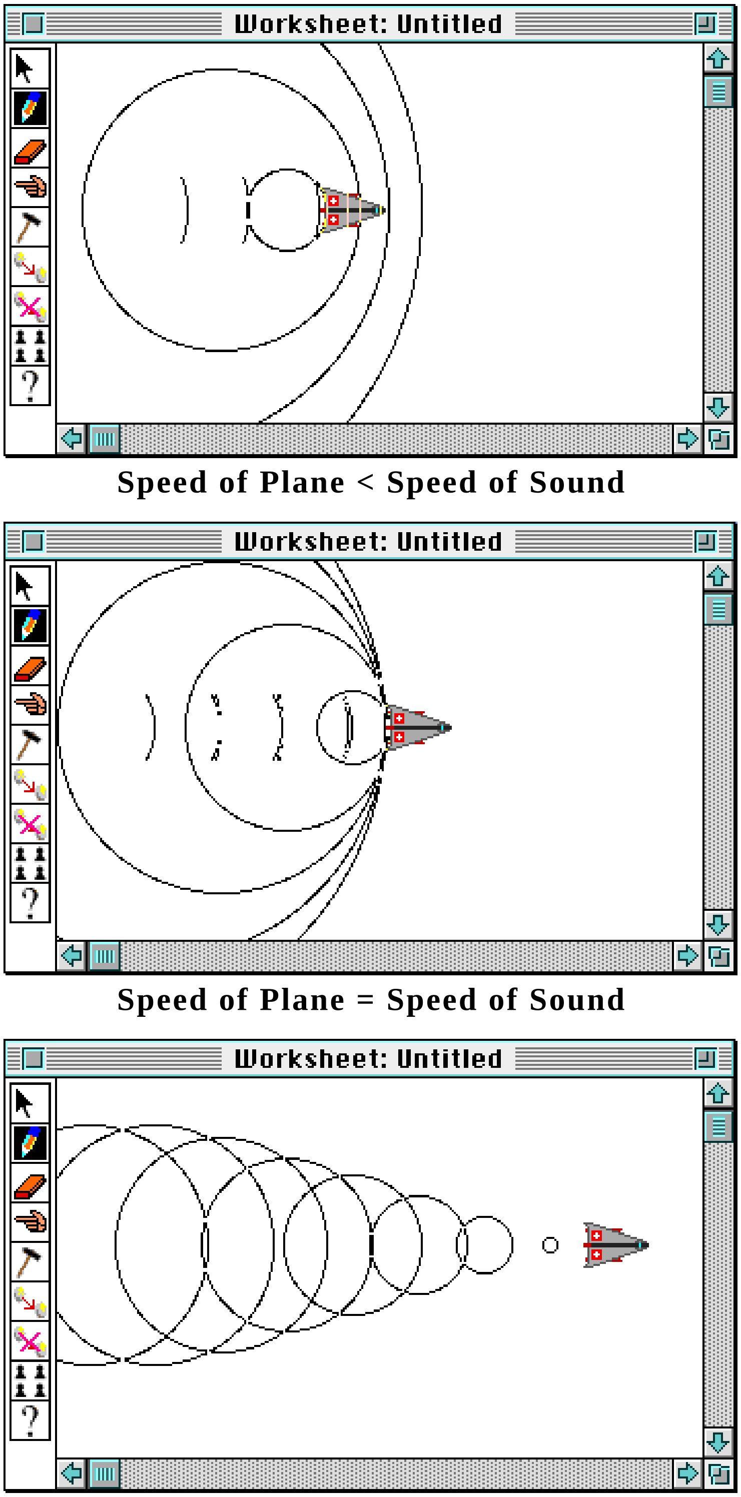


Figure 3. Melting Ice Cubes

The Melting Ice application (Figure 3) was developed after a professor at the University of Colorado reviewed some elementary school science curriculum designed to teach students about how ice melts. The goal of the curriculum was to get students to develop hypotheses about the rate at which ice melts, and then to observe actual ice melting behavior. The professor felt that what was missing from the curriculum was a component requiring students to explain the phenomena behind their observations. To augment the curriculum, the Melting Ice application was developed in Agentsheets to allow students to see how ice melts and gain insight about the process by exploring different cube sizes and shapes. Using Melting Ice, learners can see that melting is a surface operation and that the limiting factor is not the volume of the ice, but the exposed surface area.

Waves



Speed of Plane < Speed of Sound

Speed of Plane = Speed of Sound

Speed of Plane > Speed of Sound

Figure 4. The Wave Environment

The Waves application (Figure 4) was developed by one of the authors, a computer scientist whose wife was studying sound waves and the Doppler Effect. The application shows sound waves emitted by either fixed or moving sources. When a sound wave is generated, its progress is depicted on the screen by a series of expanding circles. The user can set the speed of the moving object, and by running the simulation, interesting sound wave phenomena are easily described. For instance, in the middle picture, the plane is flying at exactly the speed of sound, and the accumulation of sound waves that will make up the sonic boom is easy to see.

EcoOcean

EcoOcean (Figure 5) deals with creatures living in the ocean, such as whales, sharks, and krill. Different types of animals live in various depths of water. The system is used to illustrate predator-prey relationships and the careful balance of nature. It was developed by a graduate student whose four year old daughter was fascinated with whales and sharks.

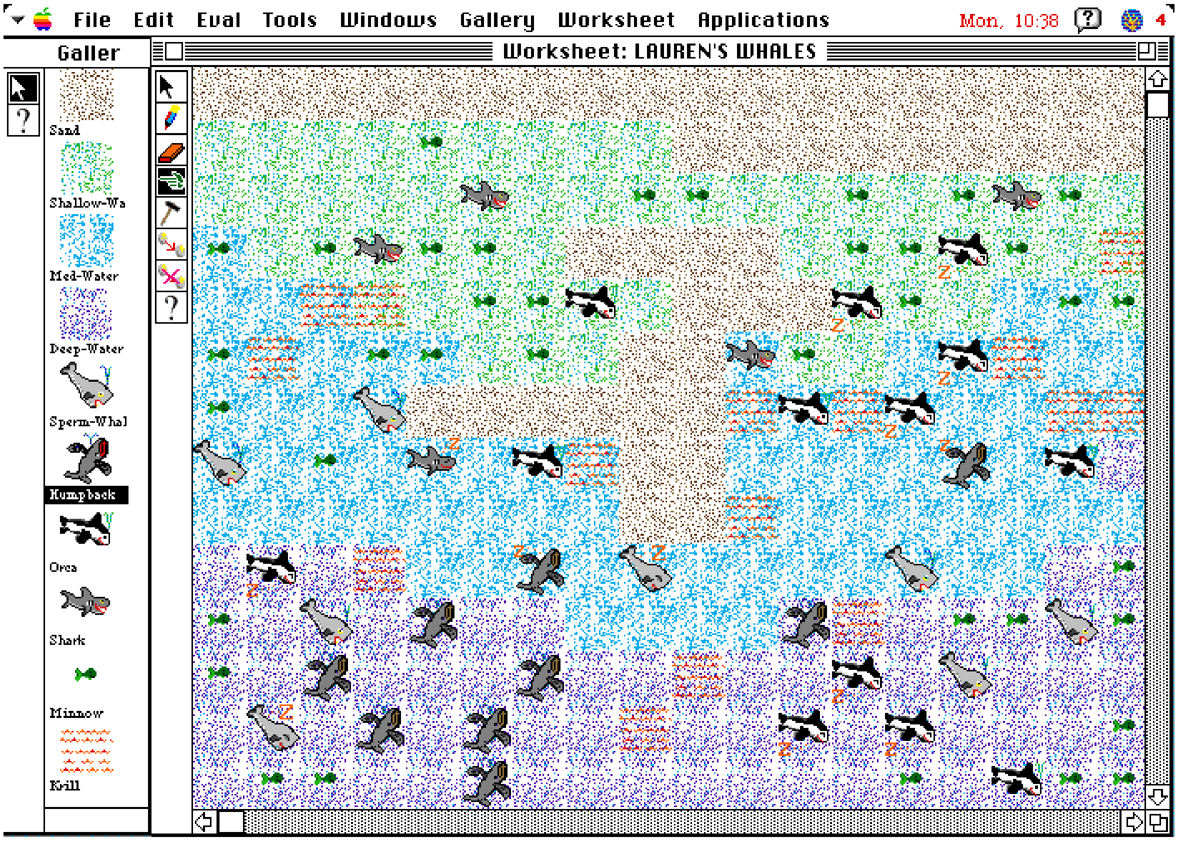


Figure 5. EcoOcean

By populating the EcoOcean, learners can answer questions regarding the life cycles of creatures and can determine how the population of one type of animal can influence other animals. This application also demonstrates how complex “emergent” behaviors can result from the interaction between simple behaviors of more than one animal type. For instance, the daughter noticed that when krill in the deeper waters were scarce, the krill-eating humpback whales would follow their food into shallow water, become stranded, and die. Although the graduate student had programmed the humpbacks with the two separate behaviors of eating krill and dying in shallow water, he had not anticipated that the krill-eating behavior would lead a whale to shallow water. While observing this unexpected behavior, the preschooler created an explanation that the humpbacks were stranding themselves because there wasn’t enough food for them in the deep water. This observation prompted her curiosity, causing her to suggest that this might also explain the mysterious whale strandings that she had heard take place in the natural world.

THE AGENTSHEETS EXPLORATORIUM: COMBINING MOSAIC WITH AGENTSHEETS

The success of our local Agentsheets community was based largely on the ability of end-users to interact directly with application developers. End-users provided application developers with domain knowledge, and application developers, in turn, could point out emergent application behavior to end users. What ensued became a collaborative community of learners, and ideally, the exploratorium could lend similar support to a distributed community.

The ideal environment for a distributed community encompasses the framework established by Pea and Gomez to support distributed multimedia learning environments [10]. Included in this framework is the ability to provide rich information networking that allows easy, reliable distribution of any kind of data including text, pictures, sounds, and movies as well as interactive information. Network access to the Agentsheets system and its applications is not sufficient because important contextual information is omitted. In the Agentsheets Remote Exploratorium, multimedia information is used to supply this context; how an application was developed, how it can be used, and even the motivation for the educational use of the application. For instance, access to the Waves design environment might be useful to someone interested in sound wave phenomena like the Doppler Effect. In addition to the environment, a learner needs a set of instructions for using it as well as a description of the principles that the model illustrates. Moreover, ideal curricular support would consist of tools that empower teachers to locate relevant sites, adapt, and organize retrieved materials to their classrooms [11] (see Stahl, Owen and Sumner ***in this issue***). Using the design environments can prompt a desire for more information, as was illustrated in the EcoOcean example. Pointers to related information available on the WWW, in this case about radar systems, music or sound in general, assist the learning process by opening doors to further exploration. Thus, providing access to such information sources is an important feature of the Agentsheets Remote Exploratorium.

The WWW operates within a client-server architecture that provides the technological infrastructure for the easy distribution of multimedia documents. WWW servers run on remote computers and are responsible for storing and distributing data, while WWW clients run on the users’ local computers and request data from the servers. Clients and servers communicate with each other over the Internet. NCSA Mosaic is one client application that allows hypermedia access to the information available from WWW servers. Mosaic is referred to as a WWW client, or browser, because it is an application that enables access to WWW servers. There are several other WWW client applications available for the Macintosh including Mosaic, NetScape®, and EINet MacWeb® among others.

We decided to base the Agentsheets Remote Exploratorium on the WWW because of its widespread use, its ability to provide access to multimedia information and the ease-of-use of its client applications. We chose NCSA Mosaic as our client application because it is easy to acquire and is the most widely used WWW client application. What was not available in any of the client applications was a means to easily distribute interactive information like Agentsheets applications. Our plan was to combine Agentsheets and Mosaic so that a WWW client could automatically launch Agentsheets locally when a particular Agentsheets application was accessed and loaded from a WWW server. By using Mosaic and the WWW, we created an information rich environment to distribute Agentsheets applications that includes scaffolding and support similar to that already in place for our local community of developers and users.

THE REMOTE EXPLORATORIUM IN USE

A remote exploratorium provides learners access to interactive exhibits. The remote exploratorium WWW client consists of the combination of Mosaic and Agentsheets running on the same machine. This client is used to access a remote exploratorium WWW server via the Internet (Figure 6).

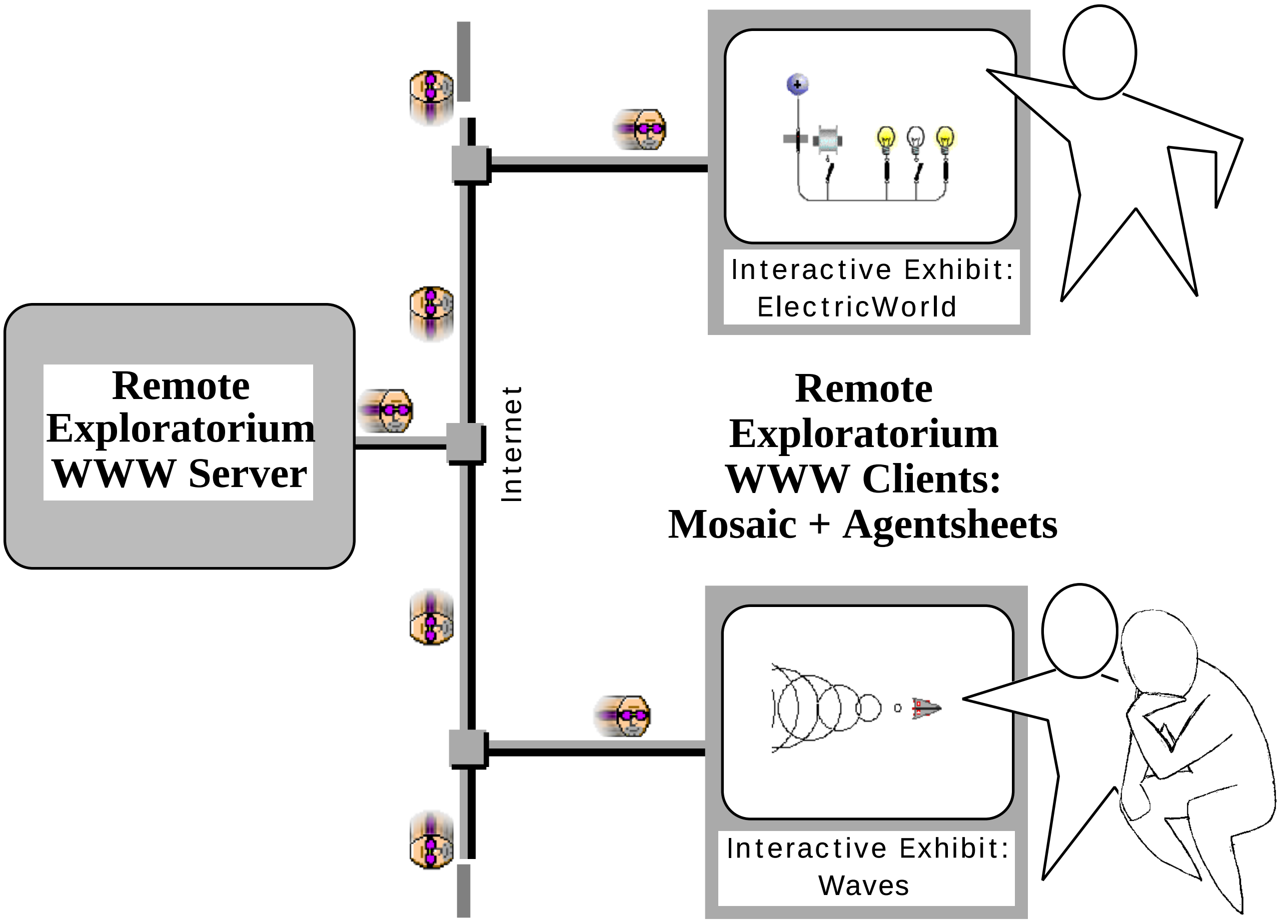
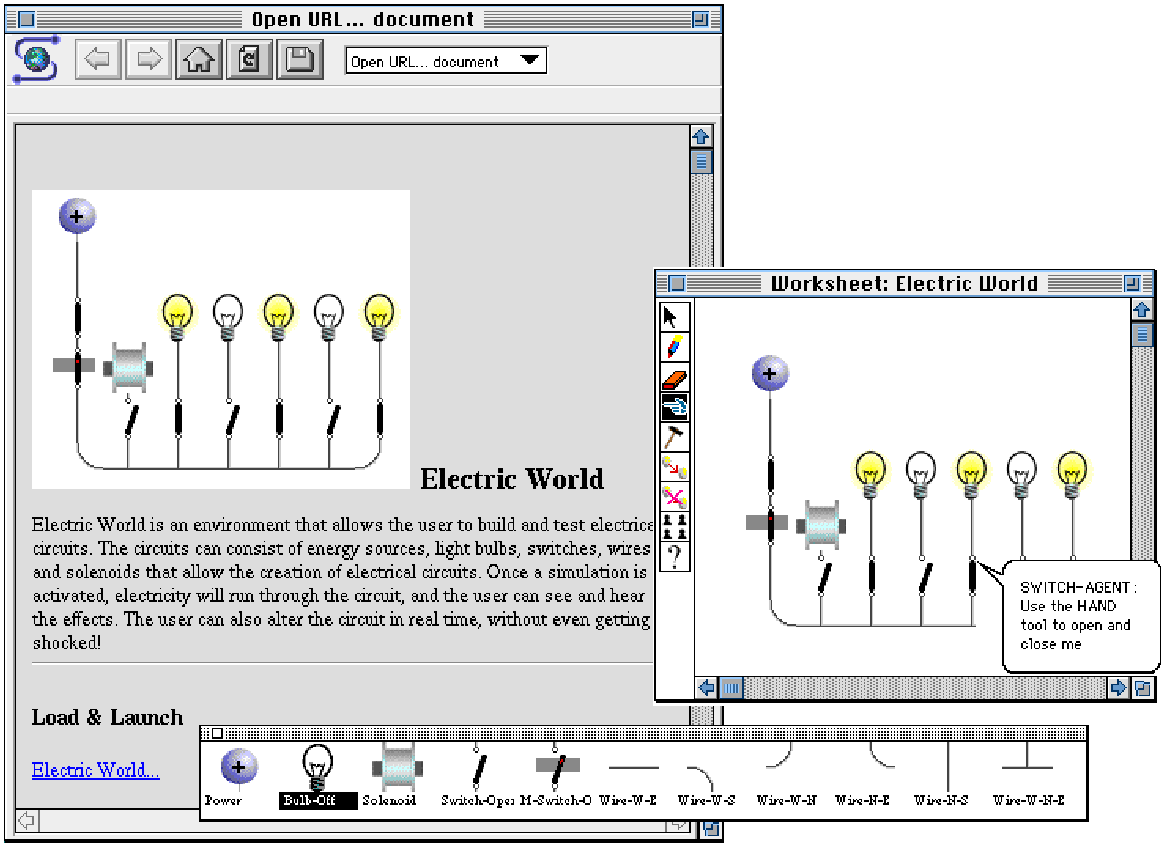

Interactive Exhibit: ElectricWorld
Remote Exploratorium WWW Server
Internet
Remote Exploratorium WWW Clients: Mosaic + Agentsheets
Interactive Exhibit: Waves

Figure 6. Remote Exploratorium: (WWW and Mosaic) + Agentsheets

The combination of network medium and design environment allows a learner to be engaged at four different levels of *interactivity*, outlined below, ranging from simple navigation to adding new components to design environments and defining their behaviors.

1) ***Navigate*** : The learner makes use only of the Mosaic part of the Agentsheets Remote Exploratorium. Mosaic serves as a navigational tool to find interesting related information consisting of text, pictures, sounds and videos.

2) ***Load and Launch Design Environments***: If learners are interested in a deeper understanding of an exhibit, they can load and run the design environment component. For instance, the Electric World interactive exhibit is about electricity. In the exhibit, (Figure 7, left) the learner can click the “Load and Launch” option to access the design environment. In response, the Remote Exploratorium WWW server sends a document containing agents to the Mosaic client. The agents within the document encapsulate not only *information*, such as sounds and depictions, but also *interaction* that specifies how agents interact with users. When Mosaic receives this document, it starts Agentsheets locally, and passes it the document. Agentsheets then loads sounds, installs agent depictions, compiles agent programs, stores agent documentation and launches the design environment. All this takes place in about fifty seconds although performance can vary depending on network traffic and access rates. The learner sees two new windows on the screen: a worksheet in which the simulation takes place (Figure 7 right middle) and a gallery of agents (Figure 7, bottom) containing electrical components. Simple documentation describing the behavior of agents and the means to interact with them can be accessed through Macintosh Balloon Help. The balloons, like the behavior, depictions and sounds of agents have been transferred from the WWW server via Mosaic to Agentsheets running on the client's computer. In the Electric World design environment, learners can operate switches and observe reactions. For example, operating the left most switch in the lower row of switches will put the circuit into a feedback mode in which an electric coil and the electromagnetic switch located left to the coil will interact with each other.



Electric World

Electric World is an environment that allows the user to build and test electrica circuits. The circuits can consist of energy sources, light bulbs, switches, wires and solenoids that allow the creation of electrical circuits. Once a simulation is activated, electricity will run through the circuit, and the user can see and hear the effects. The user can also alter the circuit in real time, without even getting shocked!

Load & Launch

Electric World...

SWITCH-AGENT : Use the HAND tool to open and close me

Figure 7. Mosaic and Agentsheets running Concurrently

3) Construct New or Extend Existing Designs: Learners have all the components to create new designs or to change existing ones. For instance in the design below (Figure 8) the learner has added some switches in a row. By doing this, the learner is able to apply knowledge gained from the exhibit.

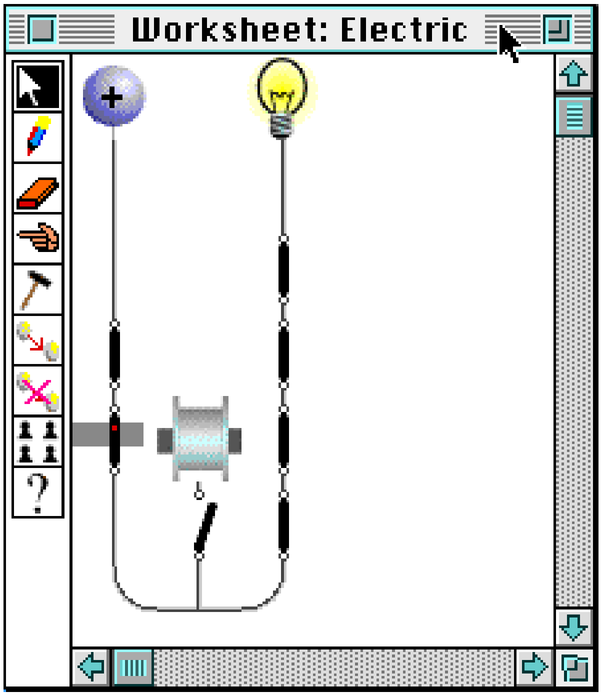


Figure 8. Changed Electric World

4) Extend Behavior of the Design Environment: Learners can change their role from end-users to designers (Figure 1) by using Agentsheets functionality to modify the behavior of the design environment, perhaps by adding custom agents. In the Electric World, a learner introduces a buzzer by first defining its depiction, , using the Agentsheets depiction editor, then defines the behavior of the buzzer either using a textual programming language called AgenTalk or by using graphical rewrite rules [12-14]. The expressiveness and modifiability of a design environment is enhanced with the addition of a programming language [15].

Once the new buzzer agent is defined, it is ready to be used in a design (Figure 9). It serves as a replacement for the bulb. When the buzzer receives a signal it will play a sound. This extensibility allows a design environment to be changed to support what is most relevant to the learner and to reflect shifts in the learner's acquisition of knowledge.

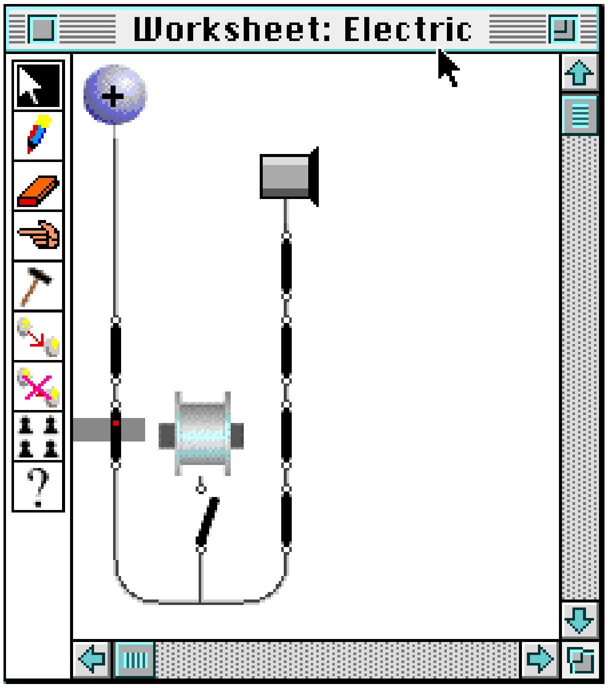


Figure 9. Electric Circuit using Buzzer

EXPERIENCES

One high level goal of our work is to support distance education. To this end, we created the Agentsheets Remote Exploratorium as a mechanism to facilitate the *effortless exchange and distribution* of interactive educational exhibits through the Internet. Although others have successfully used the WWW to distribute computer simulations [16], the Agentsheets Remote Exploratorium, through its ability to distribute agents, provides learners with complete design environments. These environments allow the participatory theater interaction scheme and can offer learner's engaging opportunities on a number of levels.

In this section we describe a number of issues arising from the combination of network media and design environments in general and from the combination of Mosaic and Agentsheets specifically. We briefly outline the technical requirements and challenges to build a remote exploratorium, and also discuss the implications for exploratorium users.

Technical Requirements

Our technical approach consists of two components: remote exploratorium servers and remote exploratorium clients (Figure 6). A remote exploratorium server is a specially configured WWW server. A remote exploratorium client is the combination of Mosaic (a WWW client acting together with the server as a network medium) and Agentsheets (the substrate for design environments).

Providing ready access to a design environment requires the ability to transmit several files containing different types of information from servers to clients. Every design environment constructed in Agentsheets has its own mix of graphical depictions, sounds, behavior, and help information, all of which is contained in several files. In order to download the desired files and launch Agentsheets with minimal user intervention, it was necessary to turn Agentsheets into a Mosaic Helper Application.

Mosaic uses a number of platform-specific Helper Applications to show various types of multimedia information. For instance, if a user accesses a video from a Web page, Mosaic will automatically download the video to the user's machine and start the Helper Application that knows how to show videos. Mosaic itself cannot show videos; it passes the video file to the application that can. In order to combine the browser capabilities with the functionality of a design environment, we enabled Agentsheets to be a Mosaic Helper Application.

Design environments incorporate a number of different files required by Agentsheets. These files contain: galleries specifying agent depictions (Figure 1 (1)), agent behavior (Figure 1 (5)), sound resources, and worksheets (Figure 1 (2)). The current configuration mechanism in Mosaic only allows the mapping of one file to a Helper Application does not support this.

We defined the **A**gent**S**heets **C**ompound **D**ocument (ASCD), a single file containing compressed versions of all of the files needed for an Agentsheets design environment. In order to allow Agentsheets to read ASCDs, we created a new document handler within Agentsheets to open an ASCD, decompress its contents and extract each of the individual files. These files are then opened and loaded.

Once an ASCD is downloaded from the exploratorium server, Mosaic launches Agentsheets and passes it the ASCD. The Agentsheets document manager then takes over, and launches Stuffit Expander$^{®}$ to decompress the ASCD into a folder containing all of the necessary files. Expander is a freeware tool commonly used to perform compression and decompression tasks on Macintoshes. Once the ASCD is separated into its components, Agentsheets opens the gallery, compiles the agent behavior files, and opens worksheets serving as design playgrounds.

Issues for Exploratorium Users

A number of problems ranging in severity from bothersome to catastrophic need to be addressed to turn remote exploratoriums into an effective distance education mechanism. Users having to deal with these problems can be roughly categorized into:

- ***Explorers***: users who are interested in using interactive exhibits. Explorers access remote exploratorium *servers* via remote exploratorium *clients* (Figure 6).
- ***Providers***: users who would like to share an interactive exhibit with other users. Providers have to maintain remote exploratorium *servers.*

The two categories of users are not exclusive. That is, at times providers can be explorers and vice versa. The problems described illustrate the limitations of the current Mosaic presentation-oriented paradigm, and raise issues relevant to providers, explorers and the entire WWW community.

Reliability: "<Connection Refused by Host>"

A crucial aspect for the success of a remote exploratorium involves the reliability of access to the information. This is especially true when supporting remote communities of learners. Imagine if a student was learning from a text book, and that text book would periodically disappear with no warning. This kind of situation is unacceptable, and needs to be addressed.

A major concern with the use of networked applications in schools is the reliability of the networking technology [17], and we found that the more users our WWW server attracts, the less reliable it becomes. If the Exploratorium is accessed during peak server usage, it is not uncommon for a user to receive a "Connection Refused by Host" error message. Changes within the network media can enhance reliability, but there are changes that we can make as providers to increase reliability as well. For instance, better hardware for the WWW server could help as well as the duplication of the Agentsheets Remote Exploratorium on several servers. Duplicating the information to "mirror" servers provides relief but causes overhead for explorers and providers. Providers need to maintain not just one but several mirror sites and keep them consistent. Explorers, on the other hand, need to be aware of the locations of mirror sites. Furthermore, explorers have no automatic mechanisms that would allow them to transparently access the server with the smallest user load. Additional features of the networking substrate could make this a more viable solution.

Configuration Complexity:

To create a remote exploratorium, communication between the design environment substrate and the network media must be established so that the network media can understand what type of information is to be delivered to the design environment substrate. Establishing this communication becomes an issue for providers because not only do they have to define a new file type, they have to communicate that new type to the network media which in turn, must understand what the new type is. Furthermore, explorers have to ensure that their client applications are configured correctly to handle the new file type. Currently, this configuration overhead can be significant for providers and explorers and prevent them from buying into this technology. An additional problem is raised on a more global level of the network community. New file or media types required by remote exploratoriums need to be standardized to avoid global incompatibility and conflicts.

The WWW establishes this kind of communication through the definition of MIME types. MIME stands for **M**ultipurpose **I**nternet **M**ail **E**xtensions, and a MIME type is associated with every file available on the WWW. When Mosaic receives a file from the WWW, it looks at the file's MIME type to determine what application should be launched. For the Agentsheets Remote Exploratorium, we defined a new MIME type called ASCD. In order to define the new type, we had to alter a file on our WWW server to indicate the definition of the new MIME. The creation of a new MIME type was a curious process, because although modification of the file on our server was not a cumbersome task, determining that this server configuration change was necessary was not readily available information within the documentation provided on the WWW.

Explorers have to configure their client applications to be able to handle the new MIME type, and this process is different for every WWW client application. Motivation for obtaining and exploring any application decreases when acquisition and set-up require many complicated steps which are unsupported and may fail. Currently, an explorer must go through several difficult steps to set up new Helper Applications and MIME-types on the WWW client to obtain Agentsheets simulations. Providers have to support explorers with instructions. We address this issue by including different setup instructions for different WWW clients (Mosaic and NetScape) in the Agentsheets Remote Exploratorium. The automation of MIME type creation and reading that allows a check for the appropriate viewer and supplemental applications, and then automatic installation of a new Helper Application and MIME type would be useful. In fact, any simplification or automation of such a procedure for the explorers is necessary for the continued growth of a distributed community.

There is also no systematic global verification of MIME types. If the ASCD MIME type is created on multiple servers for different Helper Applications, there is no central registry or standardization process to determine if ASCD has already been defined for a different Helper Application on a different server. Users may not be familiar with the configuration of their client applications, and a MIME type conflict would be extremely confusing. Automation of such a registry process would prove useful to insure that new MIME types have unique names.

Efficiency

The efficiency and ease with which the network could distribute files became a major issue in the development of the Agentsheets Remote Exploratorium. Explorers have low tolerance for files that require minutes to be downloaded successfully, and as providers, we were forced to define more efficient file structures that could be distributed over the WWW more quickly. Although we were able to increase efficiency, our solution added complexity to the Agentsheets Remote Exploratorium client environment. If the networking media could automatically provide this increased efficiency, then client environments could be made more simple.

As mentioned earlier, it was necessary to aggregate and compress multiple files. Agentsheets does not automatically perform file aggregation and compression, we had to make use of Stuffit Expander to perform these services. Thus, in order for a client to use the Exploratorium, they require the Expander application.

We asked four different WWW users located around the country to download some of our compressed and uncompressed files and report back the times to us to determine whether compression was really necessary. We found that the compression made at least a fifty percent increase in the efficiency of file transfer, and without compression the transfer rates were at times prohibitive.

Efficient file transfer is not a luxury but a necessity, and we implemented compression and file aggregation at the cost of complexity to explorers. Although Stuffit Expander is a commonly used Macintosh application, any requirements that come between explorers and access to environments can cause problems. If the network media can provide this efficiency automatically, it would remove additional barriers to the creation of distributed communities.

CONCLUSIONS AND FUTURE WORK

This paper has elaborated our experiences creating the Agentsheets Remote Exploratorium, which we envision as a tool for supporting distance education. Although there are still problems barring the overall effectiveness of the exploratorium, we believe that by combining Agentsheets with Mosaic we have enhanced existing network media by providing a crucial missing interactive component. In doing so, we have uncovered issues that will affect the formation of any distributed community of learners.

The addition of interactive exhibits to existing network media helps to effect a change similar to the change from museums to exploratoriums. There is much yet to be done beyond the issues of reliability, communication, and efficiency outlined in this paper, and in our future work we will define and address these new issues. One area we hope to make contributions to is distance collaboration. Although the Agentsheets Exploratorium provides users ready access to interactive exhibits, it does not yet enable users to easily add their own exhibits. Two-way communication of this nature will allow not only the development and exploration of shared artifacts, but a forum for their distribution as well, adding another facet necessary for supporting distance learning.

ACCESSING THE AGENTSHEETS REMOTE EXPLORATORIUM

The Agentsheets Remote Exploratorium can be accessed for viewing at the following URL: *http://www.cs.colorado.edu/~corrina/agent_remexp.html.* In order to explore the exhibits, the Agentsheets Player is needed. Questions regarding this WWW site can be addressed to *corrina@cs.colorado.edu.*

ACKNOWLEDGMENTS

We wish to thank Gerhard Fischer and the Center for LifeLong Learning and Design, Clayton Lewis and Jim Sullivan for all of the thoughtful discussions. We want to thank Tim Philip for his implementation work, and Markus Stolze, Tamara Sumner and Stevan Kalmon for their provocative and timely feedback. This material is based upon work supported by the National Science Foundation and the Advanced Research

Projects Agency under Cooperative Agreement No. CDA-940860. It has also been supported by the National Science Foundation under grant number RED925-3425.

REFERENCES

1. Fischer, G., “Domain-Oriented Design Environments,” in *Automated Software Engineering*. Kluwer Academic Publishers, Boston, MA. p. 177-203 (1994).

2. Papert, S. and I. Harel, eds. *Constructionism*. Ablex Publishing Corporation, Norwood, NJ (1993).

3. Repenning, A., “Agentsheets: A Tool for Building Domain-Oriented Visual Programming Environments,” in *Proceedings of the INTERCHI ‘93, Conference on Human Factors in Computing Systems*. ACM Press, Amsterdam, NL, p. 142-143 (1993).

4. Laurel, B., *Computers as Theater*. Addison-Wesley Publishing Company, Reading, MA (1993).

5. Papert, S., *Mindstorms: Children, Computers and Powerful Ideas*. Basic Books, New York (1980).

6. Toffoli, T. and N. Margolus, *Cellular Automata Machines*. MIT Press, Cambridge, MA (1987).

7. Nardi, B., *A Small Matter of Programming*. The MIT Press, Cambridge, MA (1993).

8. Repenning, A. and T. Sumner, “Programming as Problem Solving: A Participatory Theater Approach,” in *Proceedings of the Workshop on Advanced Visual Interfaces*. Bari, Italy, p. 182-191 (1994).

9. Repenning, A. and T. Sumner, “Using Agentsheets to Create a Voice Dialog Design Environment,” in *Proceedings of the ACM/SIGAPP Symposium on Applied Computing*. ACM Press, Kansas City, MO, p. 1199-1207 (1992).

10. Pea, R.D. and L.M. Gomez, “Distributed Multimedia Learning Environments: Why and How?,” *Interactive Learning Environments*, **2**(2), p. 73-109 (1992).

11. Stahl, G., R. Owen, and T. Sumner, “Share Globally, Adapt Locally: Software Assistance to Locate and Tailor Student-Centered Curriculum Posted to the Internet,” *Computers and Education*, (1995).

12. Repenning, A. and T. Sumner, “Creating Domain-Oriented Visual Languages: A Real-Time Collaborative Approach,” *submitted to IEEE Computer, Special Issue on Visual Programming*, (March) (1995).

13. Smith, D.C., A. Cypher, and J. Spohrer, “KidSim: Programming Agents Without a Programming Language,” *Communications of the ACM*, **37**(7), p. 55-67 (1994).

14. Bell, B. and C. Lewis, “ChemTrains: A Language for Creating Behaving Pictures,” in *Proceedings of the IEEE Workshop on Visual Languages*. IEEE Computer Society Press, Bergen, Norway, p. 188-195 (1993).

15. Eisenberg, M. and G. Fischer, “Programmable Design Environments: Integrating End-User Programming with Domain-Oriented Assistance,” in *Proceedings of the ACM CHI Conference*. ACM Press, Boston, MA, p. 431-437 (1994).

16. Smeaton, C. and A. Slater, “Integrating Simulations and WWW Courseware,” in *Proceedings of the 2nd International WWW Conference: Mosaic and the Web*. Chicago, IL., p. 459-465 (1994).

17. Koschmann, T., D. Newman, E. Woodruff, R. Pea, and P. Rowley, “Technology and Pedagogy for Collaborative Problem Solving as a Context for Learning: Report on a CSCW ‘92 Workshop,” *SIGCHI*, **25**, p. 57-60 (1993).