

Scripting Online Social Interaction

Harry Chesley, Lili Cheng, Shelly Farnham, Jennifer Landau, Suzana Seban

8/9/2001

Technical Report
MSR-TR-2001-74

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Scripting Online Social Interaction

Harry Chesley, Lili Cheng, Shelly Farnham

Microsoft Corporation

One Microsoft Way

Redmond, WA, 98052

1-425-882-8080

{harrych, lilich, shellyf}@microsoft.com

Jennifer Landau, Suzana Seban

Hammand & Landau

1697 Oak St.

San Francisco, 94117

jhlandau@earthlink.net,

suzana@network172.com

ABSTRACT

Scripting of social interaction is commonplace in off-line systems. On-line, this technique can be even more effective, using the computer as mediator. To investigate this area, we implemented a prototype system named Lead Line that adds scripting to plain text chat. Using Lead Line, we ran a number of informal trials and conducted an experiment in structured versus unstructured interaction in a goal-directed task. We learned a number of things, some confirming previous results, and some new: brainstorming and assessment tasks work well when scripted; lessons users learn from a script persist; structure enables creativity; familiar contexts can enable communication; identity is a tool; and roles can empower the user.

Keywords

Social computing, computer-supported cooperative work

INTRODUCTION

Increasingly, social interaction is taking place online. In part, this is a consequence of the sheer numbers of people using online systems (over 300 million as of March 2000 by one estimate[19]). In part, it is because people are finding online social tools such as email that allow them to communicate and maintain their social circles more effectively than traditional offline technology does.

Online social systems tend to suffer from two problems: lack of context and poor scalability. The first is a result of the nature of computer networks, where any two points are, or can be, adjacent. The second is a result of transplanting offline techniques to online systems without making use of the extra capabilities added by their programmability. Both of these issues can be addressed by the use of scripting in the design of online social systems.

In the offline world, social interaction always takes place within a shared and well-known context: a preplanned meeting in a conference room, a commercial transaction in a store, an impromptu visit to an office, a dinner party at a home, etc. The context reduces the amount of time spent

determining the goal of the interaction and introducing the participants. It also allows those involved to pick an appropriate model to structure the interaction. Mailing lists, email threads, and hosted communities all serve the purpose of providing online context. Lack of context is a primary reason that random real-time interactions in chat are so disappointing. With no common context, only the most generic and boring models for social interaction apply.

Social interaction is most effective when guided. In the offline world, this is most often accomplished by a moderator, either informally, as in the case of a manager at a business meeting, or formally, as in the case of a professional meeting facilitator at a team-building offsite. While mapping this approach to online systems is very effective in some cases, it is inherently unscalable, requiring the addition of scarce human resources for each increment in online population. While they will never be as effective as a live human, computer-based systems are capable of adding an element of facilitation that is extremely scalable.

We believe that an important piece in solving both the context and scalability issues of social systems is the addition of scripting to online social interactions. Scripts also provide a means of incorporating pre-designed, high-quality interaction schemas: best practices in the business environment and best sellers in the entertainment field.

In order to explore scripted social interaction, we have built a simple text chat system with pre-authored scripting, called Lead Line. We have explored the use of this system for both business and entertainment purposes. And we have conducted a more formal experiment (the first in a planned series) that compares structured versus non-structured interaction in a goal-directed task.

The results of our work confirm several existing findings – that brainstorming works well online, that controlling the degree of identity versus anonymity can significantly affect how openly people interact online, and that properly designed constraints can improve creativity. They also bring into question some existing results – that structured systems are overly constraining and subsequently unpopular. And we found that assessment, like brainstorming, is well suited to online interaction, and that lessons learned in a structured session carry over to subsequent unstructured ones.

SOCIAL SCRIPTING

Offline, scripted social interaction plays a part in systems designed to achieve articulated tasks. It is a large part of meetings, training and development, and knowledge sharing in contemporary organizations, as well as in the classroom.

Scripting is intentional and often even recognized, although it is sometimes traditional and anonymous. For example, an agenda for a meeting is a widely accepted and obvious script. And people take for granted innumerable every-day scripts for things like buying an item at a store or conducting an interview.

In strategy and decision meetings, real-time facilitation systems such as Open Space Technology[23], Scenarios/Future Search Conferences[34], and the Strategic Forum[25] are used to guide how people interact and subjects are explored. These systems provide structure to the social interaction to focus it and make it more effective.

Job training, teambuilding, and corporate learning centers are all specific environments that intentionally build on social interaction in order to complete the task[16]. SimuReal[15] and Gemba Kaizen[14] are two of many authored approaches that use experiential methods and strategic thinking to both train the individual and change the overall organization. Board games, outdoors ropes courses, and on-line simulations are all means through which social interaction and job skills are developed using scripts. Approaches like Bohmian dialogue[2], Think Like a Genius®[31,32], and The World Café[4,5] provide a focus on business tasks by structuring social interaction. In classrooms, teachers utilize collaborative structured activity, small-group dialogue, and role-playing for academic subjects as well as social objectives[27].

EXISTING ONLINE SYSTEMS

The most common activities on computer networks are information sharing, information publishing, and email, followed by document management, electronic forms, corporate directories and policies, and workflow management[13]. As more person-to-person interaction takes place online, there are increasing opportunities for computer-mediated structuring of those interactions. The study of group technology, especially in business environments, and including structured systems and techniques, is known as computer supported cooperative work (CSCW), and has been active since the 1980s[12].

Workflow Systems

Many workflow systems are designed based on speech act and/or situated action theory. According to Winograd and Flores, "... language cannot be understood as the transmission of information. Language is a form of human social action, directed towards... 'mutual orientation.' This orientation ... exists as a consensual domain – as interlinked patterns of activity. The shift from language as description to language as action is the basis for speech act theory..." [Winograd in 26]

Suchman pioneered a related theory, the situated action perspective: "plans are resources for situation action, but do not in any sense determine its course." [Suchman in 29] This theory "brings forth the situatedness of human work in terms of the community of practice cooperating on a common task, of the work-space they live in, of the experience they share..."[17]

The Coordinator

The Coordinator is an email system based on speech act theory and designed with conversation as a unit of work. "Instead of providing a uniform command to initiate a new message, the Coordinator provides options that identify different linguistic actions such as a request that can be promised, counter-offered, declined, cancelled, or completed..."[10]

Reaction to the Coordinator has been mixed. Flores observes that the Coordinator has been most successful in organizations in which the users are comfortable with their role in the organization[10]. Dur also found that in a strict company like EDS, the Coordinator was a "good communication tool and it gets the job done by enforcing compliance." [Dur in 26] In another study at Pacific Bell, workers stopped using the system and complained that it was "unnatural," "uncomfortable," and "made no sense." [Grantham in 26, 6] found that users thought the system opened communication within their companies, but through use of its email and messaging capabilities rather than its linguistic actions. Overall, the imposed structure led to fears of a "fascist computer system." [26]

Lead Line focuses on separate linear sessions of interaction. It does not attempt to structure all interactions between users or maintain multiple simultaneous transactions. However, we found that acceptance of the structuring depended greatly upon the design of the scripts. Some were seen as over-constraining by users, while others, just as structured, were seen as helpful and even liberating. Unlike the Coordinator, Lead Line does not attempt to distill a generic set of human interactions, but leaves it to the author to structure a specific interaction at a higher level of granularity. This allows the author to be more sensitive to the potentially constraining nature of scripting.

Group Decision Support Systems (GDSS)

GDSS's are systems that attempt to structure a group decision process in some way, sometimes through decision models. Generally, a human facilitator controls the decision process. Overall, GDSS's focus group members toward the task, increase the quantity of effort put in the decision process, and increase reaching of consensus[24].

GroupSystems

GroupSystems is a groupware technology with several components: a categorizer to allow users to enter ideas, a group outliner for subtopics, an electronic brainstorming tool, and multiple criteria voting methods. Nunamaker et al performed experiments to make group meetings more

productive. They focused on synchronous interactions in a meeting room set up to use GroupSystems with workstations and a video display as an electronic blackboard, to provide tools for session planning and management, group interaction, organizational memory, individual work, and data collection[20]. These tools supported both a facilitated style of meeting, and an interactive style using parallel, anonymous communication. Results varied depending on the group and situation. Nunamaker found that parallel communication promoted more input and less dominance by a few people, and anonymity allowed issues to be discussed more candidly.

Lead Line also found that parallel, anonymous communication made brainstorming more effective. And we found that the use of a guided, or facilitated approach was very effective. Lead Line uses a facilitatorless, computer-based script approach to providing this guidance.

MUDS and Virtual Worlds in Education

Multi-User Dungeons, or MUDs, are place-based, multi-user networked environments with roots in role-playing games of the late 1970s. MUDs represent a world of interconnected rooms, populated by active objects and avatars representing users. Actions and descriptions are displayed using text[18]. Virtual worlds are also place-based, but usually have a 2D or 3D graphical interface. There are a number of different MUD systems in use. We discuss two below that are used specifically for education.

Pueblo

Pueblo is an elementary school-centered learning community supported by an object-oriented MUD, or MOO. Pueblo emphasizes collaboration and mentoring between students, teachers, and senior citizens, and is based on the theory of informal, situated learning[21]. To manage collaboration, Pueblo's designers considered creating a technical tool but instead used a social mechanism, text-based chat, for floor control. To translate traditional power structures into Pueblo, the designers developed special types of avatars and badges that users could wear to display their status. O'Day concludes that both established authority structures and technical issues should be considered in the design of community.

Zora

Zora is a virtual world designed to help young people explore their identity and values by building artifacts as representations of themselves and creating an online community[1]. Zora supports the creation of a city with spaces and interactive objects, and encourages introspection about role models and values. It is based on a constructivist educational philosophy that says people learn better when they are engaged in building an external artifact. In Zora, students can create their own artifacts with attributes to structure a way of thinking about them: appearance, description, values, conversation, stories, ownership, and permissions. Bers found that Zora was a powerful tool in

exploring identity and values through creation of artifacts, story-telling, and social interaction.

Lead Line draws on the social and place-based aspects of MUDs and virtual worlds, but not on the object authoring capabilities. Like a MUD, Lead Line uses text to immerse the user in a particular role and place. Unlike a MUD, Lead Line sessions follow a simple, linear script.

LEAD LINE

In order to explore the use of on-line scripting in social interaction, we built a prototype called Lead Line. The prototype adds pre-authored scripting to simple text chat. We used text chat for three reasons: First, it was the simplest to develop and to write scripts for. Second, it allowed the work to focus on the social aspects rather than on the media. Third, it was the easiest to record for later analysis.

Lead Line sessions are divided into *scenes* over time. During the session, each user takes on a *role*. Authors can give different roles different instructions in each scene. Advancement from one scene to the next is primarily under the control of the users. Authors can require that a set of users must all agree in order to go to the next scene, they can require that any one user of a set decide to go on, or they can use timed scenes that continue to the next one unconditionally after a set length of time.

This relatively simple structure provides for a considerable range of potential structures. Authors can provide cooperative instructions to users; or they can provide conflicting, or even contradictory instructions. Users can be required to cooperate to complete a scene, or one user can be given control over the others. By having the users decide when the criteria for scene advancement has been met, those criteria can be made very abstract – something that is not possible in more mechanical, computer-driver systems.

User Interface

Figure one is a screen shot of a Lead Line session.

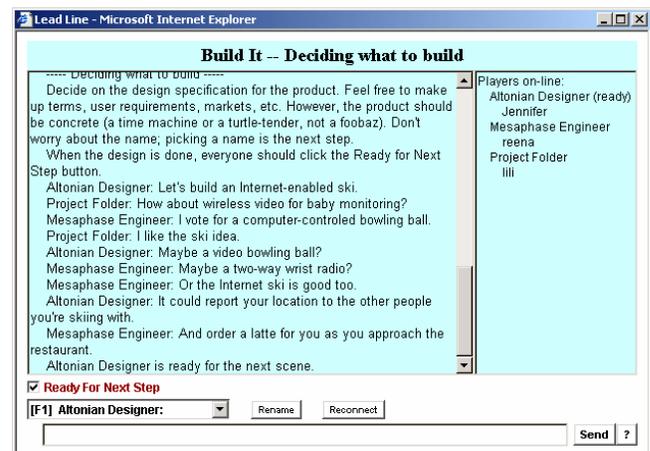


Figure 1: Lead Line User Interface

The Lead Line UI is divided into several distinct areas:

Title The top of the window contains the title of the script concatenated with the title of the current scene.

Chat History The largest section of the window, on the middle left, is the chat history. This contains a complete transcript of this and other users' inputs. It also includes the per-scene instructions given to the user by the script author.

User Input Below the chat history is the user input area. This includes a text entry field as well as a pop-up list of formats. Given that user "Fred" types in "xyz", he can choose from the following formats:

```
Fred: xyz
Fred thinks: xyz
Fred shouts: xyz
Fred xyz
xyz [Fred]
```

Scene Advance Checkbox Between the chat history and the text input area is the scene advance checkbox. This is where the user specifies that the scene advance criteria have been met and it is time to move on to the next scene. Depending upon the script, clicking this checkbox may be sufficient to cause the script to move on, or other users may need to click on their checkboxes as well.

User List On the right side of the window is a list of roles and the names of the users who are playing those roles. If a user has clicked on his/her scene advance checkbox, the text "(ready)" appears after her/his name in the list.

Other Above the text input area are two buttons: *Rename* allows the user to change his or her name. *Reconnect* allows the user to force a reconnect to the server in the event of connectivity problems. At the bottom right, a "?" button invokes a help window with information about Lead Line and about the particular script being run.

Earlier versions of the Lead Line user interface were more complex, including separate areas for scene, script, and role backgrounds. Usability tests showed that users only focused on one area. Once the text history started to scroll with new input, any attention initially paid to the other areas was withdrawn and never returned. As a result, users often missed important instructions placed in these areas. Based on this, we redesigned the interface to place all of the information in the text history, whether it comes from the script or from other users.

XML Script Format

Lead Line scripts are written in an XML tagged format[3]. This allows authors to create scripts with a simple text editor, in a format that is at least partially familiar to anyone with HTML experience, and that is extensible to allow for later changes to the prototype. The following is a simple example of a Lead Line script:

```
<!--Script-->
<!--ScriptTitle-->Debate
</!--ScriptTitle-->
```

```
<!--RoleInfo role="Pro"-->
<!--RoleInfo role="Con"-->
<!--RoleInfo role="Audience" min="0" max="4"-->
```

```
<!--Scene advance="all" participants="Pro; Con"-->
<!--SceneTitle-->Pick a subject</!--SceneTitle-->
<!--SceneBackground role="Pro"-->Pick a subject to
debate. You'll be arguing for it.
</!--SceneBackground-->
<!--SceneBackground role="Con"-->Pick a subject to
debate. You'll be arguing against it.
</!--SceneBackground-->
<!--SceneAdvanceButtonName-->Click here when
you've decided on a subject
</!--SceneAdvanceButtonName-->
</!--Scene-->
```

```
<!--Scene advance="10 minutes"-->
<!--SceneTitle-->Pro Arguments</!--SceneTitle-->
<!--SceneBackground-->For the next ten minutes, the
Pro side will argue his or her point.
</!--SceneBackground-->
</!--Scene-->
```

```
<!--Scene advance="10 minutes"-->
<!--SceneTitle-->Con Arguments</!--SceneTitle-->
<!--SceneBackground-->For the next ten minutes, the
Con side will argue his or her point.
</!--SceneBackground-->
</!--Scene-->
```

```
</!--Script-->
```

It is helpful in understanding this script to walk through it once from the perspective of the Pro role: At the start of the session, he/she sees the text "Pick a subject to debate. You'll be arguing for it." in the chat history, plus a scene advance checkbox with the text "Click here when you've decided on a subject." Following a discussion of possible topics of debate, she/he clicks on the scene advance checkbox. When the user taking the Con role also does so, new text appears: "For the next ten minutes, the Pro side will argue his or her point." For ten minutes, the user then argues his or her point. Last, "For the next ten minutes, the Con side will argue his or her point." is seen in the history and the other user takes their turn arguing his/her side. After that, the script ends.

Most of the tags allow an optional "role" parameter that specifies which role or roles they apply to. Multiple role names can be supplied in a semi-colon-separated list. This allows authors to include different specifications for each role. The most common use of this is to provide different <!--SceneBackground--> sections for different roles.

IRC Protocol Additions

We implemented Lead Line as a distributed client, using Internet Relay Chat (IRC)[22] for communications. This approach has two advantages: First, all of the code for Lead Line is in the client, simplifying the design and debugging of the prototype. Second, existing IRC servers can be used, with no special installations or modifications.

IRC provides facilities for client-to-client text messages, but not for separate, out-of-band data. In Lead Line, we use the same channel for user-to-user messages and machine-readable client-to-client messages. Client messages are prefaced with an escape character, a tilde (~). The client strips these messages out of the stream being displayed to the user and interprets them.

The client connects to the IRC server using a random nickname made up of a randomly selected nine-digit number. User-readable user names are maintained within the client and are not used as the IRC nickname. This allows us greater freedom to change names in mid-stream, to have duplicate user names if desired, and to format the user-readable messages in a variety of forms.

The Lead Line client protocol operates in two separate phases: first to find a set of users willing to fill the required roles of a script; and second to coordinate scene advancement as the script proceeds.

Finding Players When users first connect to the Lead Line system, they choose a script to run. The client reads the script and parses it. From the script, it determines the list of available roles, which are displayed as checkboxes that the user can select to state that they are willing to play that role. The clients then determine when a sufficient set of roles has been selected to start the script.

For each script, there is a single unique IRC channel that is used to coordinate the role selection process, and also to allow users to chat with each other until a sufficient set of roles have been filled. Whenever a user checks or unchecks a role selection checkbox, a message is sent on the channel to notify all of the clients. Thus, all of the clients are aware of the state of all of the other clients in terms of the role selection requests made by their users.

In order to simplify the role selection process, one of the clients on the channel is considered to be the *master*, based on the client's nickname. The master takes into account the role requests of all of the clients, giving preference to clients that have been waiting the longest – first come, first served. Once the master has found a set of clients that fill all of the required roles, it sends a message to the channel listing the clients and the roles assigned to them, plus the name of a new channel. All of the listed clients then leave the current channel and join the new one, which is used during execution of the script. The role selection process is then repeated for the next set of users waiting to play, with each new set of users being assigned a unique channel.

Coordinating Scenes Once play has started, all of the clients are considered equal – there is no *master* during this phase. The primary task is now to remember what the current scene is and when to advance to the next scene.

All of the clients start out synchronized on the first scene. However, it is possible for them to become unsynchronized in the event of an unplanned disconnect – a dropped dial-up line, for example. In that case, the client reconnects to the server, but the other clients may have continued on to a new scene in the meantime. To resynchronize the current scene, a reconnecting client sends a message asking the other clients to post their current state. Each client responds with its current state. Since scene advancement is monotonically increasing, clients can safely synchronize to the largest scene number posted.

Advancing to the next scene can take place either after a fixed interval, or due to a script-specified set of roles requesting the advance. In the case of timed scenes, all of the clients maintain their own timers and advance when they expire. In the case of a user-driven scene advance, clients post the current state of the scene advance checkbox on their machine whenever it changes. Thus, all clients have the same information concerning scene advancement, and can make the same decision, based on the script.

Java Implementation

Lead Line was implemented using Java[11] and the 1.1 version of the Abstract Window Toolkit (AWT)[8]. It was implemented as an applet intended to run within a web browser. The Abstract Window Toolkit provided a sufficient set of UI elements to implement the fairly basic user interface used with Lead Line. Its wide distribution as part of the browsers we planned to support was crucial. Java also provides a well-designed set of network classes, which made development of the prototype quick and easy.

However, there were some problems with using Java and the AWT. Because of the lack of good text display facilities in the AWT TextArea class, we were unable to employ established text chat techniques such as using different fonts, colors, or weights to set off selected portions of the chat history text. Nor could we easily insert graphical elements into the history. Since we focused on simple text as the medium, this was not a serious problem, but the inclusion of some graphical elements would have been nice. Another limitation was Netscape's incomplete implementation of AWT 1.1 in their Macintosh browser. Some features associated with user input were not available. This meant that only Internet Explorer could be used on Macintosh. Future browser releases are expected to fix this.

RESULTS

Using the Lead Line prototype, we conducted an extended series of trials to determine which types of scripted social interactions work well and which do not. We ran over forty sessions with over thirty distinct scripts, covering a wide range of business and entertainment activities.

Following this, we conducted an experiment to compare structured and unstructured interaction in performance of a goal-directed task: ranking interview candidates. The experiment involved nineteen groups of three or four people (65 total, 31 female and 34 male, with a median age of 37 years, and a wide range of previous chat experience). Each participant was given notes from a set of hypothetical interviews with three candidates. Their task was to pick the best candidate. If all of the information in the notes was shared, the ranking of the applicants was obvious. Each group participated in two sessions, one structured and one unstructured, with the order randomized in order to counter-balance the results. The unstructured session simply gave the group twenty minutes of text chat time to come to a consensus. The structured session lead them through the steps of considering the requirements of the position, considering each candidate, comparing the candidates, and deciding. More details can be found in [9].

We learned or confirmed a number of things:

Brainstorming Works

As other researchers have found[30,20], one of the most effective on-line activities is brainstorming. The ability of multiple users to simultaneously create input – a fact that in other uses of the media is actually a deficit – means there is more input generated per unit of time. And since there is no imposed ordering or control of the “floor,” users feel freer to contribute their own views, whether extending a previous point or contributing a new one. Anonymity can also be used to help equalize the participants.

One of the trials used an established meeting facilitation technique known as SWOT (Strengths, Weaknesses, Opportunities, and Threats). The script contained a scene for each of these. Within each scene, users were encouraged to do open-ended brainstorming. The result was an effective evaluation of the topic at hand, without the involvement of a human meeting facilitator.

Another trial developed a fictional product, from conception through development, naming, marketing, and post-release evaluation. Although the script was used for fiction, it followed well-established procedures for product development. Each stage of the process involved free-form discussion and idea generation. The simple structure of the script, leading the users through the phases of the product, freed them to concentrate on brainstorming the part currently at hand.

Assessment Tasks Work Well

Another category of successful tasks is assessment. There is, in fact, quite a bit of similarity between assessment and brainstorming. Both are oriented toward getting as many ideas, facts, and opinions out in the open as possible. The SWOT trial described in the previous section is really an assessment task, as well as brainstorming.

In our goal-directed assessment experiment, described earlier, we measured the quality of the decisions and

whether the groups reached a consensus within the allotted time. Figure 2 shows how many groups reached complete or partial consensus, and figure 3 shows the quality of the decisions. Note that in some cases no explicit consensus was reached even though the members of the group had, in fact, come to the same conclusion concerning the candidate ranking. In each graph, we separate the sessions where scripted chat was used first (right pair) from those where unstructured chat was used first (left pair). As you can see, when it was used first, scripting improved the quality of the decision, but even more, it increased the probability that consensus was reached within the available time.

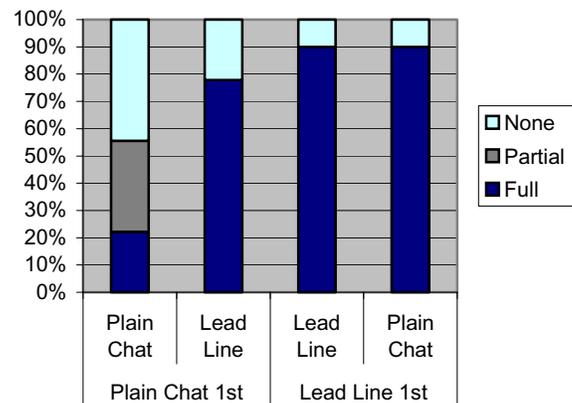


Figure 2: Consensus

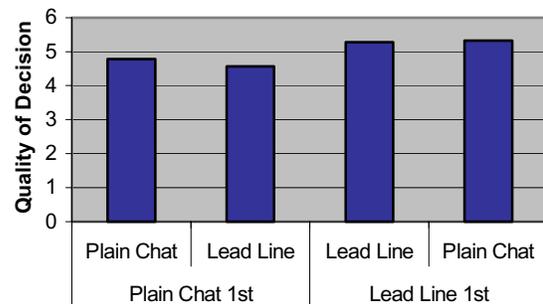


Figure 3: Quality

Lessons Learned from a Script Persist

People do not just blindly follow a script. They also learn from it, and what they learn is used in subsequent interactions, whether those interactions use the same script, a different script, or no script at all.

In figures 2 and 3, notice the difference in consensus and quality of plain text chat when it follows a structured session, as compared with the results when it comes first. When scripting was used first, the subsequent unscripted session was more effective. A closer analysis of the content of the session transcripts reveal that people learned

effective techniques from the first session and applied those techniques to the following one.

Structure Enables Creativity

It is tempting to make the assumption that adding structure dampens creativity. Previous structured online systems were considered overly constraining[7]. But we found exactly the opposite: an appropriate level of structure allows people to be more creative. It frees them from uncertainties of the context and goal and lets them concentrate their imaginations where it really counts. This has been known and exploited in improvisation for some time[33,28].

An entertainment title that we created set the stage for user-driven farce. The setting is a date at a fancy French restaurant. The roles are the snotty waiter, plus a couple who are somewhat less than sophisticated. This is a very simple setting, immediately understandable to users, with very little plot, but it provides enough context and grounding for users to create wildly imaginative content. Of particular interest was the fact that every time this script was played – and we played it a lot since it was used often to demo the prototype – the dynamics as well as the details were completely different.

Recognized Contexts Can Speed Interaction

People are used to working within an established framework or context. With an on-line system, that context is not always immediately obvious, as it usually is in a face-to-face encounter due to the process of physically arriving at it. If a script creates a setting that is well known to the users, it can speed the process of getting down to the core business at hand and can make them more at ease.

The French restaurant farce described in the previous section is an example of this. Since everyone is familiar with French restaurants, including their occasional reputation for snobbery, players immediately understood where they were, what was expected of their characters, and even what sorts of jokes to start out with.

In another trial, Lead Line was used for pre- and post-conference support for the International Forum of Visual Practitioners. These scripts were intended to break the ice before the conference and get people thinking about what they wanted to get out of the conference; and as a debriefing after the conference both for those who had attended it and those who could not. Both scripts were set in a fictional social setting, with a host that guides the conversation through a series of topics. The pre-conference script took the form of a walk in the park. The post-conference script was called Café El Marko, complete with fictional fresh-baked cookies.

Identity Can Be Used As A Tool

Unlike face-to-face interaction, it is possible on-line to control the degree of identify versus anonymity that a person exposes. This can be used to dramatically alter the nature of an interaction. More anonymity can make users feel freer to share their true opinions. More identity can

make the interaction hit closer to home. And changing the identity of the user from one associated with them as an individual to one associated with their professional role can make the proceeding more business-like and impersonal.

In one Lead Line trial, we simulated a personnel problem. An employee was found browsing X-rated sites on the Internet using company equipment. Roles included the employee, his immediate supervisor, and his wife. Even though this was envisioned as a workplace training exercise, the ability of users to distance themselves from their true identities through role-playing allowed a much freer exchange of ideas.

In another title, we explicitly tried to bring out the real person. Rather than having roles, people were simply themselves. The script took the form of a dinner party, and encouraged people to reveal personal details about themselves. When this script was used within our group, where people already knew each other, it was quite successful. But when we tried the same script in a usability trial, where the participants were strangers, they were extremely reluctant to share any personal information.

Roles Can Empower the User

Users take their assigned roles very seriously. It defines for them who they are and what they can do. They are surprisingly unwilling to break the bonds a role imposes.

In a morality play script used in the Lead Line trials, there were roles for a well-to-do character and a panhandler. The panhandler tries to get money; the other role tries to get away without giving any. In the second half of the script, the users reverse roles, but otherwise rerun exactly the same story. This proved quite enlightening, and somewhat disturbing, as users realized how much difference perspective makes.

In the product development script, described earlier, the roles are identified by professional title. Since this was an entertainment title, the titles were made up: Mesaphase Engineer, Altonian Designer, and Project Folder. But people invariably mapped these titles to real life: engineer, designer, and manager. Often, people would make product suggestions only in keeping with their perceived role.

SUMMARY & FUTURE WORK

Using a simple text chat scripting system, we have explored several aspects of online scripted social interaction. The results of these explorations have confirmed existing expectations of what types of interaction can be effective online, and added some new ones.

In the future, we plan additional formal experiments using the Lead Line prototype. We are particularly interested in determining aspects of scripted online systems that can be made as or more effective than their offline equivalents. We also want to investigate how to make scripted social interaction in task directed activities interesting as well as effective. And we have already started a project to explore computer-augmentation of face-to-face interactions.

REFERENCES

1. Bers, Marina. "Zora: a Graphical Multi-User Environment to share Stories about the Self." CSCL, 1999.
2. Bohm, David. *On Dialogue*, edited by Lee Nichol. New York: Routledge, 1996.
3. Bray, T., J. Paoli, and C.M. Sperberg-McQueen (ed). *Extensible Markup Language (XML) 1.0*, World Wide Web Consortium Recommendation, 1998.
4. Brown, Juanita and David Isaacs. "Building Corporations as Communities: Merging the Best of Two Worlds." *Community Building: Renewing Spirit and Learning in Business*. San Francisco, New Leaders Press, 1995.
5. Brown, Juanita and David Isaacs. "Welcome to the World Café." 1998. <http://www.theworldcafe.com>.
6. Bullen, Christine and John Bennett. "Groupware in Practice: An Interpretation of Work Experiences." In *Computation and Controversy*, Academic Press, 1991.
7. Carasik, R.P. and C.E. Grantham. "A Case Study of CSCW in a Dispersed Organization." CHI, 1988.
8. Chan, Patrick and Rosanna Lee. *The Java Class Libraries: 2nd Edition, Vol. 2*. Addison-Wesley, 1997.
9. Farnham, S. D., Chesley, H. McGhee, D., & Kawal, R. *Structured On-line Interactions: Improving the Decision-making of Small Discussion Groups*. In *Proceedings of CSCW 2000*. December 2000.
10. Flores, Fernando, Michael Graves, Brad Hartfield, and Terry Winograd. "Computer Systems and the Design of Organizational Interaction." ACM 1988.
11. Gosling, James, Bill Joy, and Guy L. Steele, Jr. *The Java Language Specification*. Addison-Wesley, 1996.
12. Greif, Irene. *Computer-Supported Cooperative Work : A Book of Readings*. Morgan Kaufmann Publishers, 1988.
13. IDC Intranet Fact Book: First Half 1999, IDC, July 1999.
14. Imai, Masaaki. *KAIZEN: The Key to Japan's Competitive Success*. New York: McGraw-Hill, 1986.
15. Klein, Donald C. "SimuReal: A Simulation Approach to Organizational Change." *The Journal of Applied Behavioral Science* 28, 4 (December 1992): 566-578.
16. Meister, Jeanne C. *Corporate Universities; Lessons in Building a World-Class Work-Force*. New York, Irwin Professional Publishing, 1998.
17. De Michelis, Giorgio and Clarence A. Ellis. "Computer Supported Cooperative Work and Petri Nets." *Lecture on Petri Nets II*. Berlin, Springer-Verlag, 1996.
18. Mitchell, Don. *From MUDs to Virtual Worlds*. 1995.
19. Nua, http://www.nua.net/surveys/how_many_online/index.html, as of April 2000.
20. Nunamaker, J.F., Alan Dennis, Joseph Valacich, Douglas Vogel, and Joey George. "Electronic Meeting Systems to Support Group Work." *Communications of the ACM*, ACM, 1991.
21. O'Day, Vicki, Daniel Bobrow, and Mark Shirley. "The Social-Technical Design Circle." CSCW, 1996.
22. Oikarinen, J. and D. Reed. *Internet Relay Chat Protocol*. Internet RFC #1459. May 1993.
23. Owen, Harrison. *Open Space Technology*. San Francisco, Berrett-Koehler, 1997.
24. Pinsonneault, A., and K.L. Kraemer. "The Impact of Technological Support on Groups: An Assessment of the Empirical Research." In *Decision Support Systems 5*. North-Holland, Elsevier Science Publishers, 1989.
25. Richmond, B. "The Strategic Forum: Aligning Objectives, Strategy and Process." *Systems Dynamics Review* 13, no. 2 (summer 1997): 131-148.
26. Robinson, M. "Computer Supported Co-operative Work: Cases and Concepts." *Proceedings of Groupware*, 1991.
27. Sadalla, Gail, Meg Holmberg, and Jim Halligan. *Conflict Resolution: An Elementary Curriculum*. San Francisco: The Community Board Program, Inc. 1990.
28. Scala, Falaminio and Henry F. Salerno (translator and editor). *Scenarios of the Commedia dell' Arte*. New York, Limelight Editions, 1996.
29. Schmidt, Kjeld. "Of Maps and Scripts: The Status of Formal Constructs in Cooperative Work." ACM SIGGROUP Conference on Supporting Group Work, 1997.
30. Siegel, J, V. Dubrovsky, S. Kiesler, and T. McGuire. "Group Processes in Computer-mediated Communication." In *Organizational Behavior and Human Decision Processes*, 37, 157-187. 1986.
31. Siler Todd. "Metaphormers: Connecting Our Work and Our World Through Metaphorms, " *R&D Innovator* 3, no. 7, 1994.
32. Siler, Todd. *Think Like a Genius*. New York: Bantam Books; London: Transworld, 1997.
33. Spolin, Viola. *Improvisation for the Theater*. Evanston, Illinois, Northwest Universtiy Press, 1983.
34. Weisbord, Marvin and Sandra Janoff. *Future Search*. San Francisco, Berrett-Koehler, 1995.