APPLIED THEORY

SUMMARY
- Examines new developments in interactivity for online authors and developers
- Suggests the metaphor of procedural architecture for authoring strongly interactive technical documents
- Considers rich internet applications and gaming as emerging forms of interactive technical communication

Making the Most of Interactivity Online Version 2.0: Technical Communication as Procedural Architecture

RUDY MCDANIEL

INTRODUCTION

The question of interactivity in technical communication is complex. First, building interactive technical documents means relinquishing some amount of control over authorship, readership, and document presentation, each of which can be frightening for a variety of reasons, not the least of which is the uncertainty involved in this process. Second, interactivity in technical documentation must also compete with interactivity in entertainment, a moving industry target with billions of R&D dollars invested each year. For example, a gamer who grows up with commercial video games is likely to shun poorly designed "serious" video games (Michael and Chen 2006) just as a consumer with an iPhone or iPod Touch will be annoyed by prolonged, awkward interactions with a rudimentary touch screen. Poorly designed interactive technologies may suffer from a lack of credibility, a lack of engagement, or a lack of effectiveness in training users or transmitting information from author to audience. Third, thinking deeply about interactivity necessitates the crossing of disciplinary boundaries and the consideration of other areas of research such as information architecture, experience design, and human factors usability. Such fields often have their own definitions of and ideas about interactivity, ideas that are not always compatible with traditional practices in technical communication. Finally, when these information practices occur online, there are numerous other factors that must be addressed because of the nature of the medium. These factors range from the technical (for example, systems that are too highly interactive may put a strain on server resources) to the practical (for example, online readers may have already formed information-seeking habits from their prior interactions with other media types. All of these complex issues must be considered when pondering the nature of interactivity online.

Despite its complexity, interactivity is a critical component of the informational experience in online environments, and one deserving of our attention (Chou 2003). Rarely is online information today presented in a linear format that reads as a cohesive unit to be parsed from top to bottom; instead, readers are asked to participate in different ways to move from one unit of information to the next. This movement may rely on a reader's clicking of hyperlinks, on her use of multimedia buttons in a video application to fast forward or rewind, or on her control of a virtual avatar to transport to the correct meeting location in a virtual world. Independent of the specific interactive implementation is the fundamental need of the computer system and its human users to be in tune with one another as this process unfolds.

To do this well, we need multiple disciplinary perspectives, even though this further blurs already nebulous boundaries between technical communication and related fields concerned with the design and assessment of media for communication. As the recent special journal issue *Designing Virtual Worlds*, guest edited by Sean D. Williams (2008) suggests, technical communication and digital media in particular are becoming more intertwined; one result of this is that virtual worlds, game-based learning, and cutting-edge tutorial systems now require more sophisticated types of audience interactivity. In these types of

Manuscript received 12 November 2008; revised 19 May 2009; accepted 3 July 2009.

environments, readers are asked to do more than turn pages, click buttons, or move scrollbars to access primarily textual information. Instead, they are called on to enter and negotiate virtual environments, to perform embodied actions using virtual avatars, and to learn new languages for communicating with organic and flexible interfaces. Although in many cases these types of training and documentation environments are still on the horizon, it is clear that interactivity will play an increasingly important role in the technical communication landscape of the future.

Fortunately, the consideration of interactivity as a facet of technical communication has been previously documented in the literature. In 2001, an article by Andrisani and colleagues explored the different methods of technical information presentation and the role interactivity plays in these methods. This was an interesting and important article because it considered interactivity as both an isolated and holistic dimension of technical communication from a variety of perspectives, both print-centric and online. Although only 8 years have passed since the publication of this essay, 8 years is a long time in terms of digital media technologies. For example, using Moore's Law (1965) as a timeline translates those 8 years into over five generations of technology growth (assuming technological "generations" of 18 months). We can therefore revisit some of the questions posed in that original article and consider how interactivity in this field has changed because of ideas from recent literature and new development techniques in interactive media design.

INTERACTIVITY IN TECHNICAL COMMUNICATION

The concept of technologically mediated interactivity and feedback can be traced back to the early days of telegraphy, which eventually led to the concept of wireless communication through radio waves. Telegraphy refers to communication over a distance, and as Petroski (1997) reminds us, this concept predates electricity by hundreds of years. Before wireless computers connected us all through the Internet, communication was limited by more primitive sensorial characteristics: the distance from which a smoke signal was observable or the distance from which one could hear a beating drum, for example. As these early mechanisms for communication evolved into more complex mechanical devices such as semaphores, shuttle telegraphs, and eventually Marconi's wireless radio system, geographical boundaries were no longer prohibitively restricting in terms of what would eventually become interactive communication.

Eventually, these technologies moved toward networked computing in the late 1980s and early 1990s. In the early 2000s, online developers realized that the primitive forms of interactivity enabled by the Internet would quickly be surpassed as networking technologies enabled faster download speeds and higher capacity bandwidths. It was only a matter of time before blogging, wikis, online video sharing, and social bookmarking would materialize and provide newer and more engaging forms of interactive online communication. In this new social environment, interactivity was no longer restricted to simple communicative tasks but often manifested in readers making creative and intellectual contributions to a body of work. In other words, they began playing productive roles in the creation of media content (Atkinson 2008).

This empowerment of readers through interactivity captured the attention of technical communicators in the early 2000s. In their article *Making the Most of Interactivity Online*, Andrisani and colleagues (2001) began their analysis by addressing the common misconception that interactivity is simply a function of engagement with technology (for example, clicking buttons, entering data, or moving from one virtual page to the next). As they note in their introduction, interactivity is in fact a much more complex phenomenon, and one that is largely invisible, depending as much on cognitive events as physical actions. They explain that an important dimension of online interactivity is allowing the user to a play a role in the "data discovery process" and continue with the definition of interactivity offered by Vince Cyboran in 1995. Cyboran defines interactivity as a "process in which the learner and the system alternate in addressing each other." The authors maintain that such a process "guarantees the effective exchange of information." Although we can acknowledge that such an alternating, communicative relationship does guarantee the exchange of information, we must also question the suggestion that this exchange will be consistently effective. Many online systems that include continuous feedback can hamper an audience's interpretation or understanding of technical content by including too much information. Or, perhaps, the computer provides information at too slow or fast a pace or at too complex a reading level. These are just a few possible inhibitors to effective interactive communication.

For interactivity in online help systems, the authors suggest using the book metaphor as a design heuristic when developing online content navigation. They also recommend taking advantage of the flexibility of hypertext to create accessible information and giving users control over the process of hiding and revealing information. Of particular interest in this section is the discussion of interactivity from either the user's or designer's perspective. The authors point out that both approaches involve notions of predictability. They claim that users tend to conceptualize online systems as "open," whereas designers are more inclined to see the system as "closed," predictable, and limited. This is one fundamental way in which I argue interactivity has changed in the past 8 years, and I will return to this idea in the second half of this article. The authors continue to discuss mechanism for interactivity in online help systems including interaction through information access (for example, keyword or full-text searches), navigation (for example, clicking through hyperlinks or online tables of contents), and the hiding or revealing of information (for example, clicking on different links to hide or display supplementary information).

Computer-based training (CBT) involves a variety of interactive learning techniques embedded in tutorials, knowledge-based programs, games, and simulations (Andrisani and colleagues 2001). The authors note that tutorials exhibit lower levels of interactivity, whereas games and simulations are more interactive. In terms of guidelines for building interactivity into CBT, they suggest keeping in mind the various learning styles of students and using multimedia, good design principles such as effective use of color and white space, and consistency in the placement of text and graphics to present the information. Furthermore, the authors stress the importance of feedback and note that feedback containing the correct answer to a given question can be provided in a variety of ways, from immediate corrective feedback to the more effective mandatory practice feedback, in which a learner is encouraged to continue working on problems she has missed until the concept is better understood.

To summarize, the role of technical communication in authoring effective interactive environments as suggested by Andrisani and colleagues (2001) involves:

- Setting and defining limits
- Providing accuracy and consistency of content and presentation
- Helping to gain the trust of users
- Creating information access mechanisms
- Planning effective navigation
- Allowing users to hide and reveal information
- Ensuring that linking systems are intuitive
- Using multimedia to address different learning styles
- Applying good design techniques when using color, white space, and typography
- Providing feedback effectively to sustain the interactive session
- Constructing user-centered interfaces
- Using contextual analysis and task scenarios to better understand user needs
- Using familiar metaphors (such as the paper page or book) to ease the transition from print to online materials
- Understanding the role of digital architecture in electronic documentation

Interactivity as conversation

Andrisani and colleagues began their article using Cyboran's definition of interactivity. This definition is supported by additional researchers in digital media, instructional design, and other disciplines. For example, Crawford (2005) defines interactivity metaphorically as "a cyclic process between two or more agents in which each agent alternately listens, thinks, and speaks," thereby positioning interactivity at its most fundamental level as a discursive process. Lopistéguy and colleagues (2006) used a similar metaphor, noting that interaction involves a sequence of exchanges bound by coherence and with a particular conversational goal. An interactive exchange is composed of "one or more interventions, an intervention being considered as one locutor's contribution." Although these definitions might suggest that interactivity is at heart a disruptive system, we must remember that these interruptions happen all the time in normal everyday conversations. During a good conversation, concepts flow into one another seamlessly, and the speakers rarely focus on the interruption but rather on the topic of discussion. This concept is recognized by researchers such as Schank (1995), who defined conversation as a "process of reminding." Good computer-based interactivity can be thought of as a gentle prompting and prodding—the user is smoothly steered toward a particular informational goal while the computer is continually asked to generate new information and look up any necessary content on the user's behalf.

The focus on conversation is important to note, because who better to author technology-mediated conversations than technical communicators? Much of modern applied technical communication involves empowering readers to feel as though they have agency in finding, reading, and assimilating relevant information from different types of technological systems (such as hypertext documentation, CBT, and electronic manuals). This information is applied toward solving particular types of problems. Ideally, this process would resemble the same type of good conversation described above. A conversational query is presented to a computer; the computer then responds appropriately. If more detail is required, the reader provides it, and the computer politely interrupts if it has helpful information to offer as the reader is speaking or typing. In this fashion, the interaction unfolds in a more intuitive fashion, in a way that resembles the conversational mechanics we already know so well.

We can further extend this conversational definition to consider different categories of interaction. Ryan (2001) noted that interactivity can be defined figuratively, as "the collaboration between the reader and the text in the production of meaning," and literally, as "a choice between predefined alternatives." She noted that the former definition is what she calls a "strong form," enabling the reader to play an active, constructive role in the reading, whereas the latter definition is a weaker form that requires little to no complex thought. This again relates to conversation; in a

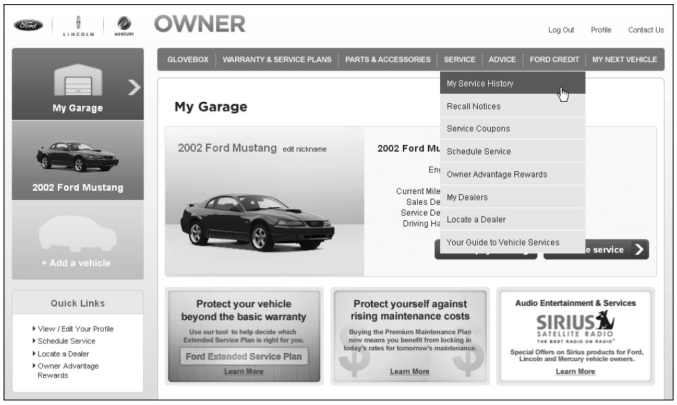

Figure 1. Personalization in interactive Web portals

good/strong conversation, each individual can contribute and has the opportunity to direct the flow of discussion. In a bad/weak conversation, one individual provides the entirety of the conversation and may only allow other individuals to interject occasionally and decide what the current speaker should talk about next.

Although the weaker forms of interactivity are fairly easy to implement in technical communication systems, the stronger forms that lead to the co-construction of meaning require a more sophisticated understanding of both audience and technology. In many ways, strong forms of interactivity such as those described by Ryan operate according to what Bühlman (2003) called the poetics of interactivity. In an ideal interactive piece, we strive for a state of immediacy in which a “newly experienced facet of the real is brought into existence by personally feeling addressed by whatever we encounter” (Bühlman 2003). Weaker forms of interactivity allow us to bring information to focus, but stronger forms allow us to personally connect with the information in a meaningful way. For instance, consider the difference between a general automotive reference Web site and one that has been personalized through registration details (Figure 1). In the former interface, a vehicle owner can look up information related to her vehicle in a general database. This is like talking to a stranger, and some preparatory dialog is needed before a really interesting conversation can occur. In the latter, deeper conversations are possible because the user has previously registered her vehicle in a virtual garage and the system has access to prior information. This is more like talking with an old friend who already knows a good deal about you. In this case, the computer can initiate conversations (did you know it is time for another oil change?), personalize content (by showing an actual image of her vehicle with the correct paint color and trim), or even predict trends (is it time to upgrade from the two-seat sports car to a family-sized SUV?). The interaction in the latter scenario is stronger because the human being on the other side of the conversation feels personally addressed and therefore invests more attention to the process. Furthermore, the “meaning” of this site is co-created by both the algorithmic rules of the information system and by the user’s prior and current activities on the site. The collection of personal details through registration systems is one strategy for accommodating this shared creation of online meaning.

Also central to this interactive process is the integration and adaptability of feedback, which Cyboran (1995) noted is crucial for providing information back to learners regarding their progress and understanding of materials. Nonverbal cues are simple examples of feedback in real world conversations; even while our audience is listening to us speak, we can gauge the quality of our conversation by observing their bodies as they react to what we are saying. At a broader level, feedback is useful in other virtual technical communication applications as the computer can use feedback to provide physical cues (for example, the location of a virtual location within a broader hierarchy of navigable pages), cognitive cues (for example, real-time performance feedback), or affective cues (for example, adjustments to complexity or challenge from perceived attitudinal or motivational measures). Interactivity can therefore be positioned as an important factor in each of the three dimensions (physical, cognitive, and affective) of information design as conceptualized by Carliner (2003).

Technical communication as procedural architecture

It is clear that technical communicators need to be familiar with the concept of interactivity and the applied use of its various forms. In fact, if we restrict our study of interactivity to those conversational processes that surround technical communication in modern workplace practices, interactivity in many senses *is* technical communication. Doheny-Farina (1992) in particular has written at length about the role of interactivity in R&D development, noting the special importance of the verbal interactive networks that act as bridges between experts in various information processing capacities. Understanding how to navigate the rhetorical spaces that are formed between social and technological systems for maximum effectiveness is therefore an important task for technical communicators to undertake.

Technical communicators can use their knowledge of audience, context, and content to help devise and design interactive technologies that are intuitive to use and yet flexible enough to satisfy a variety of informational needs. Because this process involves a careful deconstruction of rules, both technological and organizational, as well as knowledge of virtual spaces and the affordances they create, I suggest the label procedural architecture to describe the types of responsibilities these roles will entail. Procedural architecture also refers to a more specific domain in interaction design in which focus is applied to isolated rules rather than to content or data; this is similar to the concept of procedural actions as characterized by Cooper and colleagues (2007). These authors define procedural actions as actions that act on data but that do not add, modify, or delete that data; this is in contrast to incremental actions, which do include operations on data. In part, procedural architecture requires one to think about a system in terms of its rules and procedures before adding in any specialized content.

Procedural literacy is the ability to understand and work with the rules of a system and to understand the interrelationships formed between content, audience, and technology when these rules are activated (Bogost 2007). This involves an understanding of which new rules are necessary to add value to a document, as well as knowledge about how existing rules and procedures (both technological and socio-organizational) can be used effectively within a given system. Procedural literacy is at work when an author transforms or translates one document type or view into another and knows when this type of transformation will be effective for an audience. It is also applied when an author effectively packages a proposal and navigates the political hierarchy of an organization to have content approved quickly. In these types of activities, technical communicators are already highly procedurally literate.

Cutting-edge procedural architects, however, also need to be familiar with certain lower-level rules that drive technology. These are the rule-based systems that store, produce, and manipulate data in media systems, both in isolation and in terms of the broader relationships these rules form with content and organizational procedures. This is the procedural component of procedural architecture. To use an example from the field, much has been written about the importance of the eXtensible Markup Language (XML) for technical communication (Applen 2002), but very little about XML parser design. This is disappointing given that XML is rather useless without a software parser capable of doing interesting things with the data from the way in which it is coded (Applen and McDaniel 2009). XML alone is static, but XML at work solving a particular problem is procedural. Granted, much of the parser design work is done by programmers, but technical communicators can also add value to an organization by working closely with programmers to identify functional requirements and construct design documents capable of working within the limitations of available technology. Such an arrangement is already producing interesting results in industry. For example, Stavely and colleagues (2008) discussed the importance of having technical communicators and programmers together and note that this hybrid team environment can improve both computer code and technical documentation through techniques such as paired programming and collaborative editing. Practices such as this will continue to gain importance as documentation deliverables become more sophisticated and require authorship from writers, programmers, graphic designers, and information architects.

Various heuristics can be used to familiarize oneself with different approaches to procedural design. Many strategies are found in prior work done in the field of interaction design; four examples of applied techniques are summarized in Table 1. Although a detailed discussion of these methods is outside the scope of this article, additional information about them is available in the associated references listed with each approach. Although these techniques are certainly useful when designing documentation systems from the outset, they can also be valuable as analytic tools when considering the existing procedures within a system and how those procedures might be changed to deliver more effective interactions. They are obviously more effective when used in tandem with good design practices concerning usability, readability, audience analysis, information architecture, and so forth. Also, as interaction design pioneer Gillian Crampton Smith noted (quoted in Moggridge 2007), the usefulness of a product is entirely different from the usability of a product; in other words, one might consider if the procedural dimensions of a system are not only usable, but also useful in solving a particular type of informational problem or addressing an informational need.

Technical communicators also add value to interactive online media through their knowledge of data visualization techniques and the potential interactions that can occur in three-dimensional (3D) worlds. This is the architectural expertise needed for procedural architecture. This follows the line of discussion recently initiated by Williams (2008) in his special issue: what is the role of technical communicators in the construction of immersive 3D worlds? Because technical communicators are routinely asked to broaden their understanding of textuality to transcend the printed page, they are uniquely equipped to recognize the communication challenges that emerge when training or documentation materials move from 2D to 3D space. With games and simulations, such environments are now the status quo, and technical communicators equipped to recognize the interactive possibilities of 3D space will find themselves at an advantage in catering to audience expectations. If interactivity in 2D space is characterized by clicking links, pushing and pulling scrollbars, and pressing buttons, interactivity in 3D space deals with more complex mechanics at both a technological and ontological level. Technologically speaking, users in 3D worlds use key-

TABLE 1: TECHNIQUES FOR PROCEDURAL TECHNICAL COMMUNICATION

Approach	Description	Further explained in
Conceptual models	Consider the activities users will be performing when carrying out their tasks. For example, when a user is reading assembly instructions on a mobile device, what transformations of content might be useful, and which functions will they need to access? In unfamiliar applications, interface metaphors (e.g., desktops, search engines) are useful tools for building conceptual models.	Preece and colleagues (2002)
Design verbs	Design the verbs of an informational system first, even before considering the technology. For example, what are the choices available to the user/reader in a virtual library program? This example might produce initial design verbs such as start program, change screen size, move avatar, change perspective, and scroll one page.	Crawford (2003)
Scenario-based design	Use concrete examples to build information systems. Scenario-based design tools include descriptions of an environmental setting coupled with descriptions of the agents/users who will be accessing the system in various roles. Some examples of scenario types include persona-based scenarios, context scenarios, key path scenarios, and validation scenarios.	Cooper and colleagues (2007)
Human factors hierarchy	Consider the rules and procedural interactions of an information system along a continuum that increases in complexity. First, consider the rules related to anthropometrics (physical size), then overall physiology, then psychology, sociology, anthropology, and finally ecology. Identify the most important rules from these categories and focus on those. For instance, a mobile training manual might offer rules to produce scalable buttons for differently sized fingers (anthropometric and physiological rules) while also leveraging rules to automatically email specialty technicians and project managers a report of prior history and current status when diagnoses are made (sociological, anthropological, and ecological rules).	Moggridge (2007)

boards to move virtual avatars around an environment and learn the various commands to make that avatar behave in a way conducive to the learning or training outcomes of a scenario. Ontologically speaking, a user must also translate her real-world identity to a virtual identity and determine how she would like her avatar to behave. This is what Gee (2003, p. 55) refers to as the "projective identity," a term that uses two senses of the word "project." The first sense is as a verb, meaning "to project one's values and desires onto the virtual character," while the second refers to noun, as in "seeing the virtual character as one's own project in the making" (Gee 2003). Given this complexity, technical communicators have many new challenges to deal with in these types of environments.

If one thinks of 3D worlds as navigable documents that are designed to communicate technical information, many of the same tips offered in hypertext systems (such as choosing suitable technological cues and using consistent design standards) are still important for 3D interactions. For example, if objects in a virtual world are intended to be manipulated by a player, they can be designed to emit a soft glow when a player nears their proximity. The same cues can be used to differentiate nonplayable characters (NPCs) who offer useful dialog from those who simply wander around and have no direct effect on the player or the game's objectives. The ontological dimension, although more difficult to conceptualize as a design element, can also be addressed by the informed technical communicator. Using a hybrid audience analysis that takes into account both real players and their virtual avatars' likely encounters with characters in the game world, it is possible to construct a probability space in which certain actions are more likely to occur.

In this probability space, the rules of a system will ensure that certain outcomes are more likely than others, or even that they are absolutely necessary if the player is to advance further in a game. Consider this example. If the goal of a training exercise in an online *Flash*-based game is to have retail employees learn to effectively deal with difficult customers, a variety of virtual decisions could all eventually lead back to the same outcome. A player might choose to ignore the virtual customer, who would storm away from the register angrily only to later return in full force. Or, the player might choose to immediately address the customer, to the acute dissatisfaction of other customers waiting in line. Or, the player could choose to phone a manager on lunch who would deal with the customer but cause problems for the player later in the game. In each of these situations, different player interactions lead to different outcomes, but the same ontological projection (the tension in the game elevates to communicate the discomfort of such a situation to the real person playing the game) is still established through careful planning and knowledge of the complexities of procedural systems. In symbolic analytic terms, the outcomes and decisions are symbols and the analysis determines the pathways that should be constructed between these symbols within this probability space.

NEW DIRECTIONS FOR APPLIED INTERACTIVITY

Two emerging areas of applied practice illustrate some of the principles of technical communication as procedural architecture. Specifically, recent and exciting developments in Web 2.0 architectures and online games-based learning push the envelope of what is possible with interactive technical communication techniques and modern technology.

Web 2.0 architectures

Web 2.0 is a phrase attributed to Tim O'Reilly in 2005 after a series of conferences emerged in which participants tried to define the future of the Web after the dot-com burst in the fall of 2001 (O'Reilly 2005). Although there still remains a large degree of ambiguity surrounding the name itself, Web 2.0 generally refers to new types of web environments that depend heavily on social interaction and community building. Examples of Web 2.0 web sites and applications provided by O'Reilly include *Flickr*, *BitTorrent*, *Napster*, *Wikipedia*, and *Google AdSense,* as well as community-driven activities like blogging and commenting, editing wikis, tagging bookmarks with metadata, and using social networking sites.

As procedural architecture involves an understanding of the systems and rules that drive information flow, it is not very useful to simply consider Web 2.0 technologies at a broad and conceptual level without thinking about the lower level details that accompany them. Just as technical communicators can offer expertise in the writing and organization of content, so can they provide knowledge about how to arrange, encode, and describe content so that it is most effective in rule-based systems. There are many traditional technologies at work in Web 2.0 sites, but there are also some newer design strategies which have emerged to allow for more immersive and interactive browsing experiences. One such strategy is known as AJAX development (Asleson and Schutta 2006; Garrett 2005).

AJAX, or asynchronous JavaScript and XML, is a web development technique that is exciting for technical communication because it uses a unique procedural delivery mechanic and customizable semantic encoding for the asynchronous exchange of technical information. In a procedural sense, this refashions the rules of the interactive information delivery model as shown in Table 2. Immediately, this suggests new questions relevant to the technical communicator's purview that might not otherwise be considered. What types of information might the visitor wish to read about while her actions are being processed? Is the usability of the site sufficiently well considered given that normal browser cues (for example, the page refresh) are no longer present? Is asynchronous development more useful for certain types of documentation, such as when a user needs to search a system and have suggestions appear in real time, and less useful in others, such as when a user needs to register a new account? Returning to our conversational metaphor, AJAX essentially offers a means for building helpful interruptions into sustained conversation. Technical communicators as procedural architects must con-

TABLE 2: SYNCHRONOUS VERSUS ASYNCHRONOUS INTERACTIONS

Model	Delivery method	Example
Synchronous interaction	Blocking; user's actions are limited until a response is returned from the Web server	Jim fills out a search form to look for the term "Alpha Widget" on a software documentation help page. He must wait until the form has executed his search on the server, queried the database, and returned a new page before refining his search. If there are problems, Jim must correct his query before being allowed to pursue other activities on the site.
Asynchronous interaction	Nonblocking; while the information is being processed by the server, the user is free to continue browsing the site	Jim fills out a search form to look for the term "Alpha Widget" on a similar site that uses asynchronous information transfer. As soon as he begins typing, a suggestion box provides other common searches beginning with that sequence of characters. After typing "Al," Jim's search query appears and he is able to select it and perform the search. While the search is running, Jim is free to continue reading other areas of content on the site. If no results are found, the widget indicates this to Jim and he can refashion his query when he feels like it.

sider when such interruptions are useful and how these interruptions should be presented to the user.

AJAX is already widely used in commercial settings by companies like *Google*, *Amazon*, and *Netflix* to enable new types of design patterns for information access and retrieval. As both a design methodology and a set of associated technologies including JavaScript and XML, AJAX enables the construction of Rich Internet Applications (RIA), which strive to bring the same qualities and characteristics of the desktop computing experience to the Web (Eichorn 2007). Online interactivity with AJAX is therefore moving toward the same type of interactive experience that one would find in a desktop application like *Microsoft Word*. Central to the RIA model is the responsive user interface and more sophisticated interactive capabilities (Lawton 2008).

Theoretically speaking, as an example framework for building RIA sites, AJAX does some interesting things to the model of on-screen textuality outlined by Bernhardt (1993) and discussed in the original *Making the Most of Interactivity Online*. In his original analysis, Bernhardt suggested that on-screen text differed from print text in the degree to which it was situationally embedded, interactive, functionally mapped, modular, navigable, hierarchically embedded, spacious, graphically rich, and customizable/publishable. Situational embeddedness refers to the fact that unlike many print materials, on-screen text depends on the context of situation, or what he describes as the "ongoing activities and events that make the text part of the action." Interactivity is encouraged by the functional mapping of textual elements (such as hyperlinks) that encourage readers to engage with content both mentally *and* physically. Modularity means that information is chunked into fragments of a particular granularity, whereas navigability is the primary means of reader movement through a text. On-screen texts contain other texts (again, hyperlinking is an example of textual layering) and as such are hierarchically embedded. They are graphically rich and contain both visual and verbal texts. Finally, on-screen texts are organic, and this open-endedness makes them more spacious and customizable than print texts.

Although AJAX web sites maintain many of these characteristics, the relationship between functional mapping and interactivity is altered in an interesting way. For instance, one of the key functions of AJAX is to allow information from a server computer to be returned asynchronously in a "nonblocking" fashion (refer to Table 2). This is the defining characteristic of asynchronous communication; updating information from a system does not prohibit the reader from continuing to interact with the text in other ways. As Gehtland and colleagues (2006) described this, "requests posted back to the server don't cause the browser to block. The user can continue to use other parts of the application, and the UI can be updated to alert the user that a request is taking place." From a usability perspective, this functional remapping using AJAX design techniques affords new interactive possibilities for both writers and readers. For instance, Google developers used the technique in their *Google Maps* application to allow readers to use the click-and-drag map navigation and pushpin routing features. *Gmail* uses AJAX to allow readers to read, tag, spell-check, compose, or search messages, all without waiting for pages to reload (Eichorn 2007). *Netflix* will display additional information about a film without requiring the reader to click on it; instead, they can hover the mouse cursor over a virtual film's box and the additional information appears as an unobtrusive pop up (Figure 2). The interactive space of RIA technologies such of AJAX allows web designers to create full-featured applications that behave as though they are running from a user's desktop rather than from a server somewhere far away.

Technical communicators have many features to choose from when developing online information systems that use asynchronous technologies such as AJAX. For instance, online help documentation systems can now enable users to click on an unknown word that is hyperlinked and retrieve a definition from a data source while still continuing to read a series of instructions. In contrast to the pop-up method of linking described by Andrisani and colleagues (2001), this type of linking content provides a more seamless integration with the current text's visual layout, meaning that even external content can be brought in and incorporated when additional elaboration or background is needed for a task.

AJAX also allows a designer more flexibility when designing documentation systems. For instance, rather than attempting to predict precisely how a user will access

Figure 2. Netflix hover feature using AJAX

content, a technical communicator can instead author online content to fit within a certain set of parameters and then allow the user to retrieve and display information in a more open-ended, visual fashion. For instance, rather than displaying every possible route between two geographical points, *Google Maps* instead provides functionality for zooming, panning, and wayfinding, procedures that are more like other real world objects (for example, physical maps, binoculars, digital cameras, and GPS systems) rather than simply extensions of the book metaphor that were so prevalent in early hypertext documentation. Similarly, *Amazon's Diamond Search* does not initially list every gem available for purchase. Instead, the site offers sliders for gradually narrowing down a comprehensive inventory on the basis of cut, clarity, color, and other gemstone characteristics (Figure 3). Furthermore, sliding one tab to the left or right does not prohibit the user from experimenting with moving other tabs while the results are still being returned. This type of design mechanic affords playfulness and exploration on the visitor's part rather than requiring the vendor to include exhaustive documentation for every possible interactive mechanic available on the site. The simplicity of the design and the speed in which feedback is provided to the visitor makes it easy for her to learn the rules of the system on her own, without requiring a visit to the help documentation.

Games-based learning

Playfulness and exploration are even more important in another type of interactive domain useful for technical communication: online video game environments. As An-

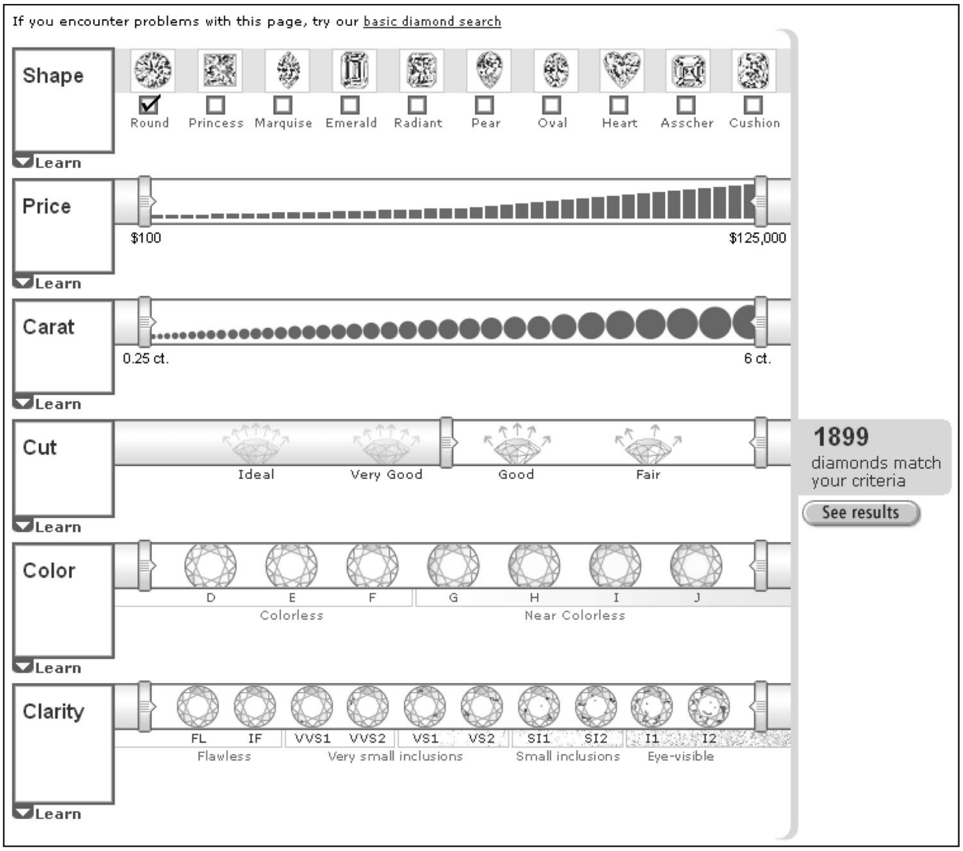

Figure 3. Amazon.com's diamond search

drisani and colleagues (2001) noted in their original article, games are a subset of CBT. They have long been used in training contexts by large companies such as Chase Manhattan, American Express, IBM, Reuters, and Bankers Trust. Today, companies continue to use games in various technical communication contexts. For instance, Jana (2006) wrote of companies such as Cold Stone Creamery, Cisco, and Canon using games to teach various technical procedures in virtual game worlds supplemented with fantasy and immediate feedback. *Stone City*, the employee training game developed for Cold Stone Creamery, simulates a physical store and allows store employees to scoop ice cream and serve customers all while learning about proper portion control (Bogost 2007). Figures 4 and 5 show screen captures of *Stone City*; note that the game's rules are constructed so that both excesses and deficiencies of ice cream lead to undesirable outcomes. Give a customer too much ice cream and the wasted ice cream amount increases. Give them too little and the customer becomes dissatisfied. The game features an "ice cream viscosity model for accurate scooping" (Persuasivegames.com), meaning that different flavors of ice cream will scoop differently depending on their composition. This is an important training concept that real world employees of Cold Stone Creamery need to know to correctly determine ideal portion sizes.

Canon uses another type of video game that borrows the board game *Operation*'s procedural mechanics to flash lights and activate buzzers when a repairman incorrectly places a copier part in the training game. Cisco teaches about networking concepts, an admittedly dry topic for most, by creating a sense of rhetorical exigency in which players must build a network in Mars during a sandstorm. In each of these cases, interactive feedback and other gaming characteristics (such as fantasy and reward) are used to activate interactive training environments with a greater degree of depth than traditional training materials (see Derouin-Jessen 2008).

As RIA methodologies such as AJAX continue to create new opportunities for practicing technical communicators, we will begin to see more games like those described above emerge in online environments. Already, so-called casual gaming is a profitable market for developers, bring-

Figure 4. *Stone City* (taking orders)

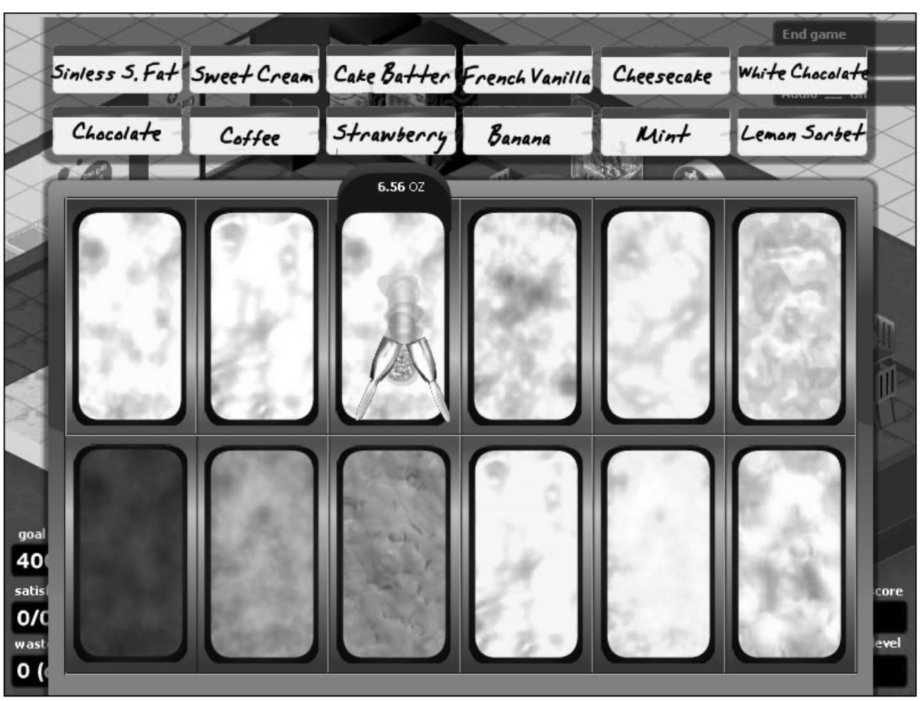

Figure 5. *Stone City* (scooping)

ing in an estimated $2.25 billion in yearly revenue (Morrison 2007). Building casual games using *Adobe Flash*®, AJAX, *Adobe Flex*®, or other modular design suites enables organizations to produce CBT games without needing the multimillion dollar budgets of commercial game titles. Because they are cheaper to develop, yet often still fun and highly addictive, casual games are very attractive for commercial training purposes. As technical communicators, we can play a role in helping these games evolve to their full potential.

Games-based learning as it relates to technical communication in general has been discussed in some detail in recent literature. For example, Eyman (2008) wrote about games from an ecological perspective, suggesting that games are “complex rhetorical spaces well suited for technical communication research and theory building” that present various opportunities for technical communication through different assemblies of technology, development, writing, and analysis. The ecological metaphor is appropriate because it takes into account both social and technological relationships that influence how information is distributed in the virtual worlds of games. Given that game design documents are often massive blueprints containing technical, creative, and logistical information (Rouse 2005), this is another domain in which boundary-spanning technical communicators should feel comfortable working.

What is the role of interactivity in these various elements? In other words, if we are to agree with the conceptualization of Eyman (2008) of gaming’s various dimensions as entry points for building symbolic-analytic technical communication skills, how does interactivity work within these complex systems? Once again, it is important to augment these symbolic-analytic skills with knowledge about procedural design. This requires an understanding of the rules and processes that shape information in different ways depending on the interactions of users and computers in various permutations (Friedl 2003). Bogost (2007) also uses the term procedural literacy to describe this cognitive ability, which he defines as an understanding of how complex rules and processes work together to make arguments about some system or complex practice.

Although procedural literacies are certainly at work in the actions of video game players, Bogost notes that even nongamers develop procedural literacy when they learn and apply the various behaviors necessary to navigate various rule-based processes in the real world. He uses the example of returning an item to a consumer sales store as a nongaming example. To successfully return an item, one must know the rules that shape that reverse transaction. How long has it been since the item was purchased? Is the original packaging intact? How strict is the company about enforcing the time limit for returns? Are there mitigating circumstances (such as prior business with the company or product recall notices that have been issued) that alter the rules of this system? Being procedurally literate means not only knowing and understanding the rules of a system, but also knowing when these rules can be bent, ignored, or reshaped into desirable outcomes for an individual or community of individuals.

Procedural literacy can also be applied during the production of games as information systems. Technical communicators exhibit procedural literacy when they make informed decisions about which physical design cues to include and which cognitive goals and affective considerations to incorporate into the design document and eventually the game itself. Players then exhibit procedural literacy by recognizing and capitalizing on the affordances provided within a game world. We can build procedural literacy competencies for interactive design by focusing on the needs of our audience. For example, if our aim is to build a documentation knowledge repository for skills related to the leveling up of characters for a massively multiplayer online role playing game, documentation should consider the same information design guidelines discussed previously. From a procedural perspective, this means understanding the rules of logical, affective, and cognitive design. Which characters are more likely to be chosen given the logical constraints of the world? Which levels produce the most frustrating experiences for players? Which puzzles are the most challenging? Considering rules such as these, although admittedly primitive, gives the technical communicator an idea about which portions of the documentation to develop more thoroughly than usual.

Because so many games are nonlinear and strive to give the player agency and the freedom to choose her own path through the game world, games are particularly useful texts to study as benchmarks for interactive design. Perhaps even more so than RIAs, online games can change the fundamental focus of interactive technical communication from predictability to probability. In other words, it is relatively impossible to predict how a player will move through a particular level of a game world, but it is much easier to create a level that encourages certain types of interactions. Following this line of thought, interactivity in game-based worlds becomes less about predictability and more about creating a probability space in which certain outcomes are more likely to occur. Different outcomes will result in different feedback to players, thereby increasing the probability of certain actions and decreasing the probability of others. Sid Meier is well known for defining gameplay as a series of interesting choices and for providing players with interesting decisions to make in regard to their virtual actions (Rouse 2005). Juul (2005) notes that this is enforced through game rules: “The rules of a game add *meaning* and *enable* actions by setting up *differences* between potential moves and events" (emphasis in original). Returning to Ryan's (2001) conceptualization of strong interactivity, probability spaces are strongly interactive because they trust the player to build her own path toward a game objective.

Eyman (2008) notes that technical communicators are somewhat behind the curve in understanding how these forms of media can be used by the profession. However, the fundamental components of games show the promise and potential of the medium as an interactive tool for training and even documentation. The games used by Cold Stone Creamery, Cisco, and Canon certainly communicate technical information, but they also virtually embody technical practice, a technique that positions learners as active agents within the rule-based systems that govern how technical or organizational parts fit together. Although the game used by Cold Stone Creamery trains players in how to scoop the proper portion of ice cream and the game used by Canon trains technicians to repair copiers, both are creating probability spaces in which a player can fail, reassess the situation, and try again as many times as she likes. This is perhaps the ultimate interactive, reader-centered scenario. Not only does the reader actively participate in the construction of the text through her actions, but she can follow multiple paths to reach the objectives set forth by the game designers.

Because of issues like nonlinear game design, players sometimes need a little push to advance forward in the direction a designer wants them to move. This feedback can be provided in a variety of ways, through dialog or mission briefings explicitly given to the player or through environmental changes that instill a sense of urgency into the world. The concept of urgency is especially important to drive a player's behavior and is one we should take note of as technical communicators. As professionals skilled in both rhetoric and technology, the design of these game spaces might be crafted around rhetorical exigencies or the urgencies that compel readers to seek or produce information to solve a particular problem. These urgencies can be reflected through technological rules that populate a game world with interesting problems. This is the tactic followed by Sheridan and Hart-Davidson (2008), who designed a game-based world named *Ink* to teach literacy skills on the basis of the urgency of a virtual neighborhood in disrepair. The interactivity in this game was encouraged by a Request for Proposals that was circulated to a group of student participants. By carefully setting up the interactions of this game world within the context of a rhetorical need, the virtual neighborhood catered to ambitious students who were willing to help find ways to recreate a sense of community within a given segment of virtual space.

FIVE GUIDELINES FOR INTERACTIVE TECHNICAL COMMUNICATION

In Web 2.0 and gaming environments, the process of interactive design has evolved, and this changes things for technical communicators. As argued in this paper, interactivity does not guarantee the effective exchange of information. Similarly, interactivity as a design heuristic is no longer just about predictability but rather about the planning of a probability space for desirable outcomes. Weaker forms of interactivity such as clicking and scrolling are still important, but should not be a given component of each and every online documentation project given the new capabilities of design frameworks like AJAX. Stronger forms of interaction, in which a reader co-creates meaning by actively participating in the shaping of the content, are well-suited for Web 2.0 and gaming environments, virtual spaces that are outfitted with reactive and exploratory interfaces. Finally, although the book design metaphor remains important, new and emerging forms of interactive technical communication will require additional metaphors, metaphors of other real world objects and architectures, to account for the embodied nature of new simulations and game-based training in online contexts. Table 3 summarizes the changes to selected guidelines for effective interactive online technical communication (as discussed in Andrisani and colleagues 2001) in light of the research and practice discussed in this paper. In particular, Table 3 focuses on emerging forms of interactive technical communication such as RIA and online games.

As Table 3 suggests, for today's technical communicators wishing to author interactive environments for documentation or training in Web 2.0 environments, symbolic-analytic work must include a procedural dimension. Technical communicators are already well versed with the effective use of rules, whether these rules concern grammar and syntax, the proper level of granularity for chunked content, or the most effective use of font-face or imagery. To summarize the major points of this article as inspired by the original ideas of Andrisani and colleagues (2001), I suggest the following five guidelines for designing strongly interactive systems in technical communication:

1. Rather than striving for predictability, technical communicators should think about constructing probability spaces where desirable outcomes are likely to occur. This facilitates a stronger sense of user agency, thereby leading to better conversations and a stronger form of interactivity. In addition to strict planning and carefully thought out interventions, playfulness and exploration should be taken into account as some readers may access content in ways not predicted by the development team.

2. In addition to the symbolic-analytic practice of understanding the production of signs and symbols and

TABLE 3: SELECTED GUIDELINES FOR EFFECTIVE INTERACTIVITY IN TECHNICAL COMMUNICATION (2001 VERSUS 2009)

2001 guideline	What has changed?
Limits	Limits are in some senses less strict; RIA tools and online games allow for unpredictable behavior while enabling multiple paths to successful outcomes (for example, retrieving the necessary information).
Accuracy and consistency	In terms of content, these characteristics are still essential. In terms of presentation, however, consistency is no longer as important or even as desirable. Both RIA and games often depend on the user feeling as though she can personalize her informational experience and make it unique.
Trust	Trust is perhaps even more important in emerging forms of TC. The user must fully trust the rules of the system to feel empowered to make personal decisions about navigation and presentation.
Customizable hiding	Allowing the user to hide and reveal information is in some cases more important, due to the potential for information overload in a RIA, for example. In games, however, this is less important since the user can simply navigate her avatar to a new location if the in-game information is not relevant or helpful. Moving from 2D to 3D spaces has in some cases reduced the need for this feature.
Effective use of feedback	Feedback is a critical dimension of interactivity, and recent developments in technology have only confirmed this. With RIA and online games, feedback is even more important than before due to the greater degree of control trusted to the user. When traditional browser cues are removed, such as the blinking effect that occurs between the press of a form's button and the return of data from a web server, new types of feedback must be provided to the user so that she knows the interactive conversation is ongoing. Similarly, feedback in games must be rapid and continuous so as to remind the player that her actions in the game are being adequately considered and responded to.
Familiar metaphors	Design metaphors are important for connecting unknown concepts to known and familiar domains. As such, they continue to be important for emerging TC, but design metaphors have broadened from the page and book to other real-world objects and spaces. Architecture is a particularly appealing metaphor because architects must consider the usefulness and usability of structures as well as the design rules that make effective use of those structures. Moving from physical to virtual spaces with this idea has been well-documented elsewhere (Morville and Rosenfeld 2007).

how to meaningfully arrange those systems to become knowledge workers, technical communicators should also focus on procedurality, especially on understanding how rules and processes shape interactive function in different types of virtual environments. Procedural literacy, or the understanding of how rules and transformations shape content both individually and as a holistic procedural system, should be stressed in moving toward this type of thinking.

3. Technical communicators should cultivate procedural literacy in two ways. First, during design, they should consider the various rules and procedures that afford different experiences for audiences and reflect on how those rules may lead web visitors or game players to desirable outcomes (keeping in mind Point 1 above). From a more reader-centered approach, they should also consider the ways in which the rules are made clear to their audiences, so that procedural literacy can also be shaped on the reader's end of an applied product. In other words, think about rules that can be created that will enable readers to feel empowered when interacting with online information.

4. It is important to provide meaningful feedback paths. Good feedback is essential for strong interactivity, and there are various types of feedback and directions in which it occurs, for example, from a user to a computerized system, from a computerized system to a user, from a user to a content author as mediated by technology, and so on. Web 2.0 conventions should be carefully studied according to the unique design opportunities for feedback offered by a particular medium. For instance, blogs and wikis offer different types of feedback interactions than *YouTube*, but each can lead to enhanced communities of practice and provide interesting opportunities for soliciting audience feedback (blogs generally have commenting features, most Wikis have a discussion area, and *YouTube* offers a video response submission option in addition to standard comments). In the examples I discussed in this essay, AJAX offers design opportunities for a feedback of immediacy, whereas games can offer various types of feedback through player-to-computer, player-to-player, and player-to-game categories of interaction.

5. Technical communicators should strive to incorporate interdisciplinary approaches to the design of interactive technical documentation and interactive virtual scenarios. The design of complex interactive technological spaces often integrates theory from outside disciplines such as digital media, psychology, computer-aided instruction, human–computer interaction, interaction design, communication, education, and information architecture. The rhetorical expertise offered by technical communicators couples nicely with materials from these disciplines, but outside perspectives are not always considered because of lack of awareness on the part of the designer or political issues that can discourage interdisciplinary collaboration in both research and practitioner environments.

CONCLUSIONS

Although not every recent development in technology or theory supports the original assertions of the authors of *Making the Most of Interactivity Online*, the core ideas remain true. Interactivity is an important part of technical communication, and its importance has not diminished in the past 8 years. In fact, interactivity is more important than ever before. Future trends in technical communication, such as the increasing use of experience design, wiki-based documentation, content management systems, single sourcing, dynamic content delivery, and social networking (Giammona 2009), all rely heavily on the effective use of interaction. This interaction is important both during the systems' design and during their use by an audience. To express this another way, both the design of modern technical deliverables and the use of these documents by contemporary readers are practices that are becoming increasingly interactive and driven by more sophisticated types of rules.

The argument was made in this article that new theories from digital media, game studies, and technical communication can offer us additional insights for dealing with interactivity using a familiar metaphor of conversation. Technical communicators can cultivate new skills in interactive media design for online environments through the cultivation of an enhanced procedural literacy. This literacy takes into account lower-level technical rules in addition to existing conceptual understandings of both new media forms as architectural spaces and socio-organizational practices. Procedural literacy means understanding the various rules and hierarchies at work in a system or organization and recognizing how to apply or navigate those rules for productive gain. This expertise can be further honed by considering design heuristics from other disciplines such as experience design and information architecture; a small subset of these heuristics was shown in Table 1. In their continual quest to empower themselves as knowledge workers in the globalized economy, technical communicators are uniquely positioned to author the types of communicative, rule-based environments characterized by strong forms of constructive interactivity and navigable virtual space.

In the latter half of this essay, five guidelines were presented to begin discussions about how technical communicators can play a significant role in the evolution of Web 2.0 and gaming technologies as tools for structured communication. Many of these guidelines can be tested by studying information delivered with new technologies like RIA and online game-based worlds. Advances in technology continue to challenge us to broaden our understanding of rules to account for additional actions that can occur in 3D, responsive, and highly interactive environments. As procedural architects with expertise in information transformations and the design of virtual spaces, technical communicators can add value to both texts and to the workflow processes of other professionals working as programmers, managers, or graphic designers. Additional design considerations will certainly emerge as online media becomes more sophisticated, technical communication continues to explore the domains of asynchronous web delivery and online games, and new forms of social and technological feedback become available. TC

REFERENCES

Andrisani, D., A. V. Gaal, D. Gillette, and S. Steward. 2001. Making the most of interactivity online. *Technical communication* 48: 309–323.

Applen, J. D. 2002. Technical communication, knowledge management, and XML. *Technical communication* 49:301–313.

Applen, J. D., and R. McDaniel. 2009. *The rhetorical nature of XML: Constructing knowledge in networked environments.* New York: Routledge.

Asleson, R., and N. T. Schutta. 2006. *Foundations of Ajax.* New York: APress.

Atkinson, J. D. 2008. Towards a model of interactivity in alternate media: A multilevel analysis of audiences and producers in a new social network movement. *Mass communication and society* 11: 227–247.

Bernhardt, S. A. 1993. The shape of text to come: the texture of print on screens. *College composition and communication* 44: 151–175.

Bogost, I. 2007. *Persuasive games: The expressive power of videogames.* Cambridge: MIT Press.

Bühlman, V. 2003. Volatile milieus: The poetics of interactivity. In *Approaches to interactivity*, ed. Association MetaWorx, 6–24. Basel, Switzerland: Birkhäuser.

Carliner, S. 2003. Physical, cognitive, and affective: A three-part framework for information design. In *Content & complexity: Information design in technical communication*, eds. M. J. Albers and B. Mazur, 35–52. Mahwah, NJ: Lawrence Erlbaum.

Chou, C. 2003. Interactivity and interactive functions in web-based learning systems: A technical framework for designers. *British journal of educational technology* 34: 265–279.

Cooper, A., R. Reimann, and D. Cronin. 2007. *About face 3: The essentials of interaction design.* Indianapolis, IN: Wiley.

Crawford, C. 2003. *The art of interactive design.* San Francisco, CA: No Starch Press.

———. 2005. *On interactive storytelling.* Berkeley, CA: New Riders Games.

Cyboran, V. 1995. Designing feedback for computer-based training. *Performance and instruction* 34: 18–23.

Derouin-Jessen, R. 2008. *Game on: The impact of game features in computer-based training.* Unpublished doctoral dissertation. Orlando, FL: University of Central Florida.

Doheny-Farina, S. 1992. *Rhetoric, innovation, technology: Case studies of technical communication in technology transfers.* Cambridge: MIT Press.

Eichorn, J. 2007. *Understanding Ajax.* Upper Saddle River, NJ: Prentice Hall.

Eyman, D. 2008. Computer gaming and technical communication: An ecological framework. *Technical communication* 55: 242–250.

Friedl, M. 2003. *Online game interactivity theory.* Hingham, MA: Charles River Media.

Garrett, J. J. 2005. *Ajax: A new approach to Web applications.* http://adaptivepath.com/publications/essays/archives/000385.php.

Gee, J. P. 2003. *What videogames have to teach us about learning and literacy.* New York: Palgrave Macmillan.

Gehtland, J., B. Galbraith, and D. Almaer. 2006. *Pragmatic Ajax: A Web 2.0 primer.* Raleigh, NC: The Pragmatic Bookshelf.

Giammona, B. 2009. The future of technical communication: Remix. *Intercom* May 2009: 6–11.

Jana, R. 2006. On-the-job video gaming. http://www.businessweek.com/print/magazine/content/06_13/b3977062.htm.

Juul, J. 2005. *Half-real: Video games between real worlds and fictional rules.* Cambridge, MA: MIT Press.

Lawton, G. 2008. New ways to build rich internet applications. *Computer* 41: 10–12.

Lopistéguy, P., P. Dagorret, and M. Latapy. 2006. Interactive applications for communicational situations: Assets of genre and verbal interactions. In *ACM SIGDOC 2006: proceedings of the 24th ACM international conference on design of communication*, eds. R. Pierce and J. Stamey. Myrtle Beach, SC: Association for Computing Machinery, pp. 202–209.

Michael, D., and S. Chen. 2006. *Serious games: Games that educate, train, and inform.* Boston, MA: Thomson Course Technology.

Moggridge, B. 2007. *Designing interactions.* Cambridge, MA: MIT Press.

Moore, G. E. 1965. Cramming more components onto integrated circuits. *Electronics* http://www.intel.com/pressroom/kits/events/moores_law_40th/index.htm.

Morrison, C. 2007. Casual gaming worth $2.25 billion, and growing fast. http://venturebeat.com/2007/10/29/casual-gaming-worth-225-billion-and-growing-fast/.

Morville, P., and L. Rosenfeld. 2007. *Information architecture for the World Wide Web.* Sebastopol, CA: O'Reilly.

O'Reilly, T. 2005. What is Web 2.0? Design patterns and business models for the next generation of software. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

Persuasivegames.com. 2009. Stone city: Stone cold creamery. http://www.persuasivegames.com/games/game.aspx?game=coldstone.

Petroski, H. 1997. *Remaking the world: Adventures in engineering.* New York: Alfred A. Knopf.

Preece, J., Y. Rogers, and H. Sharp. 2002. *Interaction design: Beyond human-computer interaction.* New York: John Wiley.

Rouse III, R. 2005. *Game design theory and practice,* 2nd ed. Plano, TX: Wordware Publishing.

Ryan, M. L. 2001. Narrative as virtual reality: Immersion and interactivity in literature and electronic media. Baltimore, MD: Johns Hopkins Press.

Schank, R. C. 1995. *Tell me a story: narrative and intelligence (rethinking theory*). Evanston, IL: Northwestern University Press.

Sheridan, D. M., and W. Hart-Davidson. 2008. Just for fun: Writing and literacy as forms of play. *Computers and composition* 25: 323–340.

Stavely, A., L. Walsh, and J. Shipman. 2008. Lightweight literate programming: A documentation practice. *Technical communication* 55: 23–37.

Williams, S. D. 2008. 3D virtual worlds and technical communication: One more tool in the kit. *Technical communication* 55:239–241.

RUDY MCDANIEL is an assistant professor of Digital Media and is the director of the Partnership for Research on Storytelling Environments (PROSE) laboratory at the University of Central Florida. He received his MA in Technical Communication and his PhD in Texts and Technology from the University of Central Florida in 2004. He is coauthor of *The Rhetorical Nature of XML: Constructing Knowledge in Networked Environments* (Routledge 2009). He teaches courses in technical communication, project management, new media studies, information architecture, game design, web programming, and interactive storytelling. Contact: rudy@mail.ucf.edu. Visit his Web site at http://www.dm.ucf.edu/~rmcdaniel/.