A Rhetoric of Electronic Instruction Sets

Stuart A. Selber
Penn State University

This article offers a heuristic for conceptualizing the broad contours of electronic instruction sets as they have developed for and in online environments. The heuristic includes three interconnected models: self-contained, which leverages the features of fixed instructional content; embedded, which leverages the features of user-generated metadata; and open, which leverages the features of mutable instructional content. Although the models overlap to some extent, their distinctions help to illustrate the changing nature of online how-to discourse.

A computer is not a computer is not a computer. The sentiment of this adage is clear to those who study literacy technologies from a critical perspective: Significations of computers vary across time and space and reflect social forces in particular contexts and cultures. I use an instant-messenger program to chat with colleagues and students, receive routine alerts from certain Web sites, and transfer research notes from library computers to home and office computers. My decisions and actions help to instantiate the instant-messenger program as an environment for synchronous communication (the main purpose for the vast majority of users), as an RSS feed with hypertext links, or as something of an FTP client. This brief example illustrates the interpretive flexibility of designed artifacts and the constitutive role of users and their tasks (my deployment of the instant-messenger program as an FTP client amounts to a personal hack that coordinates research workflow across multiple writing spaces).

So computers can come to mean different things to different people in different contexts. No big news there. What is noteworthy, however, is the ways in which the genres of technical communication are being articulated and rearticulated on the World Wide Web. Genres, of course, are as malleable as the literacy technologies that enable their production, dissemination, and use. As Miller (1984) put it, “Genres change, evolve, and decay; the number of genres current in any society is indeterminate and depends upon the complexity and diversity of the society”
Her pragmatic claim derives from a rhetorical approach that associates the textual features and meanings of genres with the evolving domains of user activities and settings. As the activities and settings of workers evolve in the context of sociotechnical development, so too do the genres of technical communication. Spinuzzi (2003) offered an example of this phenomenon in his study of a system used in Iowa to record and manage information about traffic accidents. Through a practice he termed compound mediation (pp. 47–49), Spinuzzi showed how workers bring together texts from multiple sources and often from completely different genres to create new texts that help solve local problems. This process entails procedural breakdown, reallocation of resources, shifts in power, and new hybrid genres, including genres that remix preexisting texts. But the process is more recursive than linear: Genres shape and are shaped by the workers who employ them—and by other social forces. Neither static nor timeless, genres evolve diachronically through reciprocal interactions among a diverse array of elements in a rhetorical situation.

Rhetorical situations that involve the Web both stabilize and destabilize—articulate and rearticulate—the genres of technical communication. In the case of stabilization, the Web can support traditional approaches to texts and their contexts, playing a conservative role in the scheme of literacy theory and practice. For instance, if my job is to document a process for ISO certification or FDA approval, I will want to leverage features and resources that control for those institutionalized requirements. If my job is to teach a hazardous procedure, I will want to leverage features and resources that constrain the instructional pathways of learners. A main task for people in the field has always been to domesticate complex information spaces, to impose order, coherence, and a dominant view of work/life in settings of consequence for both individuals and cultures. Such a task becomes all the more important with the rise of user-generated content, which cannot be accessed and organized on a significant scale without the implementation of standards-based frameworks (Hart-Davidson, 2007).

In the case of destabilization, sociotechnical contexts for the Web can resituate the genres of technical communication in new and unusual ways. For example, Gonzalez-Pueyo and Redrado (2003) discussed a range of issues that emerge when scientific articles are published on personal home pages rather than in academic journals and other conventional forums. This emergent practice can expand and fragment notions of audience and position scientific articles in openly promotional environments. Although the implications of such shifts remain speculative for the moment, the impulse to self-publish is perfectly understandable, very real, and encouraged by open-source movements in the sciences. For these reasons, the implications are likely to be significant and manifold. In a similar research project, Killoran (2006) investigated communication dynamics allied with résumés that have been published on the Web by people who are self-employed. He argued that sociotechnical developments can reframe the contexts for résumés by linking them
not with cover letters and job ads, but with revenue ads, business cards, and other marketing materials. A conclusion for Killoran is that Web résumés can satisfy purposes other than those accomplished by print résumés (e.g., client-seeking versus employment-seeking). In the studies by Killoran and by Gonzalez-Pueyo and Redrado, destabilizations are evident in areas as fundamental as audience, purpose, persuasion, and ethos.

In a similar vein, this article examines sociotechnical dynamics that inform and shape the contours of instruction sets as they are created and destabilized in online environments. It begins with an overview of the role of instruction sets in technical communication, arguing that designers of Web 2.0 environments have appropriated this genre in distinct and expansive ways. The main portions of the article articulate and explain three models that have come to organize electronic instruction sets: self-contained, embedded, and open. These models are more complementary than competitive and can overlap and blur together. In addition, they primarily serve a heuristic function for the field, helping to map the shifting landscape of online how-to discourse. The conclusion discusses implications of the models for research and pedagogy.

**CONTRIBUTIONS OF INSTRUCTION SETS TO TECHNICAL COMMUNICATION**

As a species of technical communication, instruction sets provide step-by-step procedures for accomplishing a physical or mental task. My formal educational training consistently involved the instruction set, a priority for the field in historical terms, and perhaps its most obvious and visible hallmark. Most of my courses at the three universities where I was a technical communication student incorporated units and projects on how-to discourse. Memorable assignments asked us, as students, to document features of an early hypertext program and contribute entries to an instructional handbook on style. The internship and portfolio required for my MA degree reflected the values of a professional program with large stakes in the success of the computer industry. And the pedagogical preparation I received as a new teacher used the instruction set to distinguish technical communication from composition studies: In 1988, and beforehand, there seemed to be little interest in the shared ground of first-year composition and advanced composition courses in business and technical communication. In fact, scholars in the field used conventions and contexts for the instruction set to argue for the unique relevance and distinct character of technical communication (for emblematic examples of this argumentative stance, see Britton, 1978; Tebeaux, 1988).

On one level, my prolonged and constant contact with the instruction set could be viewed as an artifact of my educational time frame (1983–1994). In that period, the field was consumed by the challenges and opportunities of software and hard-
ware documentation. However, I also want to suggest that the instruction set has made—and continues to make—novel contributions to technical communication, and for this reason it warrants ongoing attention and study. In a quick fashion, let me review several of these contributions here; they are relatively obvious and do not require extensive explication. On the academic side of the equation, one contribution can be found in scholarship that has helped to evolve a philosophy of language with disciplinary rationale for technical communication. As with any new field, applied or otherwise, inceptive work in technical communication was often more descriptive than analytic and more general than specific. This was not a flaw or problem but a predictable first step in the growth of a discipline, one that paved the way for future waves of research. If Miller (1979) made the general case for a humanistic approach that acknowledges the productive and constitutive capacities of language and language use, Harris (1983), Dobrin (1985), and Redish and Schell (1989) considered specific ways in which discourse operates to shape meaning making and sense making in technical communication. In the context of how-to instructions, these researchers mobilized semiotic theories and usability studies to begin to illuminate concrete practices in technical communication that influence the conduct, content, and rationalization of science and technology. In this manner, the instruction set was instrumental in efforts to achieve professional status in the academy. Early on, it functioned as an exemplar of a new research trajectory that was worthy of study by humanities scholars and teachers.

Another contribution of the instruction set is visible in the maturation of the practitioner side of the field. Connors (1982), Souther (1989), Kynell (1996), and other historians have noted that World War II, “the first truly technological war” (Connors, p. 340), multiplied by an order of magnitude the need for people who could explain how to operate expensive and intricate machinery. According to Connors, this critical need carried over in the postwar period “as wartime technologies were translated into peacetime uses” (p. 341) at prominent companies like Westinghouse, General Motors, and General Electric. The dawning age of ubiquitous high-tech products helped to create a meaningful role for instruction sets in consumer-oriented communication. In a sense, so too did government requirements for comprehensible environmental impact statements (Souther, p. 2), which often employ detailed process descriptions, a close cousin to the instruction set.

In a more contemporary context, instruction sets have become a fixture and a focus of online participatory culture, which illuminates the significance of technical communication to an ever-widening audience of authors and users. By online participatory culture I mean the activities and practices in social spaces on the Web that encourage the production and distribution of user-generated content (e.g., uploading videos to YouTube and photos to Flickr; writing and rating product reviews at Amazon; adding and editing encyclopedia entries at Wikipedia; sharing and tagging collections of bookmarks at Delicious; and offering advice in user forums at Apple or Microsoft). At present, it seems that nearly everyone on the
Internet is a technical communicator—or at least has the potential to be one. The sociotechnical interfaces that organize literate activity today are inclusive and remarkably flexible. These interfaces no longer position data and information—or people, for that matter—in one context or another. Nor do they care very much about the boundaries the field has used to define technical communication. Although the range of user-generated content is extensive and includes a wide variety of materials (new media and not), instructional discourse occupies a conspicuous position in the landscape of online participatory culture. At the previously mentioned Web sites, which incorporate so-called “Web 2.0” features, there is no shortage of how-to texts, images, and videos of both an official and vernacular nature; these items have been produced by amateurs and experts who confound distinctions between subject positions (or audience categories) and between elements in a mixture of additional binary oppositions that have come to organize Western culture (e.g., private/public, work/play, and literacy/technology). In addition, Web 2.0 sites such as Expertvillage.com, Instructables.com, and Docstoc.com have been specifically designed to support those who gravitate toward the activities and practices of participatory culture. Sites like these—there are many of them—host hundreds of thousands of instruction sets framed with metadata and mechanisms for various forms of feedback (more on this later).

So history and contemporary practice attest to the significance of how-to discourse in consequential settings over time and across disciplinary contexts. But I want to emphasize the social spaces of Web 2.0 environments because they have begun to recast the instruction set in concrete and meaningful ways. The relevance of the instruction set has been amplified and widened by an online participatory culture that encourages involvement, collaboration, and information exchange. More than simply a good example, the instruction set has become a metonym for the complex world of Web 2.0. Although such a part-for-whole substitution is certainly reductive, it has heuristic value in that it helps people understand a role and function for user-generated content—a phrase with no shortage of interpretive flexibility. In other words, the sharing of expertise, which is an easily understood and frequently practiced form of human discourse, has become an archetypal task of online engagement and interaction. With little disciplinary effort, a genre from technical communication, the instruction set, can be seen as central to an age of social media.

MODELS OF INSTRUCTION SETS IN TECHNICAL COMMUNICATION

My rhetoric of electronic instruction sets is organized by three conceptual models: self-contained, embedded, and open. The models relate to each other in a number
of salient ways, which I will elaborate in a subsequent section, and they exist concurrently as representations of equally valid approaches. For analytic purposes, the discussion in this section divides the models and provides working definitions and examples. The self-contained model should be familiar to teachers and practitioners, for it reflects conventional wisdom in the field about the shape, character, and role of instruction sets. The embedded model accounts for sociotechnical components in Web 2.0 sites that invite people to contribute metadata for use by authors and others. The open model accentuates hypertext capabilities that enable users to assume authorial functions in Web sites that host instruction sets. My discussion includes two examples for each model. The examples illustrate distinguishing features of the models and their supporting structures (see Table 1).

### Self-Contained Instruction Sets

As the name implies, instruction sets guided by this model are tightly bounded and controlled. Users encounter procedural content that is fixed, static, and absolute. Although any text can be multiply interpreted, the self-contained instruction set aspires to be unambiguous and comprehensive: People should not need other texts or resources to complete the task at hand. This usability objective is assisted by rhetorical techniques that establish information spaces with explicit hierarchical and relational structures. Farkas (2005) identified headings and links as the primary means by which technical communicators impose lines of dominance in functional texts. Lay (1989) reviewed elements of layout and design—unity, balance, proportion, emphasis, and sequence—that are instrumental to comprehension; she focused on the physical and metaphorical properties of pages and screens and how those properties can be leveraged by technical communicators. Bell (1991) considered an array of rhetorical approaches that help users process information: document templates, logical structures, text sequences, figurative forms, and the like. Rhetorical techniques like these (and others) inform guidelines for creating instruction sets, and these guidelines are presented in every textbook in the field. Textbook authors, for example, encourage technical communicators to define terms and provide advance organizers; use numbered lists, parallel grammatical structures, and the imperative voice; separate steps from each other and user ac-

### Table 1

**Models of Electronic Instruction Sets**

<table>
<thead>
<tr>
<th></th>
<th>Self-Contained</th>
<th>Embedded</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distinguishing Features</td>
<td>Fixed content</td>
<td>User-generated metadata</td>
<td>Mutable content</td>
</tr>
<tr>
<td>Supporting Structures</td>
<td>Syntactic Web</td>
<td>Semantic Web</td>
<td>Organic Web</td>
</tr>
<tr>
<td>Examples</td>
<td>PDF files</td>
<td>Screencasts</td>
<td>Knowledge bases</td>
</tr>
<tr>
<td></td>
<td>Static Web sites</td>
<td>User forums</td>
<td>Wikis</td>
</tr>
</tbody>
</table>
tions from system reactions; employ visual devices to explain and reinforce key aspects of tasks; and signal the completion of tasks. Such guidelines attempt to regulate technical communication activity from the top down.

In online environments, self-contained instruction sets assume two common forms: the portable document format (PDF) equivalents of print documents and static HTML-based Web sites. Needless to say, both are anywhere and everywhere on the Internet. Figure 1 shows a standard PDF-based instruction set, which I downloaded from Garmin.com, a Web site that supports users of Garmin navigation devices (I own the Forerunner 305, which is a personal GPS device for running and biking). There are several manuals associated with the Forerunner 305; the screenshot in Figure 1 is from a guide that shows people how to work the heart-rate monitor. It includes three pages of instructions in 11 different languages.

![Figure 1: Screenshot from an instruction set at Garmin.com. PDF files tend to function as self-contained instruction sets.](image)
In a default space of 245 × 446 pixels, the guide covers how to position the monitor on your body, “pair the sensors” so that the sensors on the watch and chest strap can “speak” to each other, and replace the battery in the chest strap. The guide also lists technical specifications and warranty information. I received a print version of this very guide when I purchased the device. For the most part, the affordances of the self-contained, PDF-based instruction set match those of the print edition. The only real difference relates to delivery: Users can download endless copies of the PDF file and view it on electronic devices, and users can activate an external hypertext link, which points to government information on how to dispose of hazardous waste batteries.

The self-contained instruction set as static Web site is similar and capable of being more or less identical to a PDF version. Figure 2 shows a standard example from Apple.com; it teaches people how to create audio podcasts with QuickTime 7 Pro on Mac OS X. This documentation is not bundled with the purchase of a new

![Creating Audio Podcasts on Mac OS X](image)

**FIGURE 2** Screenshot from an instruction set at Apple.com. Static Web sites tend to function as self-contained instruction sets.
computer or QuickTime. Although QuickTime comes with numerous help files, users access this particular instruction set via the Web, either by working through a help tree at Apple.com or by using the search engine (the help tree includes a categorized index screen with hypertext links to 18 discrete tutorials). In six steps, the instruction set covers the basics of getting started with podcasting, capturing audio content, and exporting audio content to an appropriate file format. Although this example includes hypertext links to other pages internal to the Apple Web site, those pages contain information that is supplemental to the task of creating a podcast. The figures associated with five of the six steps are static images that depict user interactions with the QuickTime interface.

As the examples in Figures 1 and 2 indicate, first-generation Web technologies constitute the supporting structures for self-contained instruction sets. These instruction sets are developed with a simple markup language (HTML and its variants and extensions) that controls the appearance of information by defining content—in a detailed manner—at the syntactic level. Although the information in my first example is controlled by PDF (which supports external hypertext links), the distribution of the file is dependent upon an HTML framework: Users download the instruction set from a Web page that describes and archives the file. The design elements in my second example—hearings, figure locations, section separators, colors, links, and more—are all coded in HTML. In both examples, users have limited command over design representations; they are determined, for the most part, by the mechanics of the design syntax (I say for the most part because users can make modest if useful adjustments with their Web browsers). Put differently, the syntactic Web defines the field of possibilities for self-contained instruction sets.

Embedded Instruction Sets

The model informing embedded instruction sets underscores the framing apparatuses of Web 2.0 sites. On one level, embedded instruction sets tend to mirror self-contained instruction sets: Strictly speaking, their content is also fixed and immutable, and it is informed by the same sorts of rhetorical techniques that shape self-contained instruction sets. The content of embedded instruction sets, however, is delivered in an environment that enables the creation and collection of user-generated metadata, the distinguishing feature of this model. Metadata is not a new concept or practice in computing contexts. In computing contexts, there has always been a need to have structured data about data, to produce and collect information that can help people administer and maintain technological systems (Baca, 2008). Although this need became critical with the advent of database management applications, it is all the more important and advantageous with the Web, which is open to a wider range of audiences, purposes, and activities. It is not difficult to identify design practices for metadata that are immaterial to users: documenting HTML code and keeping server logs, for instance (the former assists de-
signer memory and collaboration while the latter assists Web site revision). But Web 2.0 technologies raise mechanisms for metadata to the level of the interface and integrate them into user experiences. My examples of embedded instruction sets—screencasts and user forums—will clarify these mechanisms.

Screencasts are video-based demonstrations that explain how to perform a task or series of tasks. In one popular variety for computing tasks, demonstrations are created by recording the movements associated with human interface actions (e.g., clicking icons, selecting commands from menus, and completing dialogue boxes). If you, as a technical communicator, wanted to explain how to upload a photo to Skype, for example, you could perform this task yourself and capture the cursor movements as a video. You could then edit the video to include voice-over narrations, titles, transitions, and various effects. Another popular approach mixes captured cursor movements with instructor-led segments. In this method, user attention is directed back and forth between instructional portions that show and some that tell. Many organizations offer sizeable collections of screencasts: For instance, Brown University has produced dozens of them for both commercial and institutional software. But I would not characterize their screencasts as embedded instructions sets. Although users can rate speed and quality and make comments, this metadata is submitted directly to Brown and is not available to users. I am interested in environments like Techscreencast.com, Ehow.com, and TeacherTube.com, which position metadata as a key component for users. Figure 3 begins to illustrate the ways in which embedded instruction sets integrate metadata. It is a screenshot from a screencast on how to employ new interface features at Screencast-o-matic.com, a Web site for making and hosting screencasts. Users can rate the screencast and search by user ratings; recontextualize the screencast by embedding its code in another Web site (people often embed screencasts in blogs and then discuss them); add notes to help others interpret the instructions and navigate the screencast (you can fast forward and rewind as you watch); filter notes to see only those added by other users; and leave comments. In addition, some Web 2.0 environments allow users to add or suggest tags, post screencast responses, see Web sites that link to a screencast (referrers), and flag inappropriate material.

User forums can also be understood as a type of embedded instruction set, one that emphasizes rating and commenting functions and amplifies user-to-user conversations. User forums are electronic discussion spaces in which end users ask and answer questions and provide feedback to artifact designers. They exist for nearly any type of product or service, and for companies they constitute a mechanism for leveraging consumer knowledge and accumulating usability insights. The user forums I focus on tend to occupy a different time-space frame than screencasts. The metadata functions for screencasts are internal to the Web pages that host embedded instruction sets. From a user perspective, metadata functions are parasitic to screencasts, appendages, really, and thus occupy the same attention field (as in the same computer screen). In contrast, user forums inhabit a parallel
workspace not always associated in explicit ways with the targeted instructions. Figure 4 illustrates this configuration; the screenshot is from a documentation forum for Debian, an open-source, Linux-based operating system. In this forum, users have produced thousands of posts that amount to a rich, searchable database of how-to corrections, elaborations, and additions. Many of the additions include original instruction sets developed in response to questions about missing or confusing information in the Debian documentation. But there are not any technical mechanisms that unite the metadiscourse in the forum with the how-to discourse in the manual. Instead, users supply hypertext links, cite other user posts, or cite section titles and page numbers from the manual. Because of the loose linkage between the manual and the metadiscourse, users often ignore the documentation altogether and rely on the forum as a primary source of how-to instructions. In those situations, which are quite common, the Web 2.0 features that become prominent are those associated with searching the forum and learning about its participants (by way of user profiles and statistics).

Embedded instruction sets are supported by the semantic Web, a phrase that refers to an assortment of Internet-based structures for data management, collection, and integration. The common denominator of these structures is a computational
framework for machine-readable data, for data that can be acted upon by computers. Those interested in the semantic Web want computers to be more helpful to people, especially as they use and interpret data and relate data from dissimilar sources (Berners-Lee, Hendler, & Lassila, 2001). This endeavor employs technologies like XML and RDF, which describe and categorize informational content in ways that enable computers to mine and manipulate it. Although the semantic Web seems to be more promise than reality right now, it has begun to become crucial to the organization and management of user-generated content. For instance, small pieces of descriptive code permit Web site designers to add semantic meaning to HTML tags. These pieces of code, called microformats, offer consistent structures
that specify the nature of the content being marked up (see Microformats.org). In essence, microformats are standards that enable data to be accessed and reused across the Web. There are numerous microformats available to designers. The VoteLinks attribute transforms basic hypertext links to links that function as polling mechanisms with aggregate tracking pages (both Technorati.com and Delicious.com employ the VoteLinks attribute). The XFN attribute uses hypertext links to denote human relationships that can be tracked by search engines (like Rubhub.com) that support microformats (Wordpress, a popular blogging platform, allows users to denote authors of linked-to blogs as colleagues, coworkers, friends, neighbors, family members, and more). And the hReview attribute provides a common set of elements for user reviews that allow reviews to be accessed and aggregated by search engines (like Google.com) that support microformats (Tech.yahoo.com is a technology review site that relies on hReview to organize user reviews). There are microformats for marking up contact information, calendar information, résumé information, syndicated information, and other popular forms of information in Web 2.0 sites. My examples of embedded instruction sets do not rely solely on the capabilities of the semantic Web, but these are increasingly vital to the task of leveraging user-generated content as metadata.

Open Instruction Sets

The third model emphasizes sociotechnical features that encourage users to become authors and editors of instruction sets. The concept of mutable functional texts is not a new or novel one, in theory or in practice: Early understandings of hypertext imagined end-user mechanisms for editing hypertext systems (Conklin, 1987) and pre-Internet applications like Storyspace allowed for the creation of “constructive” hypertexts in which anyone could contribute to the shape and content of a system (Joyce, 1988). Also in these years, aeronautical engineers at Boeing developed an open hypertext application to support collaborative design activities (Malcolm, Poltrock, & Schuler, 1991), while software engineers at Hewlett-Packard used hypertext to create libraries of documented computer code that could be altered and reused by any member of a working group (Creech, Freeze, & Griss, 1991). These projects and others explored design requirements for online environments that diminish the physical and metaphorical distances between authors and users, enabling users to become active contributors with editing and publishing privileges. In the process, investigators started to develop responses to ongoing questions for technical communication: What might be productive about mutable functional texts? Unproductive? What are the social, cultural, and organizational requirements for texts that can be changed by users? Do such texts call for alternative composing processes? Alternative collaborative processes? Initial responses to these questions, of course, reflected the delivery mechanisms of local area networks, which constrained access to mutable texts to those
within an organization or institution. But wide area networks and the Internet expanded access from colocated users to distributed users. With the formation of the Internet and various collaborative software environments, distributed users could work together on the construction and reconstruction of mutable functional texts. In the context of instruction sets, knowledge bases and wikis completed the online infrastructure.

Knowledge bases are dynamic help systems that depend on the willingness of volunteer users to contribute their time and expertise to a shared resource for problem posing and problem solving. In knowledge bases, information is not developed through one-to-many communication patterns that function in a traditional hierarchical manner. Although experts are critical to the enterprise, people assume that users of all levels can offer valuable perspectives and insights. This many-to-many approach is an exercise in distributed usability, but knowledge bases represent more than just that. In addition, they are emblematic of a full process approach in that entries are always susceptible to modification and revision. In a very real sense, then, nothing in the system is ever concluded or closed off. Figure 5 is a screenshot from the knowledge base for Firefox, a popular Web browser; it shows an entry on how to reset preferences. Any user can submit feedback on this instruction set, but, as the Actions heading indicates, users who become contributors (by simply making a free account) have two other options for participation: They can edit the instruction set or translate it into another language. Contributors can also add new entries to the knowledge base and request new entries for others to write. All this activity, it should be noted, is moderated to achieve quality assurance: Revisions and requests are reviewed by experts for accuracy and compliance with the documentation style guide. Experts, however, are nothing more than users with a good track record—or new users who know experts that can vouch for them. As with other volunteer efforts, knowledge bases maximize sweat equity and goodwill.

Wikis are editable Web sites that share certain features and philosophies with knowledge bases. Yet wikis can also be distinguished from other open environments in at least one unique way: As collaborative systems that encourage users to write and edit content, wikis keep track of user changes and provide interfaces for understanding those changes and responding to them. Wikis put a premium on the ability to display modifications at different levels of temporal granularity. One can see modifications—minor and major—that have been initiated recently and not so recently. One can even see changes that have been made over the lifetime of texts. Furthermore, wikis make it possible to compare two different versions of a Web page; the software aids this task by providing line-by-line, color-coded highlights that point up variations. Figure 6 is a screenshot from an instruction set on how to make a temporary laptop bag out of cardboard. It is hosted at Wikihow.com, titled “The How-to Manual That You Can Edit.” Wikihow.com hosts more than 60,000 instruction sets on every imaginable topic, which helps to explain why contribu-
tors are encouraged to follow a somewhat predetermined format. More than not, my example adheres to this format: title (in how-to form), introduction, procedural steps, video (not required), tips, warnings, materials list, links to related instruction sets, and sources and citations. But the unique aspects are associated with the items in the tabbed navigation bar. Users can discuss the instruction set in a discussion forum, edit the instruction set, view a history of the edits, and compare edited versions. Users can also indicate that they want to be notified when changes are made to the instruction set. As of December 2009, this instruction set has been edited by eight different contributors and accessed more than 88,000 times.

The supporting framework for open instruction sets is the organic Web. This phrase does not enjoy widespread usage in cyber-elite circles, but it does convey the sense that certain instantiations of the Web function like living, breathing...
structures, constantly being adapted and changed by the motives and perspectives of users. It also connotes the bottom-up process of content creation that has been encouraged by mutable environments. The original graphical Web browser (Mosaic) actually included a program for making Web pages; it was both a writing and reading machine. In an interview with Mark Lawson (2005) of the BBC, Tim Berners-Lee, who created the first Web site, commented that his original vision involved features that aimed to make the Web a creative, productive space for all users. Such features, however, became relatively complex (and powerful) as Web design programs evolved into full-featured, stand-alone applications like Microsoft FrontPage and Adobe Dreamweaver. For many users, access to the writable Web did not (re)occur until simple composition tools were reintegrated into browsers and emphasized by wikis and various other Web 2.0 developments. This low-tech approach demanded little in the way of new technologies: The original wiki engine (WikiWikiWeb) was a CGI (common gateway interface) program written in Perl (Louridas, 2006). The software architectures for most wikis and knowledge bases comprise application servers, file systems, relational databases, and other mature components of software engineering. So, in terms of sociotechnical change, the onus seems to be less on inventing novel capabilities and more on

---

**FIGURE 6** Screenshot from an instruction set at Wikihow.com. Wikis tend to function as open instruction sets.
constructing philosophies and practices that are sympathetic to the communicative nature of open instruction sets.

**JUXTAPOSING AND CLARIFYING THE MODELS**

The previous section segregated the models in order to define their component parts and illuminate distinguishing features. This section orchestrates a conversation between and across the models, a conversation that examines various socio-technical relationships as well as ruptures in the boundaries. But first I want to stress several important points about this project. My objective is not to provide definitive models or prescriptive recommendations for electronic instruction sets. Such an objective would be premature and in many ways counterproductive (generally speaking, prescription functions best when it follows a thoughtful and thorough research agenda). The models, instead, should be considered to be heuristic in nature, for their structures, examples, and elaborations help map an emergent landscape and elucidate rhetorical dimensions. Also, this project does not value one model over another or posit a linear trajectory of progress from self-contained to open environments. Because technical communication settings are varied, complex, and continuously unfinished, genres tend to fragment and multiply rather than contract. It seems self-evident to predict that the need for conventional documentation will persist. To finish, one could easily locate counterexamples that defy my categorizations. Screencasts, for instance, can lack functions for user-generated metadata; knowledge bases can furnish vendor-supplied content for users to comment on, especially in the context of closed-source initiatives (the example I use is from an open-source project); and wikis can be secured from the public Internet. As with my categories, the examples are meant to be instructive and illustrative, not absolute.

Table 2 presents a set of heuristic categories with conceptual vocabulary for juxtaposing and clarifying the models. The table includes four dimensions in which to imagine the territories of self-contained, embedded, and open instruction sets: metaphors, modes, activities, and emphases. Although metaphors abound for all aspects of the Internet, three are particularly relevant to the purposes here. The contexts for self-contained instruction sets tend to be organized around a document metaphor. Appropriated from the realm of print, this metaphor is ubiquitous and certainly not a hindrance to novice learners. In design and communication fields, it is accepted practice to use metaphors to bridge both representational and functional deviations in literacy technologies (Selber, 1995). The obvious example is the desktop interface, which of course incorporates the document metaphor. The self-contained instruction set as document leverages deeply ingrained understandings of print, preserving conventional information structures and familiar notions of authorship. In this few-to-many communication approach, knowledgeable
writers produce effective documents, and lay readers consume them with success. The electronic instruction set is a well-wrought urn—at least until the product or process changes. In contrast, the contexts for embedded instruction sets tend to be organized around a database metaphor. No longer only a file server for completed texts, the Web also becomes a searchable warehouse of user-generated metadata. This variously structured warehouse, which on some level can be accessed by computers as well as by people, constitutes a rhetorical back end for fixed and immutable instructions. In metaphorical terms, the front end (the instruction sets) and the back end (their metadata functions) couple with each other to shape user experiences in real and significant ways. The database as embedded instruction set values dynamic sociotechnical feedback loops between these mutually reinforcing components. So, too, do open instruction sets, but their contexts tend to be organized around a platform metaphor, which further diminishes the distance between production and consumption. Open instruction sets (and certain embedded instruction sets) are not created in one environment and then used in another. Rather, their software supports key aspects of both development and use. As a platform for computer-based work, therefore, open instruction sets combine two sides of the same communication coin.

Modes and activities, the next two rows in Table 2, should be discussed in concert, for modes help to determine what users are able to do and imagine. Self-contained instruction sets function in a read-only mode. To use jargon from computer culture, this model can be classified with “WORM” media, as in Write Once, Read Many. Although self-contained instruction sets support user actions, only one interpretation of a task is typically accommodated. In technical communication circles, this interpretation is arrived at through rhetorical approaches to discourse and operationalized by information design grammars and presentation technologies. Users can, and do, hack the monolithic character of self-contained instruction sets—for example, one could download a static Web page, revise the HTML, and repost the page to the Internet. But these changes do not impact the original text or context, and they might not even be noticed by others (the original is unlikely to reference the remix, though the remix could certainly point to the original). Embedded instruction sets could be employed in a read-only mode: Web 2.0 sites do not require users to involve themselves in a more direct manner. However, what

<table>
<thead>
<tr>
<th>Metaphors</th>
<th>Self-Contained</th>
<th>Embedded</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modes</td>
<td>Readable</td>
<td>Extensible</td>
<td>Writeable</td>
</tr>
<tr>
<td>Activities</td>
<td>Action</td>
<td>Reaction</td>
<td>Interaction</td>
</tr>
<tr>
<td>Emphases</td>
<td>Publication management</td>
<td>Content management</td>
<td>Knowledge management</td>
</tr>
</tbody>
</table>

### Table 2
Characteristics of the Models
differentiates embedded instruction sets is an extensible architecture that accommodates user reactions to authorial interpretations of tasks. The idea of extensibility in computing contexts dates back to the 1960s. In historical terms, it did not take forever for software developers to begin to value the prospect of technologies that could be augmented by peers (this view surely adumbrates open-source movements and online participatory cultures). Notwithstanding the social and technical challenges to extensible programming (see Wilson, 2004–2005), extensibility as an interface affordance has burgeoned with Web sites that support user-generated content. Users encounter a range of embedded options for reacting to formalized instructions, from quick polls to text fields that accept extensive comments. In active Web sites, these options can recontextualize original interpretations of tasks in a dynamic and continuous fashion. Open instruction sets broaden such capabilities to include write-mode settings. In a Write Many, Read Many (WMRM) mode, people can interact with instructions by physically rewriting them to align with other explanations of tasks. This deep level of interactivity encourages a steady practice of reinterpretation and revision. In open instruction sets, no task representation is immune to modification.

A final dimension for juxtaposition considers the management emphases of the models. Self-contained instruction sets are products of mature approaches to publication management, an expansive domain that includes a vast array of practices and concerns. In their edited volume on the subject, Allen and Deming (1994) recognized not only managerial issues but also cultural, ethical, legal, and interpersonal complexities that impinge upon the organizational work of technical communication. These complexities became increasingly discernable to the field throughout the 1980s, which was a watershed period for technical publications management (Souther, 1994). In this decade, processes for product and documentation development began to parallel each other. No longer a late addition to the product development cycle, technical communication was understood to be too complex and valuable to leave until the end. For both documents and products, the process approaches that prevailed start with analysis and planning, continue with design and testing, and end with delivery and maintenance; many of these phases, of course, are rough, recursive, and not so easily separated. In terms of instruction sets, the result is a coherent document artifact that is finalized upon publication and not revisited until there is a need for version 2, which is a major revision that replaces version 1. If the process is recursive, the product is not. The fixed content in embedded instruction sets is also managed with a publication process, but the emphasis on metadata introduces a role for content-management systems. In technical communication, content management is often exemplified by the single-sourcing movement. Using markup languages, database structures, and various output technologies, the methods of this movement modularize document components into objects that can be combined and recombined for different audiences, purposes, and contexts. Although people can produce and deliver immutable func-
tional documents with single-sourcing methods, the focus is on managing a rich database of technical communication content. For embedded instruction sets, a part of that content is metadiscourse about discourse and is user generated rather than author generated. Open instruction sets benefit from techniques for both publication management and content management. Their mutable nature, however, creates a need to manage the knowledge of disparate and distributed author-users. The concept of knowledge management is multidimensional, and it has been appropriated in different ways across the disciplines. One assumption of most approaches is that all too often human knowledge remains implicit and unshared in settings that could profit from collective intelligence. Although technical communicators have always defined themselves as user-advocates, capturing and leveraging user knowledge is a relatively recent interest for the field. Open instruction sets instantiate the principles of knowledge management and illustrate the challenges of designing and running alterable documentation systems that support functional tasks.

This last dimension, presented in Table 2, helps to show how the models interrelate. If one reads across Table 2 in either direction, it becomes evident that the items are more additive than substitutive. Content management involves aspects of publication management, and knowledge management involves aspects of both content management and publication management. Ordered from self-contained to open, the models become ever more complex in rhetorical terms. This relationship also holds for the rows for modes and activities: Open instruction sets support user actions, reactions, and interactions, and they tend to be extensible as well as writable. Embedded instruction sets support user actions and reactions, and they are extensible and readable. The metaphors for the models interrelate in at least two ways. On the one hand, they scaffold like the other items in the table: Work platforms can encompass databases, which can encompass documents. The nested domains of knowledge-management systems compound the number of technical communication issues by an order of magnitude. Likewise, technical communicators who work on content management systems must design both documents and databases—an expanded scope of engagement and obligation, to be sure. On the other hand, the metaphors could be considered to be parallel structures that offer different perspectives on the same project. To this extent, such metaphors reinforce the fact that sociotechnical forces shape electronic instruction sets in multiply valid ways. The models conceptualize three of the more prominent articulations.

IMPLICATIONS AND CONCLUSIONS

The rhetoric presented in this article offers a heuristic framework for conceptualizing what happens to the instruction set when it moves online. On one level, very little seems to happen: The self-contained instruction set is alive and well and see-
ing plenty of good use on the Web. In fact, it is hard to conceive of a technological world in which people no longer need immutable how-to documents—in print or electronic form. On another level, however, instruction sets have evolved and diversified in real and tangible ways. They now take the shape of screencasts, user forums, knowledge bases, and wikis, and there are hybrid versions that combine sociotechnical aspects from several different systems. In digital contexts, instruction sets have become much less monolithic and much more complex and variegated.

What might this change mean for technical communication? What are the implications for research and pedagogy? There are a number of areas that should be attended to by researchers. For example, because there are now multiple models to select from, when does it make sense to use a self-contained versus embedded versus open approach? What about hybrid approaches? By what criteria does a technical communicator make this decision? These are the broadest and most crucial lines of inquiry for the field, which would benefit from the development of rhetorical heuristics that can lead to defensible answers in particular situations. There are also research questions associated with the embedded and open models. Embedded instruction sets are often supported by environments that employ generic Web 2.0 features: polls, comment boxes, lists of referrers, and the like. Although these features seem useful, they were not designed specifically for the contexts of instruction sets. Do these contexts demand other sorts of mechanisms for metadata? What types of user responses would be valuable to cultivate and capture? Open instruction sets pose a key question related to collaborative authorship: What is the role of the technical communicator in environments in which functional texts are mutable and always subject to social transformation? These questions are empirical and rhetorical, and constitute a significant research agenda.

Pedagogical implications are related to the fundamental research questions that need to be addressed. Because there is no research base to draw on and demonstrate, current textbooks do not discuss embedded or open instruction sets, even though they are widespread and common to the Web. I examined the chapters on instruction sets in two established textbooks (Lannon, 2008; Markel, 2010) and in two new textbooks in their second editions (Dobrin, Keller, & Weisser, 2010; Johnson-Sheehan, 2007). Dorin, Keller, and Weisser (2010) mention writing online help pages and the fact that videos and animation sequences are used on the Web to communicate how-to procedures. Johnson-Sheehan (2007), Lannon (2008), and Markel (2010) make mention of online instructions but does not provide guidelines for how to write them or distinguish between various approaches. Again, I do not blame the authors for this state of affairs. I consider all the textbooks to be excellent and reflective of current research in the field. But this is a major instructional gap that has consequences for students.

To conclude, let me simply mention an implication for the status of the profession. If the self-contained model embodies received assumptions about the nature of technical communication, the embedded and open models suggest challenging
new directions. It is fair to say that the instruction set has been animated in new ways by the activities of online participatory cultures and by other sociotechnical developments associated with Web 2.0 environments. But although technical communicators can claim expertise in this major area of Internet life, the competencies required to design and manage it are tentative and less than clear. Clarifying those competencies would contribute in central ways to the visibility and recognition of the field.

ACKNOWLEDGMENTS

I wish to thank Rebecca Wilson Lundin, Michael Faris, and two anonymous reviewers for their helpful comments on a draft of this article. My research for this project was supported by a faculty fellowship in Education Technology Services at Penn State. At ETS, I received valuable feedback on ideas in development from Cole Campless, Elizabeth Pyatt, Mary Janzen, and Erin Long.

REFERENCES


Stuart A. Selber is an associate professor of English at Penn State. He is a past president of ATTW and CPTSC.