APPLIED THEORY

SUMMARY

- **Examines the role and value of the technical communicator as creator of knowledge and focuses especially on the role's internal value**
- **Offers an expanded value proposition for technical communicators and examines its practical implications**

Moving from Information Transfer to Knowledge Creation: A New Value Proposition for Technical Communicators

MICHAEL HUGHES

"What do technical communicators do?" asks the ambitious question on the Society for Technical Communication's FAQ Web page (STC 2001). The answer lists typical job titles for technical communicators and then says, "All these professionals take technical information and make it understandable to those that need it." This description is consistent with the way that many technical communicators describe their role, that of transferring information from those who have it (subject matter experts or SMEs) to those who need it, and they define the value of the technical communicator as packaging that information to be more accessible and more readily understood by the user. Figure 1 illustrates this definition.

Although this image of technical communicators certainly provides a useful picture of part of what they do and part of the value they add, it has three limitations:

- It relegates technical communication to the *information* domain versus the more valuable *knowledge* domain.
- It implies that the source information "exists" and someone "has" that information.
- It places the technical communicator's entire value proposition on the requirement that end users read the documentation.

Technical communicators often do more than that. As stated by Henry (1998),

> *. . . when seen as wordsmiths who package other people's content rather than as professionals who tap many kinds of expertise to yield better content, communicators' 'added value' is typically underestimated. (p. 207)*

This article first reviews the current literature that addresses the value of the technical communicator. Whereas those discussions focus on what is delivered to the user (reader), this article examines the value the technical communicator adds by creating organization (internal) knowledge. The article then examines the philosophical underpinnings that support any discussion of knowledge and defines the role of technical communicators as creators of knowledge. Finally, it offers an expanded value proposition for technical communicators and examines its practical implications.

CURRENT VALUE PROPOSITION

Most of the discussions of value and technical communication in the recent literature focus on financial impact associated with the end product, either in terms of what costs are saved (or additional revenue generated) by having the product "out there" or how the costs to produce it can be reduced. In other words, the typical discussion of value examines to what degree the money invested in producing quality user documentation provides an attractive return on investment (ROI). For example, Redish's flagship article in the special issue of *Technical communication* on how technical communicators add value (1995) focuses on identifying measurements that can be applied to ROI calculations. From the perspective of user value, Redish stresses user satisfaction or improved performance. From the perspective of internal value, she stresses cost reductions, such as reduced support costs or reduced de-

Manuscript received 28 December 2001; revised 3 March 2002; accepted 4 March 2002.

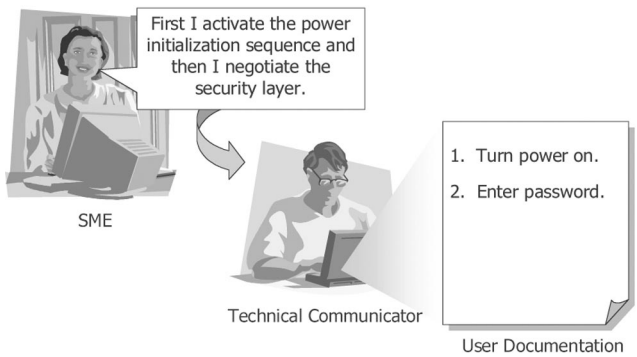

Figure 1. Technical communicator as "packager" of information.

velopment costs by having writers do the documentation instead of developers. In that same issue, Ramey (1995) categorizes the following measures of value for technical communication:

- User satisfaction ratings
- Users' general perception about the value of the information product
- Reduction in support costs
- Prediction of improved performance resulting from better information
- Data about actual performance improvements
- Data about quality of proposals

Carliner (1997) builds on Kirkpatrick's model for evaluating training to emphasize the important role that user performance and client satisfaction play in determining the value that technical communicators add. Mead (1998) speaks extensively about calculating ROI but also addresses softer issues, such as the credibility good documentation establishes in the market and the value technical communicators can have in the development process.

Inherent in all these discussions is the concept that technical communicators add value by increasing return on investment in one of the following ways:

- Directly, through improving the quality of the output and, thus, competitiveness in the marketplace
- Indirectly, through reducing the internal costs of development, production, and support

In making these points, all the works mentioned above have established a solid foundation on which technical communicators can build an attractive value proposition. But Peter Senge in his book on organizational learning, *The fifth discipline*, quotes Arie de Geus, head of planning for Royal Dutch/Shell, "The ability to learn faster than your competitors may be the only sustainable competitive advantage" (1990, p. 4). It is around this added dimension of learning as a competitive advantage that this article tries to expand on what others have established as the technical communicator's value. This article examines the role and value of the technical communicator as creator of knowledge and focuses especially on the internal value of that role.

KNOWLEDGE VERSUS INFORMATION

Although the opening definition from the STC Web site talks about *information*, most technical communicators deal in *knowledge*. Nonaka and Takeuchi (1995) draw a clear distinction between information and knowledge, pointing out that although both center on meaning, knowledge deals with beliefs and is "essentially related to human action" (p. 59). O'Dell and Grayson (1998) state it even more directly: "Knowledge is information in action" (p. 5).

Technical communicators who focus on user-centered writing have already crossed the line from information to knowledge. By reinterpreting technical information in user contexts, they are creating new knowledge by presenting that information in actionable terms and by relating it to specific applications.

Knowledge creation at the user end (which we could call delivery-side knowledge value) is a well-known contribution that technical communicators make. The abundance of literature on user-centered writing attests to that fact. What often goes unnoticed—and therefore unvalued—is the role that technical communicators play in creating knowledge at the front end (which we could call development knowledge value). The rest of this article examines this aspect of knowledge creation—that is, how technical communicators make their companies smarter.

POSITIVISM AND CONSTRUCTIVISM

Technical communicators can be categorized in broad strokes into two worldviews: *positivism* and *constructivism* (see Table 1). For many centuries, formal education (arguably the genesis of technical communication) had been dominated by positivism. Knowledge was seen as something that existed in the world and which could be neatly packaged and transferred from teacher to learner through instruction—illustrated, for example, by the fabled *funnel of Nurnberg* through which knowledge could be easily poured into the head of the learner (Carroll 1990).

Educational reformers, such as Dewey in his paradigm-shifting *Democracy and education* (1997; originally published in 1916), and educational psychologists such as Bruner (1986), offered a different vision, one in which knowledge exists *within the knower* and where learning is an *active social act* (versus a receptive individual state) on the part of the learner. This belief has come to be known as *constructivism*, which Duffy and Cunningham (1996) define as the belief that

> *(1) learning is an active process of constructing rather than acquiring knowledge, and (2) instruction is a*

TABLE 1: POSITIVISM AND CONSTRUCTIVISM CONTRASTED

	Positivism	Constructivism
How they view reality	Singular and Rigid: An "apprehendable reality is assumed to exist, driven by immutable natural laws and mechanisms. Knowledge of the 'way things are' is conventionally summarized in the form of time- and context-free generalizations." (Guba and Lincoln 1994, p. 109) *A product is defined by a finite set of features and functions. An accurate and complete cataloging and description of the features and functions will render an accurate and complete description of the product.*	Pluralistic: "Reality is expressible in a variety of symbol and language systems." Plastic: "Reality is stretched and shaped to fit purposeful acts of intentional human agents." (Schwandt 1994, p. 125) *A product is defined by how people interact with it. No description can ever be complete or totally accurate since the permutations of possible user contexts are too complex.*
How they view the relationship between the knower and the thing being known	Dualistic: There is a distinct separation between the knower and the known. Objective: Facts are true or false. *The technical communicator must be an unbiased describer of a product's functionality.*	Transactional: "Meanings are created, negotiated, sustained, and modified within a specific context of human action. The means or process by which the inquirer arrives at this kind of interpretation of human action (as well as the ends or aim of the process) is called Verstehen (understanding)." (Schwandt 1994, p. 120) Subjective: Facts are deemed viable or not viable within a community of practice. *The technical communicator must interpret product functionality in light of both the user contexts and the developers' intentions.*

> *process of supporting that construction rather than communicating knowledge. (p. 171)*

They quote von Glaserfield who says,

> *Instead of presupposing knowledge is a representation of what exists, knowledge is a mapping, in the light of human experience, of what is feasible. (p. 172)*

Positivism remained the underlying worldview of the hard sciences.

> *Until the 1950s logical positivism was the leading philosophy of science; today its influence persists especially in the way of doing philosophy, in the great attention given to the analysis of scientific thought and in the definitely acquired results of the technical researches on formal logic and the theory of probability. (Murzi 2001)*

Meanwhile, constructivism gained popularity in the social sciences (Lincoln and Guba 1985). Because early technical writers tended to come from the sciences (or tried to emulate the scientists they worked with), they were likely to hold positivistic worldviews—an influence still strongly felt. However, as more and more technical communicators entered from the humanities and social science fields, a more constructivist worldview shaped their practice.

For example, common concepts in the contemporary literature of technical communication, such as "user context" and "discourse communities," reflect this constructiv-

ist influence in that they locate reality and meaning within specific social contexts.

If one adheres to the positivistic worldview, then the restricted view of the technical communicator shown in Figure 1 is an accurate description of the technical communicator's role. If, however, one accepts a more constructivist worldview, a broader, more exciting definition starts to emerge: *Technical communicators negotiate meaning within development communities and between those communities and user contexts, and they capture the resulting consensus as knowledge assets.*

KNOWLEDGE DIMENSIONS

In examining the nature of knowledge creation within organizations, Nonaka and Takeuchi (1995) describe the epistemological dimension using Polanyi's (1966) distinction between explicit knowledge and tacit knowledge. They describe the ontological dimension as including individual knowledge, group knowledge, and organization knowledge.

Explicit and tacit knowledge

Explicit knowledge is knowledge that we know that we know. It can be articulated, codified, stored, transferred, and recalled through symbols. In short, it is the knowledge that can be transferred through documents. *Tacit knowledge* is knowledge that we do not know that we know. It is difficult to articulate and generally is expressible only through action. For example, which of the following two sentences is odd?

a) There are three new company cars in the parking lot.

b) There are company new three cars in the parking lot.

Although you immediately knew that answer b was the incorrect sequence of modifiers (because of your tacit knowledge of English) you would probably be hard pressed to articulate (make explicit) the rule you were applying. You would be equally hard-pressed to find a grammar book that articulates the rule. Yet the tacit knowledge of this rule is universal among native speakers of English.

The concepts of tacit and explicit knowledge can be illustrated by Howell's (1982) description of the learning process. In his model, the learner progresses through the following levels:

1. *Unconscious incompetence*—One is not even aware that he or she lacks knowledge or skill (that is, one lacks an awareness that certain knowledge or skills even exist).

2. *Conscious incompetence*—One is aware that he or she lacks a knowledge or skill.

3. *Conscious competence*—One acquires the missing knowledge or skill and applies it in articulated or codified ways.

4. *Unconscious competence*—One's knowledge or skill becomes second nature, applied seemingly without thought or effort.

If you replace the terms *competence* and *incompetence* with *knowledge* and *ignorance*, respectively, and likewise replace *conscious* and *unconscious* with *explicit* and *tacit,* you can derive the model of knowledge-states within an organization shown in Figure 2. The boxes and interface within the gray oval describe the traditional domain of technical communication—that is, transferring what is explicitly known to those who consciously want to know it.

The problem is that the knowledge of greatest value to an organization—expert knowledge—is often tacit (Nonaka and Takeuchi 1995). On the other end of the spectrum, user ignorance is often tacit as well. Horton, for example, states that one problem with novice users of software products and their associated help files is that the novice user does not know enough to phrase useful inquiries of the help system (Horton 1994). In other words, such users don't know what they don't know. So in practice, technical communicators find themselves between experts who cannot articulate what needs to be known (or even that it exists to be known) and users who do not know to even ask. The following personal anecdote illustrates this dilemma.

I was standing at the copy machine one day producing sets of collated and stapled documents, which happened to be versions of our user manuals that had been translated into Spanish. My boss came up and was surprised.

"How is it doing that?" he asked. I assumed he wondered how the copies were coming out in Spanish.

"Oh no," I explained, "The original is in Spanish."

He looked at me with obvious impatience. "The stapling; how did you get it to do *that*?"

It had never occurred to my boss to ask someone *how* to make stapled copies because it had never occurred to him that a copier could do that. His ignorance was tac-

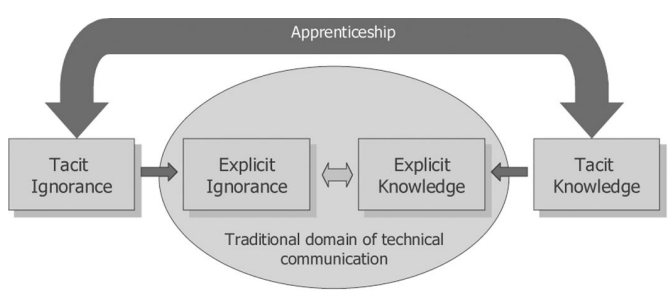

Figure 2. A knowledge states model.

it—he didn't know that he didn't know—until he saw it being done. On the other hand, my knowledge of how the copier worked was so tacit that I could not imagine someone not knowing that it could staple copies. Therefore, I made the embarrassing assumption that my boss thought the copier was somehow translating our documentation into Spanish.

Nonaka and Takeuchi (1995) point out that tacit knowledge becomes explicit through *action*. The frustrated music teacher who grabs the guitar from the student's hand and says, "Here, let me show you what I mean" is trying to make her tacit knowledge of a technique explicit through the action of demonstrating the technique. Much practical teaching in the past bypassed the requirement to make tacit knowledge and tacit ignorance verbally explicit by relying on apprenticeships, as shown in Figure 2.

In apprenticeships, teaching and learning are accomplished through direct action. Experts do what they do, operating in their state of unconscious competence; novices watch, help, and sometimes ask specific questions in direct response to something they see the master practitioner do. Much practical learning today still relies on apprenticeship (Brown, Collins, and Duguid 1989).

Two requisite elements of apprenticeship that can limit its efficiency within complex organizations, however, are *proximity* and *synchronicity*—that is, the teacher and the learner must work close to each other and at the same time. Global organizations, virtual offices, skills specialization, and departmentalization can make either of these two requirements difficult if not impossible to meet.

Lacking an apprenticeship capability, bringing tacit knowledge to the aid of tacit ignorance often first requires making both the expert's knowledge and the user's ignorance explicit. On the user side, active inquiry methods such as usability testing and contextual task analysis help make users' tacit ignorance explicit.

For example, a user who does not know that a Web page needs to be refreshed to see the latest results of an update is not going to just say, "I don't know that Web pages need to be refreshed." That tacit ignorance only becomes explicit by participating in a usability test or by observing workers on the job. Once again, the need and concern to create delivery-side knowledge has been well documented. Less focus has been placed on the need and the methods to help developers and SMEs make their tacit knowledge explicit so that it can be captured, stored, and transferred.

Scales of knowledge

Nonaka and Takeuchi (1965) categorize knowledge along an ontological dimension that includes individual, group, and organization knowledge. Figure 3 shows how knowledge, when categorized this way, can be measured along two scales: the scope of the knowledge—how much is known—and the durability of the knowledge—how long it stays known. Its pyramid shape represents a hypothesized distribution of knowledge within an organization.

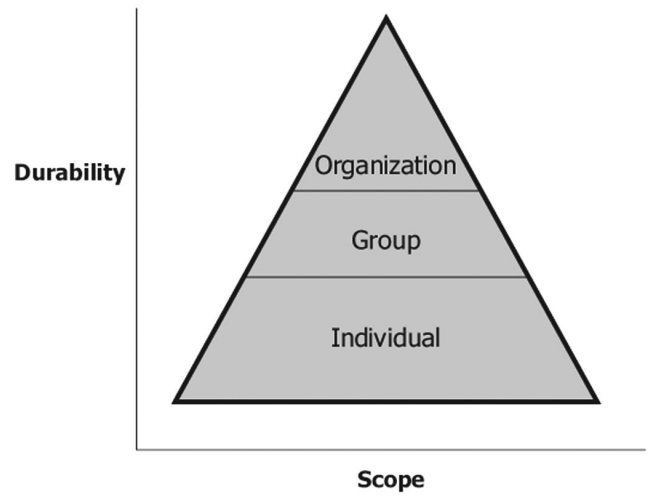

Figure 3. Scales of knowledge.

- Most of the knowledge within an organization exists within individuals (hence the wide scope). Individual knowledge is also the least durable of the three types of knowledge—for example, if an individual leaves the organization, that individual's knowledge leaves as well.
- Group knowledge is knowledge constructed by and shared within groups, such as departments. If one member leaves, the group knowledge endures because knowledge is distributed and replicated within the group-as-organism—each member contains the group's knowledge in much the same way that each cell in an organism contains the total DNA for that organism. For this reason, group knowledge is shown as being more durable than individual knowledge.
- Organization knowledge is knowledge that resides at the enterprise level. It is the knowledge that is embedded in the processes, practices, tools, and repositories within the enterprise. As such, it is the most durable, being the least dependent on the persistence of persons and personalities. O'Dell and Grayson (1998) credit organizational expert Leif Edvinsson of Skandia in calling this *structural knowledge* because it often resides in tools and physical artifacts, such as knowledge management systems and documentation.

To illustrate the three levels of knowledge, let's consider a fledgling technical writing department in a company. Each writer brings a large amount of individual

Technical communicators help design teams arrive at consensus about what the product is or does.

knowledge to the job, including personal vocabulary, knowledge of grammar rules, word processing skills, and the ability to compose informative prose. Each writer also has his or her own style for writing and producing the manuals for which he or she is responsible.

If any individual writer leaves, his or her project stops, and he or she must be replaced by another individual with the equivalent knowledge gained from previous individual learning. For this reason, job ads for technical writers typically include requirements for an English or communication degree and a certain amount of experience with specific authoring tools. The new writer coming on board would still have to study and decompose the style and layout rules the previous writer had been employing or else redo the project to match his or her own style decisions.

As the department matures, the writers realize they need to present a consistent look and feel within their documents. They begin to identify areas that need to be consistent, such as fonts, page layouts, and use and spelling of certain industry-specific terms. They debate their options within each of these areas and collectively decide what the department standards will be (during formal meetings and *ad hoc* discussions held in response to specific situations). These decisions represent *group knowledge*.

If one writer gets sick, another writer can step in and take over that writer's work and can feel familiar at least with issues of style and presentation. But if a serious cutback occurs and the entire department is laid off, that knowledge is lost. A new department formed later would have to start over.

But let's say that those style and presentation decisions had also been incorporated into a template in which font attributes, margins, headers, footers, and so forth were preset within the word processing software. Furthermore, let's say that company-specific terminology had been added to the spell checker's list of words within the word processor. Then the knowledge would be contained within the *organization's* memory in the form of an artifact (the template).

A new writer starting a new department would merely have to use the template, and then all the knowledge created by the previous team would be applied, *without the writer, as an individual, having to relearn or even know what that knowledge was*. For example, a novice writer might not know that 10-point Times New Roman had been chosen for the body text because of its well-researched legibility, or that Arial 14-point bold had been selected for level 1 headings to provide a contrast with the body text and subordinate headings. And perhaps the writer would not know that *token-ring* was to be hyphenated in this particular company's documentation.

Even though the writer had no individual knowledge of these prior decisions, he or she would still do the correct things because that knowledge had been embedded into one of the organization's tools—the word processing platform—and as such, had been transformed into organization knowledge. Organization knowledge, applied through an artifact, has the same effect in this case as tacit expert knowledge—it enables *acts of unconscious competence*.

Any time knowledge is escalated from individual to group or from group to organization, it gains value as a *knowledge asset* (knowledge that gives a competitive advantage to the enterprise) for three reasons:

- Escalation increases the durability of the knowledge.
- Escalation forces a degree of public scrutiny that validates or challenges the accuracy and worth of the knowledge. Individual knowledge, for example, must gain professional peer acceptance (consensus within its community of practice) to become group knowledge.
- Escalation makes transfer easier. For example, once the technical communication department in the example had created a template, that template could be shared with other departments to improve the quality with which their documentation is produced.

In essence, this discussion lays the groundwork for an expanded value proposition for technical communicators: *They escalate knowledge within an organization, thereby increasing the value of that organization's knowledge assets.*

TECHNICAL COMMUNICATORS AS CREATORS OF KNOWLEDGE

Technical communicators make three important contributions as creators of knowledge within an organization.

- They help experts make their tacit knowledge explicit.
- They help design teams arrive at consensus about what the product is or does. In this regard, they are facilitating knowledge creation at the group level (or at a minimum, escalating individual knowledge to the group knowledge level).
- They create knowledge assets. That is, they transform tacit knowledge into explicit artifacts so that it can be accessed by others within the organization.

Making experts' tacit knowledge explicit

Much of what a technical communicator does when working with SMEs can be described as decomposing the final phase of Howell's learning model, unconscious competence, back into its previous level, conscious competence. Restated in terms of the knowledge model of Figure 2, this means transposing tacit expert knowledge into explicit knowledge. Only then can that knowledge be codified into artifacts such as documents and made available to others. Technical communicators can accomplish this in two ways:

- Probing interviews
- Critical reverse engineering

Probing interviews Sometimes dealing with an SME can feel like interviewing a hostile witness. It may seem as if the objective of the interviewee is to withhold information. However, within the knowledge model this article has been discussing, the problem can be described not as reluctance on the part of the SME to share information, but rather as difficulty in making explicit what has become tacit. The image of the technical communicator shifts from one of a journalist beating against a stonewalling adversary to that of a knowledge specialist chipping away at an unintentional mental block—somewhat akin to a psychologist helping a patient unlock a hidden memory. The tone then shifts from an adversarial one to creative engagement.

The practical implication of this perspective is that some technical communicators may need to change their SME interviewing techniques. The assumption going into most SME interviews is that the SME knows what he or she is talking about. In fact, many interviews uncover only the SME's explicit knowledge. Tacit knowledge (which is often the most valuable) goes undiscovered. The following guidelines apply.

- **Interview within an action context.** Don't just talk about a product or service with the SME; work with the process or service with the SME. SMEs automatically perform some steps they would never think to tell you about.
- **Probe body language cues.** For example, if the SME is breezing along in a demo and then suddenly stops and looks puzzled for a moment, ask why. Probe for explanations with questions such as, "Something seemed to surprise you; what did you think was going to happen?"
- **Interrogate for unspoken reasoning.** SMEs often say things such as, "It doesn't matter what number you put in here initially" and then they type a number into a field. Ask why that number and not a bigger one or a smaller one. One useful technique is to offer extreme alternatives. For example, say, "Great; let's put 1 million in." Sometimes it is easier for experts to access their tacit knowledge in response to such an extreme suggestion.
- **Be patient.** The SMEs are not stonewalling. It is just that their knowledge has grown so tacit that they are unable to articulate it. I recently worked with an SME who was demonstrating a sales order entry screen. She entered the order and then said, "Oh good; the items didn't backorder." I asked how she knew that, and she went into an explanation of backorder rules. I asked again, "But you looked at the screen and said, 'Oh good.' What did you see on the screen?" After several exchanges (during which we both got frustrated), she finally said, "The Supply/Demand screen didn't pop up—that's how I knew the items had not backordered." It was so natural for her to "see" that something had not happened that she did not grasp that someone else would not. But in the exchange, I let myself *feel* stonewalled, and in return, I am sure I made the SME feel browbeaten.

Critical reverse engineering Another way that technical communicators make tacit knowledge explicit is by using the product to understand how it works—a technique called reverse engineering. Reverse engineering a software application, for example, entails interacting with the various screen elements and noting what actions result. *Critical reverse engineering* means asking why such an interaction would have value to a user or how such a feature would add value to the user's job or task. The questions that result from this activity often unveil subtle advantages or features that were never explicitly stated in design documents. Sometimes these questions challenge features or functionality, forcing designers to think explicitly, and thus they make their design rationale more accessible to public scrutiny. In these cases, the value added by the technical communicator can escalate into the next level, team consensus formation.

Critical reverse engineering does not replace usability testing or other active ways of getting user data. It is meant to help designers articulate what they have put into the product and why. This technique is especially useful when a feature's benefit or application is not intuitive. What could seem initially to be bad design could be a benefit to the user if explained.

For example, a step might seem redundant, such as pressing an "Update" button immediately after pressing an "Approve" button on a variance report. If challenged, the developer might point out that this process gives the user the option to delay the actual update until after the core working hours when it will have less impact on the processing time of other transactions. That would be useful information to pass on to the user—who might not think of it. That is also the kind of information that might not appear in a specification. It could have been an add-on by the

Figure 4. The emergent nature of design.

developer who anticipated the negative impact that updating large reports could have on transaction speeds.

Consensus forming within the design team

A case study that examined the effects of usability testing on team learning found that the development team being studied did not fully understand how their own product worked (Hughes 2000). This finding is consistent with my industry experience working with and within development groups. One explanation is that organizations often do not end up building what they set out to invent (see Figure 4).

Part of this mismatch between what is planned and what is delivered results because the designers experience an emergent understanding of the tools and technology they are designing with, as well as an emergent understanding of the application they are designing for. In short, they get smarter as they go along and end up doing things they did not know they would be able to do or would need to do. Other reasons for the mismatch are the unanticipated constraints placed on designers by those same tools and technology, as well as by resource limitations.

In these cases, the designers end up not doing things they had hoped to do. But because of the incremental and distributed nature of complex design (different individuals participating in the invention process at different times and phases of the project), these emergent understandings and constraints *are not experienced by the design group as a whole*, but rather by different combinations of individuals along the path from specification to production. In the study referred to above, one function provided by the usability test was to create a unified product vision at the group level. The usability test was a learning catalyst for several important reasons.

- It made tacit ignorance explicit.
- It made tacit knowledge explicit.
- It allowed the team to see the product interpreted within a user context.

In making tacit ignorance explicit, some members were able to see functions they did not understand and ask, "Why does it do that?" Members who understood those functions were able to explain them in the context of what had just occurred in the test, thus making their tacit knowledge explicit. And most importantly, when the design team saw how users interpreted the product in the act of applying it within a user context, their own definition of the product changed. The design group emerged from the usability test with a new, shared vision of what the product was, both in terms of functionality and in the context of how users would apply and value it. New knowledge had been created at the design group level.

Technical communicators act as similar catalysts for creating group knowledge at the design team level, sometimes by organizing and moderating usability tests, but also through simpler, more conventional means such as the documentation review process. Oftentimes, the documentation that technical communicators distribute for review is the first documented description of the product's functionality within an application context. If the correct perspectives are included in the review audience, the resultant

review process becomes a forum for meaningful public investigation and scrutiny.

The technical communicator who understands the value of this process approaches the document review process differently than one who thinks it is just to see "Is this document accurate and complete?" Note, by the way, that this question is framed within a positivistic worldview. The knowledge constructivist is more apt to see the process as initiating a dialogue within a community of practice, one that seeks consensus on "Is this a viable representation of what the product does?" The differences between the two approaches are mostly a function of

- Who gets included in the review process (positivists seek experts; constructivists seek both experts and stakeholders)
- How differences are resolved (positivists seek objective methods to determine who's right; constructivists seek dialogue to form consensus)

Creating organization knowledge assets

So far the values discussed have had to do with knowledge creation, but what about the durability of that knowledge? The most durable level of knowledge is organization knowledge, explicit knowledge that has been articulated and codified into artifacts that can be distributed to or accessed by others within the organization.

Technical communicators who see themselves as creators of knowledge think beyond the concept of documentation and think in terms of knowledge management systems. Writing becomes a secondary and subordinate activity to content management. The user community becomes just one of many stakeholders who can benefit from the knowledge the technical communicator has created, and the technical communicator seeks ways to distribute that knowledge to all the stakeholders—for example, marketing, operations, engineering, and so forth.

Wick (2000) describes four layers of knowledge management:

- Document-centered
- Technological
- Socio-organizational
- Knowledge organization

The first layer is already widely used by technical communicators in the form of document repositories. The technological layer involves more interactive access and open inquiry mechanisms, not unlike many online systems developed by technical communicators today. It also involves a genre of products called knowledge management systems. The more technical communicators do to create systems that make the knowledge within their documentation searchable or otherwise accessible within the enterprise, the more value they add to the enterprise. The last two layers speak more to what this article has tried to address: the technical communicator as a facilitator of knowledge creation and dissemination.

The most durable level of knowledge is organization knowledge . . . .

A NEW VALUE PROPOSITION

If one accepts de Geus's assertion that the ability to learn (that is, create knowledge) is the most sustainable competitive advantage, then technical communicators can add to their current value proposition by showing that their value goes beyond being cost-effective providers of information to end users. An additional role, that of an agent of organizational learning, carries additional value to the organization's competitive position.

This aspect of technical communicators' contribution can be even more valuable than the documentation they provide end users. By expanding the technical communicator's value proposition to include the creation of knowledge assets that improve the competitiveness of the enterprise, his or her value persists regardless of whether end users ever even look at the documentation. Given what we know about end users and their reluctance to go to help files or manuals, this expanded view of the technical communicator's value is welcome indeed.

PRACTICAL IMPLICATIONS

Technical communicators can improve an organization's wealth by increasing the value of its knowledge assets. This article has discussed a progression of activities that create knowledge and escalate it to higher asset levels. First, technical communicators help experts make tacit knowledge explicit. They do so through probing interviews that help SMEs discover and articulate that tacit knowledge. They also do so through critical reverse engineering of products.

Once tacit knowledge has been made explicit at the individual level (SME), technical communicators help promote it to team knowledge through techniques such as review processes, usability testing, and other collaborative techniques. These activities bring individual knowledge into public forums of scrutiny and help negotiate meaning within and among communities of practice and stakeholders.

Technical communicators then promote knowledge to organizational levels through artifacts such as templates and through knowledge management systems.

One implication of this discussion is the question of when technical communicators need to be involved in a project. If one holds an information-centered view, technical communicators should become involved at the *information/production sweet-spot*—as late as possible, so

that the information is stable but there is still enough lead-time to produce the documentation. In this approach, rewrites due to product changes are seen as inefficient.

However, a knowledge-centered approach argues that technical communicators need to be brought in as soon as possible so they can facilitate the complex process of knowledge creation and promotion. Rewrites are seen as the natural emergence of knowledge and, therefore, a source of value rather than inefficiency.

Another practical consideration is the skill sets and training required to be a creator of knowledge rather than an information packager. Whereas the information packager uses authoring tools and publishing systems, a creator of knowledge must become familiar with content management tools and knowledge management systems as well. Creators of knowledge must also be skilled facilitators and shrewd interviewers (similar to anthropologists and sociologists). Degree programs in technical communication should include courses to develop these skills.

Most importantly, technical communicators who wish to be creators of knowledge must envision their value in new ways. Things that an information packager sees as inconveniences, such as SMEs who cannot agree or products that change during development, are seen as opportunities to add value by creators of knowledge. Disagreements become opportunities to forge a consensus around a new vision of the product and not a need to bring in a management referee to decide who should "win" the argument.

Design changes need to be communicated to a broader community of stakeholders, another opportunity for adding value. Closed-mouthed SMEs cease to be seen as adversaries and are instead seen as tacit experts who need help in making their knowledge explicit. In short, all the things that make life tough for information packagers make value-add opportunities for creators of knowledge.

CONCLUSION

This article advocates that a technical communicator who sees himself or herself as an information packager should broaden that perspective to that of a creator of knowledge. This redefinition of self may require a reflection and reevaluation of one's worldview within the contexts of positivism and constructivism. In this regard, this article invites those technical communicators to accept the definition of technical communicator offered earlier:

> *Technical communicators negotiate meaning within development communities and between those communities and user contexts, and they capture the resulting consensus as knowledge assets.*

But many technical communicators are already there, although some at a tacit level. This article encourages those technical communicators to articulate their value in more explicit terms and to formulate strategies that reflect the value proposition of a creator of knowledge. It encourages them to redefine what have been traditionally viewed as inconveniences to their job as the natural outcomes of transitions between knowledge levels, and to see themselves not as victims of these circumstances (for example, noncommunicative SMEs, continually shifting product specifications, groups isolated from each other through departmental silos), but as change agents who can facilitate the enterprise's evolution to more competitive levels.

Finally, perhaps STC's Web site should say, "All these professionals help their organizations define and communicate technical knowledge in ways that create value both for those who create products and services and for those who use them." T**C**

REFERENCES

Brown, J. S., A. Collins, and P. Duguid. 1989. "Situated cognition and the culture of learning." *Educational researcher* (January-February):32–42.

Bruner, J. 1986. *Actual minds, possible worlds*. Cambridge, MA: Harvard University Press.

Carliner, S. 1997. "Demonstrating effectiveness and value: A process for evaluating technical communication products and services." *Technical communication* 44, no. 3:252–265.

Carroll, J. M. 1990. *The Nurnberg funnel*. Cambridge, MA: The MIT Press.

Dewey, J. 1997. *Democracy and education*. New York, NY: The Free Press.

Duffy, T. M., and D. J. Cunningham. 1996. "Constructivism: Implications for the design and delivery of instruction." In *Handbook of research for educational communications and technology*, ed. D. H. Jonassen. New York, NY: Simon & Schuster Macmillan, pp. 170–198.

Guba, E. G., and Y. S. Lincoln. 1994. "Competing paradigms in qualitative research." In *Handbook of qualitative research*, ed. N. K. Denzin and Y. S. Lincoln. Thousand Oaks, CA: Sage Publications.

Henry, J. 1998. "The value added by technical communicators in collaborative writing situations." *Technical communication* 45, no. 3:207–220.

Horton, W. 1994. *Designing and writing online documentation*. 2nd ed. New York, NY: John Wiley and Sons.

Howell, W. C. 1982. *Information processing and decision making (human performance and productivity)*. Mahwah, NJ: Lawrence Erlbaum Associates.

Hughes, M. 2000. *Team usability testing of Web-based training: The interplay of team learning and personal expertise*. Doctoral dissertation, University of Georgia.

Lincoln, Y., and E. Guba. 1985. *Naturalistic inquiry*. Newbury Park, CA: Sage Publications.

Mead, J. 1998. "Measuring the value added by technical documentation: A review of research and practice." *Technical communication* 45, no. 3:353–379.

Nonaka, I., and H. Takeuchi. 1995. *The knowledge-creating company*. New York, NY: Oxford University Press.

Murzi, M. 2001. "Logical positivism." *The Internet encyclopedia of philosophy*. http://www.utm.edu/research/iep/l/logpos.htm

O'Dell, C., and C. J. Grayson, Jr. 1998. *If only we knew what we know: The transfer of internal knowledge and best practice*. New York, NY: Free Press.

Polanyi, M. 1966. *The tacit dimension*. London, UK: Routledge & Kegan Paul.

Ramey, J. 1995. "What technical communicators think about measuring value added: A report on a questionnaire." *Technical communication* 42, no. 1:40–51.

Redish, J. 1995. "Adding value as a technical communicator." *Technical communication* 42, no. 1:26–39.

Schwandt, T. A. 1994. "Constructivist, interpretivist approaches to human inquiry." In *Handbook of qualitative research*, ed. N. K. Denzin and Y. S. Lincoln. Thousand Oaks, CA: Sage Publications.

Senge, P. 1990. *The fifth discipline: The art and practice of the learning organization*. New York, NY: Doubleday.

Society for Technical Communication. 2001. "For the press: Frequently asked questions (FAQs)." http://www.stc.org/pr_faqs.html#1. Viewed on 23 December 2001.

Wick, C. 2000. "Knowledge management and leadership opportunities for technical communicators." *Technical communication* 47, no. 4:515–529.

MICHAEL HUGHES is president and founder of Working Knowledge, Inc., a consulting company that helps clients develop training, documentation, and easy-to-use interfaces for technology-based products. He has managed technical communication departments for technology companies and taught in the master's program in technical and professional communication at Southern Polytechnic State University. He has a PhD in instructional technology and an MS in technical and professional communication. Contact information: mikehughes@mindspring.com