Version 1.2
Last updated: 3/24/2012
Prepared for DCMI by
David Talley, iSchool, Univ. of Washington

A Process Documentation Model for DCMI

Case Study for Instruction Design of Procedural Documentation

The Dublin Core Metadata Initiative (DCMI) encounters frequent needs to develop instructional materials associated with its mission to provide essential metadata vocabularies that facilitate interoperable systems for discovery and management of information resources. Instruction becomes a priority in central DCMI efforts around linked data to enhance semantic interoperability and more generally around support for exchanges between separately maintained metadata resources built on mutually incompatible frameworks. DCMI organizes meetings among these and other constituencies, resulting in a second tier of instructional priorities for meeting management and applications of supporting technologies.

Recognizing this priority, DCMI undertook an effort to create a consistent framework for documentation to streamline creation of internal instructional resources of the second type discussed above. Much of its documentation describes applications of its metadata schema, especially for interaction between systems. Those materials require their own structure and content priorities and are not addressed by the framework presented here. The documentation model developed here applies specifically to internal DCMI processes and activities undertaken in support of its mission.

This effort to build a template for internal process documentation focused on DCMI's preferred meeting management tool, Open Conference System (OCS), as an exemplar. The result is a comprehensive and flexible structure for documentation of tools, procedures, and activities to be adapted for instructional needs in a range of areas. While the template is intended specifically to guide development of documentation for procedures involving OCS, it also should have general applicability to task and process documentation across DCMI's internal operations and activities.

Such a general-purpose documentation model must rest on a solid foundation of theory and experience accumulated by researchers investigating priorities for effective practice. That need motivated a literature review to inform development of a theoretically justified and practically useful template.

Literature on Documentation Effectiveness

Considerable work was completed in the middle and late 1990s, with relatively settled results summarized and selectively extended by researchers working in the following decade. Eiriksdottir and Catrambone (2011) provide a rather comprehensive summary of this literature, and the discussion here will emphasize applicability of established results within the DCMI operating environment.

At its most basic level, "procedural discourse is largely about telling someone who is in one set of circumstances how to transition to another set. In other words, . . . procedural discourse describes system states and actions that change system states" (Farkas 1999, p. 42). The task of developing documentation breaks down to an effort to describe three system states and transitions between them (Farkas 1999, Barber 2003). The prerequisite state defines a starting point that establishes essential supporting conditions, including collection of data inputs, that allow a process to begin. The interim state obtains when process activities are underway toward achievement of a desired state, which represents the preferred process outcome. A fourth unwanted state describes variation from the three-step sequence resulting in corrective action to restore progress toward the desired state (Barber 2003).

Documentation can engage this sequence at varying levels of detail, depending on the learner's situation and instructional needs. A three-tier high-level model has emerged (Barber 2003, Eiriksdottir & Catrambone 2011), beginning with tutorial documentation suited to pure beginners whose lack of experience with a process creates a need for comprehensive information. In particular, tutorial documentation provides extensive background detail about process context and objectives. Procedural documentation meets the needs of users already familiar with that context and focused on accomplishing particular tasks. Reference documentation supports quick access by relatively experienced users to specific facts essential for completion of otherwise well-understood tasks.

This usage from the instructional literature may create a possibility for confusion with applications of the term *reference* in library and information settings. Much of the instructional material produced by DCMI is properly reference documentation under the library meaning of the term – content accessed by discrete look-up to obtain specific information, as opposed to sequential reading of a complete resource for full understanding. Use of the term *reference* within this discussion of the DCMI process documentation model addresses the meaning specific to instructional documentation –information intended for experienced users needing quick access to particular facts that allow them to complete the documented process or procedure.

Considerable discussion has addressed the inclusion in task instruction of contextual information about the process and environment of which the task is part. A minimalist model of documentation describes system functions only as they directly help to complete user tasks (Carroll 1990, Manning 1998). This task focus derives from analysis of user needs with the intention of transforming a mere inventory of all possible system actions into useful instruction that puts system resources to work for practical purposes (Barber 2003). Delanghe asserts, however, that "although dressed up as 'user tasks,' most manuals still simply describe features" (2000, p. 201). Farkas (1999) observes that some high-context instructional situations are best served by adoption of "rich-step" models, which embed background information into direct task instruction. Even paragraph-formatted discussion, he says, can sometimes help to "emplot" the learner (Goodwin 1991), or embed the learner in an unfolding story involving system functions, which promotes personal identification with a complex context.

Indeed, instructional goals are formulated within a situation broader than just the learning itself. According to Lentz and Pander Maat (2004, p. 395), "instructive texts have as their main purpose to explain to people how to use a product, but they often also attain secondary persuasive purposes . . . for example, acceptance of a new technology." Those authors identify three essential sections of an instructional document, Background, Problem, and Solution, then they "introduce a fourth: the Motivation. This is a section that presents the motivation (or justification) for solving this problem" (p. 395). DaSilva and Henderson (2007) agree that the purposes of an instructional document strictly defined do not include such "relational purposes" but those ancillary goals "clearly imply additional constraints on the design space" (p. 213).

Research on the effectiveness of instructional documentation tends to narrow that broad context and focus instead on how learners develop strong mental models to support task completion. At a minimum, learners approach task-focused learning already in possession of tacit knowledge gained within their environment (Knapheide 2000, Barber 2003). The task environment also includes colleagues who "know the relevant subset of the relevant missing knowledge of a confused user" (Knapheide 2000, p. 423), leading learners to consult experienced users before resorting to documentation. Dabbagh and Denisar (2005) describe a more individual process of mental modeling. In particular, they question assumptions that structured discourse must imply a rigidly hierarchical outline. Instead, they make a case that heterarchical networks of information nodes support learning through "relational or network-like hypermedia representations of ill-structured problems or cases" (p. 20). In particular, "the heterarchical case structure facilitated more exploratory-type tasks, which put students in control of problem solving, encouraging them to try out different strategies and hypotheses and observe their effects" (p. 20).

Learners often favor such exploration strategies, according to Eiriksdottir and Catrambone (2011), and "providing explicit invitations to explore helps to keep them motivated and relates the instructions to the tasks that they want to complete" (p. 760). While exploration induces mental modeling by learners, as they set their own goals or adopt the goals embedded in the system as their own, they still need to acquire domain knowledge to guide practice in new skills. Directed problem-solving helps to guide that modeling more effectively than does more general exploration (Eiriksdottir & Catrambone 2011).

Roy (2007) reports some benefits from early introduction of a procedure's focus through an "object-sentence first" structure that may "help readers to mentally animate the final goal, even before reading through the step-by-step process" (p. 161). Steehouder (2004, p. 2) observes that "steps were read faster when they were preceded by outcome information." Users studied were able to increase both the amount of text read within a given time and understanding of the material when step outcome was stated first, followed by steps for achieving that outcome. Karreman and Steehouder (2004) report no improvement in initial task success when instruction text contains contextual information, but retesting learners after a delay reveals some improvement in longer-term error recognition and understanding of system logic.

So a need for mental modeling may provide some rationale for including explanation of context and principles in task-focused instructions (Eiriksdottir & Catrambone 2011), still practical task completion defines the milestones and nodes within the mental model. Manning (1998) describes a process where learners initially work from schemas, or mental models built on past experience, to structure new perceptions and create hypotheses, which they immediately test by acting on their provisional understanding. In this process, Manning states, they ignore contradictory indications or evidence not directly and immediately relevant to accomplishing their intended tasks. Accomplishing goals in this way reinforces the original mental schemas (Manning 1998). Similarly, Carroll (1990) acknowledges that learners function in context but notes also that they come to understand their situations by observing the outcomes of their actions. A user is ready to learn something new when a current understanding fails to accomplish goals (Manning 1998), that is, when trial results in error.

Many researchers agree with Knapheide (2000) that users tend to seek help from documentation only when they encounter problems, and they typically read only enough to resume progress toward task completion. Even where they seek help from colleagues, it is because those advisors' awareness of task context and history allows them to suggest quick fixes (Knapheide 2000). Karreman and Steehouder (2004) observe that "system information," or contextual explanation of a procedure, does not improve initial task success, and it reduces errors "only if the information describes the internal functioning of the product, not when it merely stresses motivational aspects or general principles" (p. 34). In fact, these authors suggest that such system information could actually impede task learning by increasing cognitive load on learners while also increasing their perceptions of system complexity, which may reduce their feelings of self-efficacy.

Roy (2007) describes contextual information within task instruction as a potential source of learner frustration rather than assistance, because "instructional designers often create instructions involving the whole assembly context (global context), while readers often may have a smaller or limited context (local context) in mind" (p. 149). This sort of disconnect is sometimes evident in style of presentation, as well, where learners mostly favor a "focused" style emphasizing practical coaching and concrete examples, while technical writers tend toward a "dynamic" (exploratory) or exhaustively "rigorous" style (Delanghe 2000).

To establish the right level of contextual information in relation to procedural detail, Farkas (1999, p. 45) describes a minimalist model of system documentation centered on a decision-action sequence that "enables users to quickly decide (while reading the initial components) whether the particular procedure meets their needs, and if the procedure is appropriate, to move crisply from decision-making to carrying out actions." Barber (2003) warns of increasing risk from omission of detail in situations where failure due to inexperience can lose or corrupt data. To control this risk, task guidance can anticipate user error and provide topical warnings along with action-relevant recovery steps (Barber 2003, p. 80). Delanghe (2000) also suggests giving significant amounts of detail only with error recovery information and otherwise summarizing enough to support task completion. Eiriksdottir and Catrambone (2011, p. 753) conclude that "users who do read procedural instructions often scan large parts of the instructions and read carefully only when they need clarification."

Detail provided in system documentation must balance support for learners with "letting go" that frees them to engage the system and discover via self-directed learning both what actions to take and how they fit together (van der Meij & Gallivej 2004, pp. 7-8). Invitation to explore within an "On Your Own" section of an instructional document can cue the right sort of exploration to discover important context for current conceptual or procedural elements. Oberman (1983) emphasizes the value of promoting such self-regulation, even where gaps in minimalist documentation may leave learners momentarily confused. "Students who cannot successfully self-regulate encounter frustration which often results in failure. 'Constructive frustration' or better yet 'controlled frustration,' however, may stimulate self-regulation and, therefore, help students sharpen their reasoning abilities" (Oberman 1983, p. 25).

Elements of a Documentation Model

If DCMI is to adopt a rather minimalist documentation model, excluding most background and context discussion, then what elements should that model *include*? Prerequisites to start a procedure are stressed by van der Meij and Gallivej (2004, p 7), potentially including specification of system states, user skills, and user knowledge. Explicit statement of a starting screen may be especially important. Steehouder (2004) confirms the selection of elements from Farkas (1999, pp. 46-48), which begins with a "nearly mandatory" title that helps the learner judge the relevance of the procedure to a task at hand. An optional conceptual element may help to confirm that judgment. A list of procedure steps, the only universally mandatory element according to Farkas, explicitly describes system responses to user actions, sometimes with a "facilitating modifier" such as the location of a required control in the user interface. Subheads may cluster related steps within a complex procedure, stated with an infinitive verb indicating the desired outcome (e.g., "To set up the first part, …"). Optional notes may be interspersed within steps to surface warnings and links to related procedures.

Farkas (1999, p. 45) gives more detailed guidance for development of the essential core of procedure in a "streamlined step model." Each step must incorporate:

1. A short action statement (or two) per step
2. Simple formatting
3. Phrasing based on an imperative verb
4. Little or no explanatory information (Some explanation may follow in notes.)
5. "Layering" via hyperlinks within online documentation to give access to contextual information at the learner's discretion

Other authors confirm the emphasis by Farkas on layering to control levels of detail confronting learners. This structure includes only basic instruction in a procedure step, but learners may click or hover on-page controls, where appropriate, to display additional detail at their option (Barber 2003, pp. 67-68, 444-445; Delanghe 2000).

In contrast, repetition even of basic information may result when multiple procedures involve similar steps. In such a case, the technique of "fading" can gradually decrease detail in instruction for basic-skills tasks with recurring step sequences. Fading provides "successively less complete reminders . . . [which] change from predominantly procedural instructions toward conceptual information" (van der Meij & Gallivej 2004, p. 7). Eiriksdottir and Catrambone (2011, p. 766) agree that learners respond well where procedures decrease detail and increase abstract information in second and later appearances of similar step sequences, or when they combine familiar steps. These authors note that fading works better when it drops the last step in a procedure than when it omits the first step.

The number of steps within a procedure is important to manage the scope of the learning presented. Multiple authors agree with van der Meij and Gallivej (2004) that an optimal sequence contains three to five steps, or five steps plus or minus two, especially where a user must memorize a sequence to complete a task. Longer procedures can group steps within task-specific subheadings to avoid overly long sequences.

Farkas (1999) advocates use of imperative verbs in step content, creating relatively direct commands that promote clarity and learner engagement. DaSilva and Henderson (2007) discuss a Rhetorical Structure Theory model for overall documentation planning that may also help to guide step content creation. RST defines "segments" of an overall text, composed of "nuclei" which are essential to understanding and "satellites" that provide supporting information. A segment articulates a relationship between segment elements, usually one nucleus and one satellite (e.g., solution isSolutionFor problem). Some relationships can link multiple elements of equal importance (e.g., eventA precedes eventB precedes eventC). DaSilva and Henderson (2007) illustrate relationships as diagrams, providing visual indicators of formal content structure that may help to focus procedure steps on essential outcomes.

In addition to steps within instructional procedures, practical examples can emphasize task-based problem-solving. Judicious use of practical examples may combine benefits of specific steps and more general system information in context of one another to "help people instantiate abstract concepts and provide them with an instance of how a rule governing the task applies to a particular situation" (Eiriksdottir & Catrambone 2011, p. 756). This strategy brings some risk of overly literal interpretation of examples rather than extrapolation from those instances to more general procedural actions.

Timely warnings interspersed with procedure steps provide important protection against error as well as guidance in recovery from errors that occur (Barber 2003). To avoid errors, van der Meij and Gallivej (2004) emphasize a see-think-act sequence in procedure step construction; in contrast, discussion of error recovery and problem-solving should emphasize a detect-diagnose-correct sequence, with emphasis on just-in-time detection of errors, because early intervention limits damage. Graphic illustrations of procedures, especially screen images, are most effectively used for error prevention and recovery. In particular, images help to emphasize goal states to be achieved by correct actions (Barber 2003). Graphics should illustrate interim states only to highlight warnings and prevent or recover from errors, in most cases (Barber 2003, van der Meij & Gellevij 2004).

Roy (2007) cites Tufte (1983) in asserting that illustrations can effectively provide instructional text in procedural context by incorporating labels within graphic process depictions. This combination reduces the need for learners to switch between instructional elements. Indeed, "labels may contain the most important text in the document" (Roy 2007, p. 150).

Farkas (1999) emphasizes use of illustrations, especially flowcharts, as support for learners' construction of mental models. In particular, flowcharts can represent conditional procedures with less linear paths than streamlined steps describe. Steehouder (2004) tested task completion by learners working from text step lists, tables, flowcharts, and logical tree illustrations. Flowcharts and logical tree charts were linked to the highest learner performance, and text step lists to the worst. However, learners sometimes preferred less effective but more familiar formats. Steehouder (2004, p. 2) concluded that "any graphical format is preferred to prose," at least for tasks involving conditional selection among alternatives.

A Process Documentation Model for DCMI

DCMI relies heavily on volunteers engaging within narrow sets of tasks to complete much of its work, and this reality powerfully affects adaptation of the results reported so far to its need for instructional process documentation. Training materials must carefully position instruction within the tutorial/procedural/reference sequence with little expected benefit from learner familiarity. DCMI cannot generally expect learners to take time to work through extensive tutorial documentation, since most engage with its processes in roles ancillary to their central work responsibilities. These users can be expected to focus on near-term task completion within a limited scope, so the procedural level of documentation will have the greatest prominence for DCMI.

This is especially true of engagement with OCS, because the episodic nature of meeting planning tends to expose users to tasks within annual or longer cycles. Procedures repeated so infrequently may not be retained from one meeting cycle to the next, creating a need to refresh memory. Nevertheless, some users important to DCMI do work regularly with its systems, so its documentation may need to support reference lookup to retrieve limited information about specific functions.

DCMI can expect little opportunity for its learners to consult colleagues for task guidance, as described by Knapheide (2000). Its volunteers work mostly independently in a distributed network of individual contributors, so they will need to rely on document artifacts for their learning.

Such task-focused, directed problem-solving is especially important for documentation of OCS functions within DCMI activities, because users will be coming to role-based tasks out of context of their familiar daily activities. Individuals may also lack a mental model of the overall conference process, since much of it proceeds outside the boundaries of any one specific role. Even if context information can aid long-term system understanding, as some researchers report, DCMI's process needs are more episodic, with time intervals between tasks, responsibility for which is spread among multiple individuals. That dynamic may be expected to impede long-term retention, so a minimalist, task-focus model best suits DCMI's needs.

The streamlined step model of Farkas (1999) represents a good overall structure for DCMI documentation, with some amendments. A combination of effective title and concise statement of a procedure's key objective will help both procedural and reference learners to scan for instruction relevant to their task. A clear statement of prerequisites for a procedure, such as data to be gathered and a starting point, will help to orient unfamiliar learners. Concisely phrased steps based on imperative verbs should form the basis for any procedure, with layering controls that give access to additional context information at the learner's discretion. DCMI may consider defining categories for such layering information and tracking user access via web analytics to determine which sorts of context information users choose to access. Analysis of event logs could help to refine this step content and layering strategy.

Step documentation for OCS may be too simple to benefit from the complexity of Rhetorical Structure Theory, as described by DaSilva and Henderson (2007). However, RST diagrams bear a strong visual relationship to linked data triples, an area with significance for other DCMI activities. That resemblance may suggest a useful role for the technique within any DCMI instruction developed around linked data.

Screen captures and similar illustrations within step lists may provide useful context in OCS documentation, and in DCMI process documentation in general, as described by Barber (2003) and van der Meij and Gellevij (2004). A highly configurable environment like OCS presents a significant risk of overuse of screen illustrations, perhaps driven by an attempt to depict all possible settings. Screen captures should be included judiciously, in cases where they can help users to avoid likely confusion or error, similar to guidelines for warning notes. The principle of layering might be applied, as well, presenting a cropped or thumbnail-sized image and allowing a user to click or hover to display a more detailed image. As in 'more detail' text links, this tactic would control potentially overwhelming visual detail at first glance and for relatively experienced users while still making full information available for users who choose further exploration.

User interactions with OCS are strongly procedural, emphasizing a predictable linear event sequence more than a conditional if-then structure. Illustrations highlighting that sequence may support learners' mental modeling by arranging tasks along a timeline. Such a graphic device would have to accommodate variation in user roles over the duration of an event-management process. A timeline could accommodate the matrix of process sequence and user role by grouping tasks by the phase of their occurrence within an event's lifecycle, with color-coding or icons (or both) to highlight specific user roles associated with particular tasks. A tabular matrix could do the same job but at the risk of overwhelming individual learners with information not relevant to their immediate needs for task completion. A tabular display would also be less visually concise than a graphical timeline. Other processes within DCMI likely involve less linear sequences, so graphic modeling in those cases may have to accommodate more complexity, such as parallel sequences to be completed under certain specified conditions.

DCMI would have to resolve questions about how to implement graphical or other access methods to guide users to parts of process documentation relevant to current tasks. Most such materials would be posted for online access, probably involving a content management platform. Most current CMS tools implement navigation as HTML lists, formatted for display by cascading stylesheets (CSS). Flexibility inherent in CSS would allow display either in traditional list format or as a linear model with relationships indicated by background images. Familiar techniques of collapse-expand and flyout menus could help to control the amount of procedural detail confronting learners, or fully expanded lists could display comprehensive detail. Multiple list formats might be combined in a single display, perhaps to present a graphic model of an overall process as a top-level menu with an expanded submenu listing detailed procedures that form part of a particular process step.

Outside OCS, such linear flows may be ill-suited to aid mental modeling of some instructional situations within DCMI. In some cases, navigation within a flexible heterarchical network structure, as described by Dabbagh and Denisar (2005), may help learners to move from a current task to related tasks in a more ad hoc process. Such a heterarchical structure may suit DCMI's involvement in the Learning Linked Data program, for example, and possibly other instructional situations.

Finally, DCMI may need a metadata structure for managing digital assets that form part of its instructional materials. Flow diagrams, screen captures, and similar images may be included in procedure documents, and they will need to be retrievable for reuse, in revisions or repurposed documents, and potentially in documents for related procedures. A relatively simple set of attributes should suffice for this task including:

- DCMI program involved (e.g., OCS, Learning Linked Data, etc.)
- Document in which the resource is included (e.g., OCS event formatting instructions)
- Relevant procedure, possibly listed by numbered procedure title
- User role involved

Ability to cross-reference digital assets by user role could enable retrieval of collections of assets suited to instruction for particular groups of learners for quick reference. For example, user roles identified in OCS instructional resources are:

- Sponsoring organization representative(s)
- Conference manager (decision maker/implementer, accountable to sponsoring organization representatives)
    - Conference information manager (communication responsibilities, accountable to conference manager)
    - Program manager (responsible for managing conf sessions, call for papers, submission reviews, accountable to conference manager)
        - Track manager(s) (responsible for defining tracks, organizing sessions within specific track, accountable to program manager)
        - Reviewers (responsible for commenting on submissions assigned, accountable to program manager)

  - Registration manager (responsible for attendee registration functions, accountable to conference manager)
- Speakers (submit proposals and papers for accepted sessions)
- Attendees (register and attend)

One person may play more than one of these roles – maybe all – but OCS functions will break down into these logical groupings.

References

Barber, T. 2003. Writing software documentation: A task-oriented approach, 2nd ed. New York: Longman.

Carroll, J. M. (1990). The Nurnberg funnel: Designing minimalist instruction for practical computer skill. Cambridge, Mass.: MIT Press.

Dabbagh, N., and K. Denisar. 2005. Assessing team-based instructional design problem solutions of hierarchical versus heterarchical web-based hypermedia cases. Educational Technology Research & Development 53(2): 5–23.

DaSilva, N., and P. Henderson. 2007. Narrative-based writing for coherent technical documents. SIGDOC'07, October 22–24, 2007, El Paso, Texas, USA.

Delanghe, S. 2000. Using learning styles in software documentation. IEEE Transactions on Professional Communication 43(2): 201-205.

Eiriksdottir, E., and R. Catrambone. 2011. Procedural instructions, principles, and examples. Human Factors (53)6: 749-770.

Farkas, D. 1999. The logical and rhetorical construction of procedural discourse. Technical Communication 46(1): 42-54.

Goodwin, David. 1991. “Emplotting the reader.” Journal of Technical Writing and Communication 21, no. 2:99–115.

Karreman, J., and M. Steehouder. 2004. Some effects of system information in instructions for use. IEEE Transactions on Professional Communication(47)1: 34-43.

Knapheide, P. 2000. Synergy and subsidiarity: The systematic determination of software, user, and operating instructions. International Journal of Human-Computer Interaction 12(3-4): 415-430.

Lentz, L., and H. Pander Maat. 2004. Functional analysis for document design. Technical Communication 15(3): 387-398.

Manning, A. 1998. Minimalism beyond computer documentation. IEEE Transactions on Professional Communication 41(3): 200-204.

Oberman, C. 1983. Question analysis and the Learning Cycle. Research Strategies 1(1) (Winter): 22-30.

Roy, D. 2007. Significance of configuration and subassemblies in sequential procedural instructions and role of text-graphical aid: An explorative study. IEEE Transactions on Professional Communication 50(2): 147-162.

Steehouder, M. 2004. Acquiring procedural knowledge of a technology interface. IEEE Transactions on Professional Communication 47(1): 1-4.

Tufte, E. 1983. The visual display of quantitative Information. Cheshire, Conn.: Graphic Press.

van der Meij, H., and M. Gellevij. 2004. The four components of a procedure. IEEE Transactions on Professional Communication (47)1: 5-14.