Converting to Information Mapping: A Case Study

Aviva Garrett, Haggai Mark, and Jan Johnston-Tyler

Cisco Systems, Inc., uses electronic media as the primary delivery means for customer documentation and training. Information Mapping® techniques are being developed as a methodology for creating and linking modules of customer information. After selecting the Information Mapping methodology, we found it necessary to customize it for our needs. To implement Information Mapping methodology, we defined a system architecture consisting of three main subsystems: a document management subsystem, an authoring environment, and a publishing or delivery subsystem, In parallel with the customization and development of a system architecture, several writers began to implement the Information Mapping techniques to provide content to be put into the system being developed.

THE NEED FOR MODULARIZATION

As the quantity of technical information grows and the number of delivery methods increases, technical writers and course developers at Cisco System, Inc., have found redundancy both in the content and in the effort required to produce technical information for our customers.

To address these issues, as well as to improve the quality of the documentation and training courses, we decided to investigate the concept of modular documentation. To this end, we created a team to define modularization objectives and methods, assess the audience, prepare a prototype, and develop modularization standards. The team consisted of representatives from documentation, training, and customer engineering.

SELECTING THE METHOD

The modularization team identified two methods to investigate, the Information Mapping methodology (offered by Information Mapping, Inc.) and a technique called modular documentation (taught in the University of California, Santa Cruz, extension program by JoAnn Hackos).

The team took classes in both methodologies and found strengths and weaknesses in both. They particularly liked the ideas of a single chunk of information with a single concept, a presentation style that is consistent for each type of information, and clear headings. They also liked the methodology for developing information that involved considerable of up-front planning, but once the planning was done, the writing would fall into place.

The team decided to focus on the Information Mapping methodology, but also to implement aspects of both methodologies as they applied to Cisco’s needs.

DEFINING INFORMATION MAPPING TO FIT CISCO'S NEEDS

After selecting the Information Mapping methodology, the modular documentation team identified the following tasks to perform:

- Learn about Information Mapping methodology
- Customize the methodology
- Develop a protoype system
- Perform analysis and usability testing
- Implement the system and train the department

Educate Ourselves about Information Mapping Methodology

Cisco arranged an in-house training course in Strategies for Developing High-Performance Documentation from Information Mapping, Inc. This course outlined the theory and application of information mapping.

Customize the Methodology

We customized the Information Mapping methodology in two broad steps. In the first step, we created a series of exercises based on our existing deliverables (technical manuals and courses). The purpose of these exercises was to understand the concepts of Information Mapping and to determine how they could be applied to our deliverables. These exercises provided a forum to work through many questions and issues, such as how we would handle vital command-related information that could not fit into a procedure table, how we could ensure that warnings were always associated with their appropriate subjects, how our style guidelines and templates would be affected, and how we could ensure adherence to the methodology.

In the second step of customization, we created a series of information maps that would serve as templates, or road maps, for determining what information was necessary for each deliverable and how that information would be sequenced. This, coupled with a high-level outline of each deliverable, would give us direction when it actually came time to write.

Develop a Prototype System

The biggest challenge in implementing modularity is determining how to store and retrieve reusable chunks of information. We were told that this was an important issue-and this was the case.

After much research, we settled on a strategy that allows us to write at the information map level (roughly equivalent to a section or subsection), yet store at the information block level (roughly equivalent to a paragraph) by describing an interface between the authoring subsystem and the document management subsystem that would “burst” a map into its smaller components.

After installing the software and writing the content, we will be ready to develop a prototype system. The prototype will demonstrate development of content through the application of structured templates, the generation of deliverables from this content, and the explanation of how will we store and retrieve information and set the attributes of information blocks, maps, and products.

Perform Analysis and Usability Testing

After the prototype is developed, the team will analyze both the system and the process, and create a small user's guide. The team will then give the system and user's guide to an in-house usability team to be tested. Finally, the team will recommend to senior management whether or not to implement the Information Mapping methodology at Cisco.

Implement the System and Train the Department

If the team decides to continue the Information Mapping methodology, and if senior management agrees, the final steps are to implement the Information Mapping methodology in the department, and train the writing and training staff. Team members might serve as trainers and project leads as we implement the Information Mapping methodology.

DEVELOPING A SYSTEM TO SUPPORT THE INFORMATION MAPPING METHODOLOGY

Soon after we started looking for a system capable of implementing our Information Mapping methodology, it became clear we would not find an off-the-shelf system that would do exactly what we needed. Our requirements, specified in a product requirement document, pointed to a system architecture consisting of three main subsystems: a document management subsystem, an authoring environment, and a publishing or delivery subsystem.

Document Management Subsystem

The document management subsystem stores and retrieves templates and "instances" of information blocks, maps, and products. Templates are empty placeholders that define the structure and attributes of each information object (block, map or product). Instances are templates with specific content. The document management subsystem handles issues such as revision control, information object locking, and searches based on different criteria. This subsystem also handles workflow tasks to support information object content reviews, content change notifications, and format conversions. The document management subsystem manages a variety of file and application formats, but for the first phase of implementation it will handle only Standard Generalized Mark-up Language (SGML).

Authoring Environment

The authoring environment is a desktop publishing application that enables structured writing and supports SGML. It provides an efficient and robust way of creating carefully structured information blocks, consisting of text and illustrations, and associating these blocks to create

meaningful information maps and information products based on predefine templates. The authoring environment serves as the front-end to the document management subsystem and allows the author to work at the appropriate level (information block, map, or product). For example, an author can work at the information map level. When the author saves the map in the document management subsystem, the map is "burst" into blocks that are stored individually, the revision number is modified, and the old copy is overwritten as specified by the author.

Delivery Subsystem

The delivery subsystem is responsible for retrieving information objects from the document management subsystem, converting them into one of several predefined formats, and "pushing' them to one of several possible destinations. Some of the supported output formats are PostScript, Hypertext Mark-up Language (HTML), and native FrameMaker. Some of the supported delivery destinations are local printers, a print vendor, CD-ROM, World Wide Web servers, training partners, and business partners.

Challenges

The main challenges we are facing as we design and plan the system implementation fall into several broad categories. One challenge is handling of information object attributes—what are the correct attributes, how do we store and manage them, what is the best way to use them in searches.

Another challenge is the integration of the various subsystems and the ability to add new functionality and applications if we need to enhance the system. Handling legacy information and tools is also an issue we need to deal with.

Finally, while we believe that automation and system support of Information Mapping will improve the quality of our deliverables, as well as the quality of life of our content creators and reviewers, it is not clear, yet, by how much. System implementation and enhancements will require constant balancing between cost and effort, and performance and results.

TESTING THE METHOD: TRIALS ON A REAL DOCUMENT

After the modularization team chose the Information Mapping methodology and developed preliminary documentation, one technical writer in the department implemented the methodology in two documents that she was developing. The reasons for doing this were for the writer to provide feedback to the modularization team and for the team to get a feel for what a document written with Information Mapping techniques would look like and what processes would be involved in preparing such a document.

The writer first prepared a draft of several chapters in our existing format, and then worked with members of the modularization team to refashion the chapters according to the Information Mapping methodology. The remaining chapters of the manuals were then written using the Information Mapping techniques.

When comparing the two manuals written with Information Mapping techniques to our standard manuals, several major differences emerged. First, the information flow in the manuals written using Information Mapping techniques were much clearer, making the technical content and meaning easier to follow. Second, the writing in each chapter and the organization between chapters was more consistent. Finally, the table of contents in each new manual acted as an easy-to-follow summary of the manual's contents. This view of the table of contents as a summary of the manual pointed out the need to revise our templates to allow for more heading levels. We believe these differences improve the quality of the manuals.

The writer who prepared these manuals initially disliked the Information Mapping methodology. She also found that some of the techniques, especially those regarding the wording in headers, differed dramatically from previous styles in which she had written. However, once she became familiar with the Information Mapping techniques and saw the results of the modularization, she admitted that the methodology was useful for creating readable, high-quality customer documentation.

Aviva Garrett
Haggai Mark
Jan Johnston-Tyler
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California, 95134-1706 U.S.A.
408 526-4000
Aviva Garrett is a technical writer, Jan Johnston-Tyler is a manager of technical writers and course developers, and Haggai Mark is an electronic performance support system specialist at Cisco Systems, Inc., a networking products company in San Jose, California.