



## The Role of an Information Architect in the Technical Information-Development World



*JoAnn T. Hackos, PhD, CIDM*

In working with information-development groups who want to move into content management and a structured writing environment, I often find that the potential for the role of information architect is not well understood. Some groups believe that an information architect is a technical expert responsible for the tools and technology used to author, maintain, and publish content. Others believe that an information architect need simply concentrate on formatting final deliverables such as print, HTML, and help. Still others want a programmer who can produce a DTD (Document Type Definition) or similar code that reflects the existing format of the organization's book.

I believe that each of these assumptions falls short of the actual role of an information architect, especially one who focuses on meeting the needs of those who use the information as well as the needs of those who author.

An information architect in the technical information-development world has core responsibilities that go well beyond the development of style sheets for print or online delivery or the use of XML. The information architect must assume responsibility for

- ◆ investigating the requirements of the customers for the content and structure of information deliverables
- ◆ understanding the underlying content structure of the information types that authors must produce and developing standards based on these structures
- ◆ instantiating business rules into the structures to support authors and encourage compliance
- ◆ creating structures that promote finding and reusing content in multiple contexts, including metadata schemes to label content appropriately for delivery to customers
- ◆ creating an authoring environment that accommodates both the preferences of authors and the needs of the business for compliance with standards
- ◆ developing standards for content assembly in multiple media that meet customer and business requirements
- ◆ building style sheets that apply appropriate formatting to content for each type of deliverable

Let me begin by stating an important assumption behind my definition of the information architect role. I believe that much of the technical information we produce today in support of products should be developed as topics that can be assembled into final deliverables, rather than produced as final deliverables. As such, I have been involved in the development and promotion of the Darwin Information Typing Architecture (DITA), a recently approved OASIS standard that advocates a topic-based architecture in which topics are information structures that can be understood in and of themselves by readers. Thus, in planning for a topic-based architecture and moving a group of information developers from book-based to topic-based and structured authoring, the information architect plays a pivotal role.

Let us examine the responsibilities in more detail.

### INVESTIGATING CUSTOMER REQUIREMENTS

Despite all the support heard in the field of information development for a customer-centered approach, too often groups embarking upon content management and the development of a new Information Model for their organizations ignore the customer. By failing to investigate thoroughly the customer's need for information, we are likely to spend a great deal of time and effort creating a structured Information Model for content that is needed by no one.

The best approach to developing a new structured model for authoring and delivery of information is to begin with the minimalist agenda. According to the tenets of minimalism, technical information is most effective if it

- ◆ supports the user's need to take an action-oriented approach (i.e., using a software or hardware product effectively in a business or personal setting)
- ◆ is anchored in the user's domain, supporting the user's real world goals, not goals that are derived from the product (i.e., registering a patient in a hospital rather than using the file menu)
- ◆ focuses primarily on information needed to solve problems (i.e., troubleshooting and problem-resolution information)

- ◆ supports users who want to find just the information they need, users who are primarily interested in performing tasks, as well as users who wish to study information to gain knowledge and understanding (i.e., conducting a search for information, enabling successful task performance, and providing background and conceptual information)

When information architects begin with a minimalist approach, they are often able to reduce verbiage, eliminate content that serves no one, and focus content more precisely. Thus, the remaining content is ready for structuring.

**DEFINING THE UNDERLYING STRUCTURE**

In the process of creating a new Information Model to support structured authoring and content management, the information architect must determine the underlying structure of the technical content, not its superficial structure. The underlying structure of the content reveals the intended meaning of the text and the sequence that best supports the user’s performance and understanding. The underlying structure is not to be found in the format of the text for desktop publishing.

We often are asked to review deeply flawed Information Models created by so-called information architects who look only at the surface, building templates that support formatting styles rather than units of content. Table 1 illustrates the differences.

In understanding the underlying structure, the information architect must see beyond the superficial structure and format and identify the meaning of the content. In addition, the architect must analyze many examples so that the “ideal” structure is identified, despite the myriad of errors that occur in a text because of poor design or poor execution. As we know, much poor writing and design creeps into our documents through their years of development.

The information types identified should be free from poor design and writing and should conform to the best possible underlying structure. An ideal task topic, for example, might assume the structure illustrated in the example above.

**INSTANTIATING BUSINESS RULES**

Once the information architect has uncovered the underlying structures and accommodated the needs of the information customer, the new standards for authoring can be expressed in the Information Model. The model identifies the standard structures and encourages compliance through hierarchical DTDs or other template designs, authoring guidelines, and editorial reviews for quality assurance. Accordingly, the information architect, in concert with other team members, is responsible for ensuring that

- ◆ templates, which may use DTDs to support XML-based authoring, are properly constructed

SUPERFICIAL STRUCTURE	UNDERLYING STRUCTURE
Heading 1	Topic title
Paragraph	Pre-requisites
Paragraph	Context
Bulleted list	Definition list
Paragraph	Warning
Numbered list	Action step
Indented paragraph	Step result
and so on	and so on

**TABLE 1: A COMPARISON OF THE STRUCTURAL DIFFERENCES BETWEEN SUPERFICIAL AND UNDERLYING STRUCTURES IN DOCUMENTATION**

- ◆ guidelines are prepared to train authors
- ◆ a process is created in which editors or peers in the role of editors ensure that the author’s work conforms to the new structures

These activities support the rules that the business needs to ensure consistency in its information source and deliverables. For example, a business rule might state that all topics must begin with a title that is appropriate to the information type (i.e., task, concept, or reference, to use the core DITA information types).

**CREATING STRUCTURES THAT SUPPORT CONTENT REUSE**

Much structured authoring today uses XML (eXtensible Markup Language) to instantiate business rules and to assist authors in complying through DTDs or XML schemas. However, because so many so-called information models fail to account for the underlying structure of the content, the XML structure often fails to support content reuse. Only by labeling content with semantic information does it become searchable and addressable in multiple contexts. For example, by using the standard topic structure of DITA, authors label their topics according to the information type. A topic might be a task, a concept, or a reference or another specialized information type needed to support a particular type of content for the customer. Once the topic is correctly labeled, authors may search for it by type in a content repository or use programmed scripts to filter the content for delivery to an output medium.

Let us say, for instance, that we wish to develop a quick reference installation guide for a laser printer. If our content repository contains both task and conceptual topics properly labeled, we can prepare a script to search out and assemble only tasks for the installation guide. Further, if our task topics label and isolate the title and the set of action steps by using semantic markup, we can address that particular content and separate it from other components of the task for the quick installation guide versus the longer, more substantive version.

Only through semantic markup, in addition to metadata at the topic level, are we able to address and use content at many levels. Semantic markup using XML might allow us to label some text as a warning `<warning>Do not look at the arc.</warning>`. If we were to search for and locate all warnings in our documents, we could store them separately to facilitate their reuse.

### MODELING METADATA

Semantic markup in XML labels content according to its intended meaning rather than its format. In doing so, the semantic XML tags serve as a form of metadata, or information about information. If an author labels a content unit as a syntax diagram, we can use that label to address, filter, and output the content. However, the tags are not the only metadata that the information architect must address in the Information Model.

Metadata may take many forms to be useful in managing content. Some metadata is used to manage workflow, a process in which content is routed from author to reviewer to editor to approver to publishing. By labeling topics by their state in a process, one can learn which topics are still in draft form or which are in the hands of the editor.

Some metadata labels topics with information that will assist in recalling its history. We may inquire through metadata, for example, who originally authored a particular topic and by whom it was updated most recently. We may also learn on what date the topic was reviewed and whether it is ready for final approval.

Metadata can help us to know the languages into which a topic has been translated or whether it has been localized. Metadata may be used to identify topics associated with particular products or release dates or whether a topic is appropriate for particular users, be they programmers, administrators, installers, maintenance engineers, or end users.

The information architect, together with colleagues from the organization, must build a metadata model that reflects internal requirements for workflow, process management, and search. At the same time, the metadata model must support the users' search for information, the delivery to the users of content customized to his or her special needs, or a facility in which users can select for themselves the content they wish to receive.

### CREATING A USABLE AUTHORING ENVIRONMENT

I have used XML authoring as an example for creating an information architecture that is standardized, idealized, and supports reuse. However, the information architecture must also facilitate authoring. Depending upon the information architect's organization, authors may need to create content in many environments.

For the professional technical communicator, the information architect may want to provide an XML editing environment. Many writers prefer to work in an editing environment

in which they can see the XML markup, as well as view what the content may look like once a particular style sheet is applied.

In a business environment in which many people create content as a secondary part of their jobs, the information architect may want to hide the XML markup by using forms-based information entry or modified versions of MS Word or other MS Office applications. The architect may support a desktop publishing system like Adobe FrameMaker that supports both XML and a fully formatted view of the text.

In any case, the information architect is responsible for finding solutions that will encourage authors to use standard structures and follow the business rules for those structures.

### DEVELOPING STANDARDS FOR CONTENT ASSEMBLY

Although I quite clearly advocate a topic-based, structured authoring environment, topics must be assembled into final deliverables. The list of deliverables might include print publications, PDFs delivered on the web or through CD-ROMs, HTML presentations in web browsers, help systems embedded in software, or others. The list of possible media for information delivery continues to grow as new devices become available.

Although the information architect may not be responsible for creating the outputs, the architect is responsible for ensuring that the outputs are structurally consistent. DITA refers to the structured output as "maps." A DITA map might be a table of contents for a print or PDF output, providing a standard sequence in which a set of topics is assembled. A DITA map might provide for the linking relationships in an HTML-based web site in which tasks, concepts, and reference topics are interrelated.

The Information Model should include standard topic sequences or relationships for standard document types. For example, an organization may produce many installation guides for different products. By standardizing the sequence of chapters, their names, and their typical content, the information architect ensures that the customers experience a repeatable and consistent structure that is more easily accessible.

In developing the Information Model, the architect must consult with those responsible for mapping the final assembly of topics so that the sequences and interrelationships of topics may be standardized.

### BUILDING STYLE SHEETS TO ADD FORMAT TO DELIVERABLES

The information architect may not, in fact, be completely responsible for the styling of the deliverables. In some organizations, that role is assumed by a graphic designer or other specialist in layouts and fonts. However, it is important that the information architect ensure that the designs do not obscure the semantic message of the content. Formatting may be used, for example, to differentiate between print and online styles to promote readability. Formatting may suggest to read-

ers that they are viewing a concept rather than a task. Formatting may effectively ensure that reference information is easily scannable in correct tabular layout or other alternatives.

As technical communicators recognize, the appearance of information on page or screen influences how the reader perceives and assimilates the content. Indeed, the medium is an integral part of the message.

**THE INFORMATION ARCHITECT’S PROCESS**

In this discussion, I have focused on the many responsibilities of information architecture in establishing an environment that fosters structured authoring, compliance with business standards, delivery of content in a variety of contexts, and support for multiple output media.

We might illustrate these responsibilities in this way:

Information user analysis → Information authoring → Information selection, filtering, and assembly → Information delivery

The information architect is responsible for establishing the framework in which consistent, reliable content is produced and delivered to customers and doing so in a way that promotes the business goals of the organization.

It is indeed a daunting task. Nevertheless, the challenge of doing the task well is exciting, placing the information architect at the center of the development of sound technical information. □

**REFERENCES**

*Content Management for Dynamic Web Delivery*  
 JoAnn T. Hackos, PhD  
 2002, John Wiley & Sons  
 New York, NY  
 ISBN: 0471085863

*Information Architecture: Blueprints for the Web*  
 Christine Wodtke  
 2002, New Riders Publishing  
 Indianapolis, IN  
 ISBN: 0735712506

*Information Architecture for the World Wide Web: 2nd Edition*  
 Louis Rosenfeld and Peter Morville  
 2002, O’Reilly Media, Inc.  
 Sebastapol, CA  
 ISBN: 0596000359

Darwin Information Typing Architecture (DITA XML)  
*XML Cover Pages on DITA*  
 <<http://xml.coverpages.org/dita.html>>