

Beyond Schemas: Planning Your XML Model

By Jennifer Linton

Copyright © 2007 O'Reilly Media, Inc.

ISBN: 978-0-596-52770-9

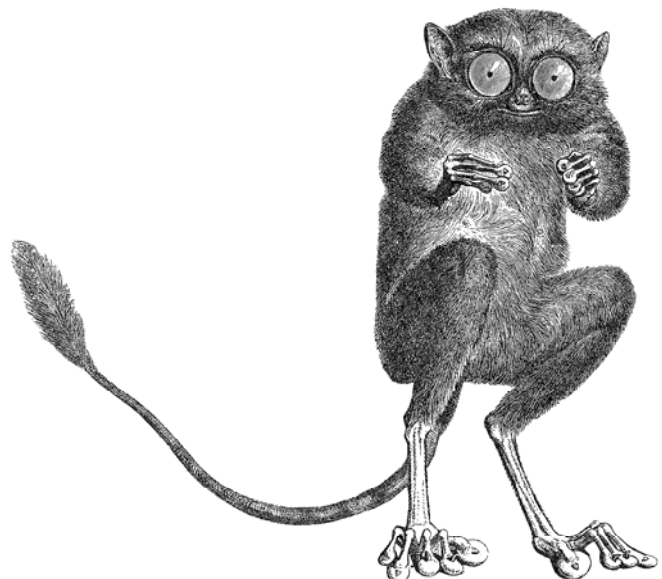
Release Date: March 2, 2007

Have you ever wondered how to get started writing your own schema? As you prepare to create your schema, you must consider a number of factors. This guide explains each of those factors in detail and recommends an approach for documenting your schema development plan in an information model.

Your information model can not only be used as a planning mechanism to develop your schema but can also be used as a training resource and as a reference guide for those using the schema after it is developed. By putting a well-thought-out information model in place, you are bound to produce a schema that you can use indefinitely and build upon easily.

Contents

Conducting a User Study and Competitive Analysis	2
Identifying Structure	4
Creating a Reuse Strategy	6
Developing an Information Model	9
Information Types	10
Content Units (Semantic)	18
Body Elements (Syntactic)	21
Inline Elements (Syntactic)	24
Documents and Deliverables Structures	26
Metadata Attributes	31
File Structure and Navigation	34
Naming Conventions	35
Conducting a Small Pilot Project	40
Putting It All Together	41
References.....	41



While many developers see using XML as a matter of creating a schema and applying it, this simple-sounding path has a number of pitfalls. Heading directly into schema writing without spending time analyzing what information and structure needs to be represented may work in the short term—but in the long term, it can turn your schema into a barrier to creating efficient processes rather using it as an aid.

Avoiding this fate calls for thinking ahead and modeling information rather than defining XML structure immediately. Developing an information model requires analyzing your information to define structures, standards, and policies to create information in a consistent manner. When done well, information modeling includes:

- Completing a user analysis
- Creating a reuse strategy
- Developing information model definitions
- Conducting a small pilot project

Every part of the modeling process is essential to a successful implementation. Although there are many kinds of information—such as e-commerce web sites for ordering books, policies, procedures, recipes, letters, workshop lessons, log files, and more—technical documentation is a broadly useful field to treat as an example.

Technical documentation describes products and/or how to use a product of any sort. There are many types of technical documentation, such as user guides, maintenance manuals, quick reference cards, product catalogs, and more. End users receive technical documentation by a variety of methods, including via paper-based and online web-based delivery, through product interfaces such as help systems, or even as labels on the product itself. Technical documentation provides a full suite of information to analyze.

While the user guide example here won't be exactly the same as the information you're modeling, treat the modeling process as an approach for conducting a similar process for your information, whatever it may be.

Conducting a User Study and Competitive Analysis

When you first begin any information development project, what is the first thing you do? Do you jump right in and investigate the top-rated tools to help automate your process? Do you just start writing your information without planning in advance? Do you understand what your users really need?

Users are rarely considered at the beginning of the planning process. However, if you don't understand your users, the likelihood of needing to redo your information deliverables at a later time increases. An initial user study will help you deliver accurate and usable information so that your users can find it.

When conducting a user study, you must start by selecting a significant group of users to identify their needs. Your user group should include many different kinds of users—some whom you may have not even considered in the first place. First, *external users* are the people who use your product. These users may vary from people who are just beginning with your product to advanced users who need less guidance about how to use the product, but need more specific and detailed information. You may also find that some users like to use different types of media including online or paper delivery, or may even learn how to use the product by taking a class or tutorial in which you deliver your information using a more interactive approach. It's best to consider all the possibilities when identifying your user group for your study.

Another set includes *internal users*. These may include the information developers, maintenance technicians, support, or even the sales and marketing staff. All of these people must learn how the product works in order to provide adequate information to the external end users. How do you deliver up-to-date information to these individuals? What are they looking for? How do they use the information daily?

After identifying your user base to set up interviews with the internal users, it's a good idea to prepare a process to follow throughout the user study to ensure consistency from one user to the next. Typically, when conducting a user study, it is best to observe your users as they conduct tasks and work through how they would find and use your information. One way to work out a process is to prepare multiple user scenarios for your users to interpret in an interview. For example, ask them to find a particular piece of information that you know exists in your deliverable and observe how they do this.

Many times, people think of user studies as merely asking users a series of questions or conducting a survey. Surveys and questions will limit your full understanding because your users will only be able to express their needs according to your interpretation of their needs, not their own.

Consider, for example, that you are conducting a user study about a web site you must create. It will contain information about how to use a new computer your company is designing. Hopefully, your product development team has already identified the best way to design the product and offered the most efficient ways to complete the tasks for the features that your computer provides. Your first approach to understanding what information your users need, and how they find it

for the new computer, is to identify how they use the information that you provided about previous computer versions. Or, if you don't have this, ask them to search through a competitor's web site and use the information they find about the competing product.

As you walk through the site with a user, ask him to verbally explain to you every step he takes, including describing what he clicks on (this will help determine how he searches for information and how to set up your navigation), as well as what he is looking for, whether he found it, what he finds frustrating, and most importantly, what he likes about the web site. Continuing to keep your users actively engaged in the user study will encourage them to provide you with meaningful results.

After interviewing and observing about 10 to 15 users, you may begin to find redundancies and similarities among them. Consider each of your users: beginners, advanced, internal, external, and any one else.

Once you have gathered your notes about each user interview, you can begin to determine the information your users need and how you may structure your information and navigation to make it easiest for the users to find. As a result of your user study, you should have solid scenarios on which to base your information development and delivery. The scenarios you identify can be used throughout the task as you work through your model and pilot project. Consider even translating the scenarios to visual representations. *Visual*, or *graphical*, representations help users more quickly identify with user needs. These are also known as *process flows*.

Identifying Structure

Once you have identified how and where your users want to find the information needed to complete their tasks, you must consider how to structure it. Providing your information in a structured way makes it easier for users to find the same information efficiently and consistently every time for different subjects.

When you first learned how to write a letter, your teacher most likely explained that every letter has a standard structure that includes a date, address, greeting, body, and salutation. The structure for a letter is different than the structure you would use for a recipe that includes ingredients, preparation instructions, and a result. Each of those types of information contains different structures than a user guide. User guides typically provide solutions for how to do some task with your product, some kind of background information about the product, or even some specifications about the product.

In order to identify structure in your information, you must conduct an internal analysis of your information. Identifying structure does not mean that you identify the different heading styles, list styles, and paragraph styles. Identifying structure

based on formatting is referred to as *syntactic structure*. These pieces of your information are only aesthetics and formatting that you can change to match your users' wants.

An information analysis must focus on the content of the information—i.e., the *semantic structure*. What is the information trying to say, and what is the purpose of the information? By understanding the underlying content, you can develop a structure to follow each time you create a piece of information.

Take, for example, two tasks in your user documentation. One is broken out into steps that your user can easily read through, and the other is a more narrative task that your user may not be able to find or identify as a task because it has a different structure than the rest of the tasks you develop. Delivering your information in a consistent structure ensures that your users can identify the information more quickly and readily when they need to find something.

As you work through your information deliverables, annotate what the structure should be for each piece of information. During your analysis, you may find that each paragraph should hold a certain structure to provide a consistent flow through your information deliverable. Don't be afraid to get into the granular pieces of your information during the analysis. The more you work through the details of your structure in the beginning of your project, the more likely you will meet your user's needs. Example 1 shows a user guide section, annotated to show how you might structure it.

What is the Super Computer 210? (title; summary)

The Super Computer 210 is the best computer available in the world. It provides you with features such as a hidden camera, translation to any language, and interchangeable screens that allow you to write on one and type on another. It also provides a built-in scanner, printer, and copier, and it has 12 USB ports for you to plug in more add-ons. Everything you need in one little computer—and it is little. It is the size of a mobile phone. If you are worrying about being able to see the screen, don't. You will be able to project the screen and an interactive keyboard on the wall and desk respectively so you can easily use the tools and applications. You may also be excited about having 300 possible games to play whenever you want. **(This is the feature summary.)**

Computer backups (title; concept)

Computer backups are a way to take a snapshot of your computer files on a particular day at a particular time. The Super Computer 210 allows you to schedule up to 200 backups of your files per day. You can also create custom backups that allow you to back up just photographs or just music files. Because your computer has 100,000 gigabytes of space, you should have no problem creating as many backups as needed. However, if you do find you are losing space, you may want to consider decreasing the number of backups or purge backups that are over two weeks old. **(This is the overview.)**

Backing up your computer (title; task)

It is important to back up your computer routinely once or twice a year so that you don't lose any of your files. You can select which pieces you back up for each backup. Use the following process as a guide to back up files on your computer. **(This is the short summary/task purpose.)**

Prerequisite: In order to back up your computer, you must turn it on and log in. **(This is the prerequisite.)**

1. Press Start→All Programs→Accessories→System Tools→Backup.
2. In the Backup Wizard, press Next.
3. Select "Backup files and settings" and press Next.
4. Select "My documents and settings" and press Next.
5. Specify a location to which you can save your backup.
6. Press Finish.

(These are the steps.)

Please see the example screens for the Backup Wizard at <http://www.supercomputers.com>. **(This is the examples/related material.)**

When you back up your documents, it will save a file that you can use to restore from later. **(This is the result.)**

Once you complete backing up your information, you may want to consider doing a system cleanup. **(This is the post-requisite.)**

Example 1. Annotating existing material reveals the structure.

Creating a Reuse Strategy

After identifying structures in your information, consider creating a reuse strategy. How do you want to take information from one place and reuse it in another? By knowing the structures of your information before creating the strategy, you can decide how to deliver your information and at what granularity it needs to be maintained. Once you identify the structures, begin to outline the categories in which you will reuse your information.

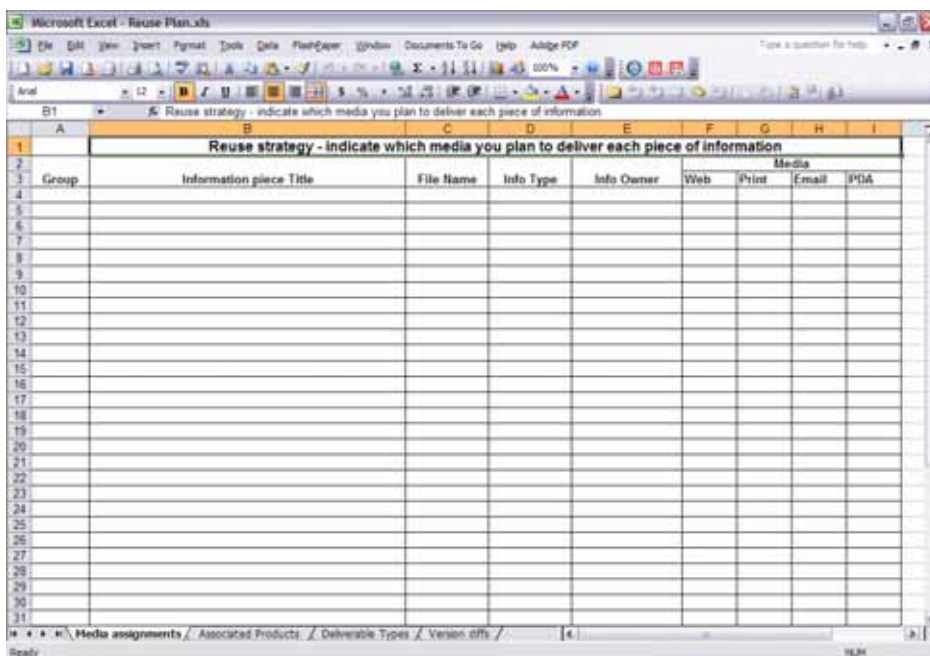
Reusing content provides the best return on investment in most projects. Whether you create a web site containing a calendar, a news feed in more than one place, a

bill of materials for a product you are shipping out next week, or user guides explaining similar products, there are opportunities for reuse. Reusing information improves our ability to see patterns and structures.

There are two ways to reuse information. The first is to accidentally come across information you can use in your deliverable because it applies in some way or another. The other way is to plan for reuse in advance. If you do so, it is likely to be more beneficial than if you accidentally reuse pieces of your information.

The best way to plan for reuse is to set up a spreadsheet. Use this as an iterative planning approach to create your documentation. If you or someone else you work with have already authored a piece of information, you may not need to write it over again—maintaining your spreadsheet is how you will know if the information already exists..

Your spreadsheets may provide reuse categories between different product deliverables, different users, different media you deliver your information to, or even different versions of your information products. The example spreadsheets displayed in Figures 1 through 4 show how to set up a series of spreadsheets for maintaining different pieces of information in a user guide. If you duplicate your list of information pieces on each spreadsheet, you'll see that you can have reuse in many different areas, such as overlap in product and feature, media, versions, and deliverable types.



Reuse strategy - indicate which media you plan to deliver each piece of information								
Group	Information piece Title	File Name	Info Type	Info Owner	Media			
					Web	Print	Email	PDA

Figure 1. Spreadsheet for developing a reuse strategy based on which information is used in a particular kind of media

Microsoft Excel - Reuse Plan.xls

File Edit View Insert Format Tools Data Window Documents To Go Help Adobe PDF

Reuse strategy - indicate which products your pieces of information apply to

1	Reuse strategy - indicate which products your pieces of information apply to					
2	Products - Super Computer					
3	Group	Information piece Title	File Name	Info Type	Info Owner	210 220 500 5000
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						

Figure 2. Spreadsheet for developing a reuse strategy based on which information is associated with a specific product or with multiple products

Microsoft Excel - Reuse Plan.xls

File Edit View Insert Format Tools Data Window Documents To Go Help Adobe PDF

Maintenance/Support guide

Reuse strategy - indicate which deliverables your information pieces apply to

1	Reuse strategy - indicate which deliverables your information pieces apply to					
2	Deliverable					
3	Group	Information piece Title	File Name	Info Type	Info Owner	User Guide Training Tutorial Product catalog Maintenance/Support guide
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						

Figure 3. Spreadsheet for developing a reuse strategy based on which information is used in a particular deliverable

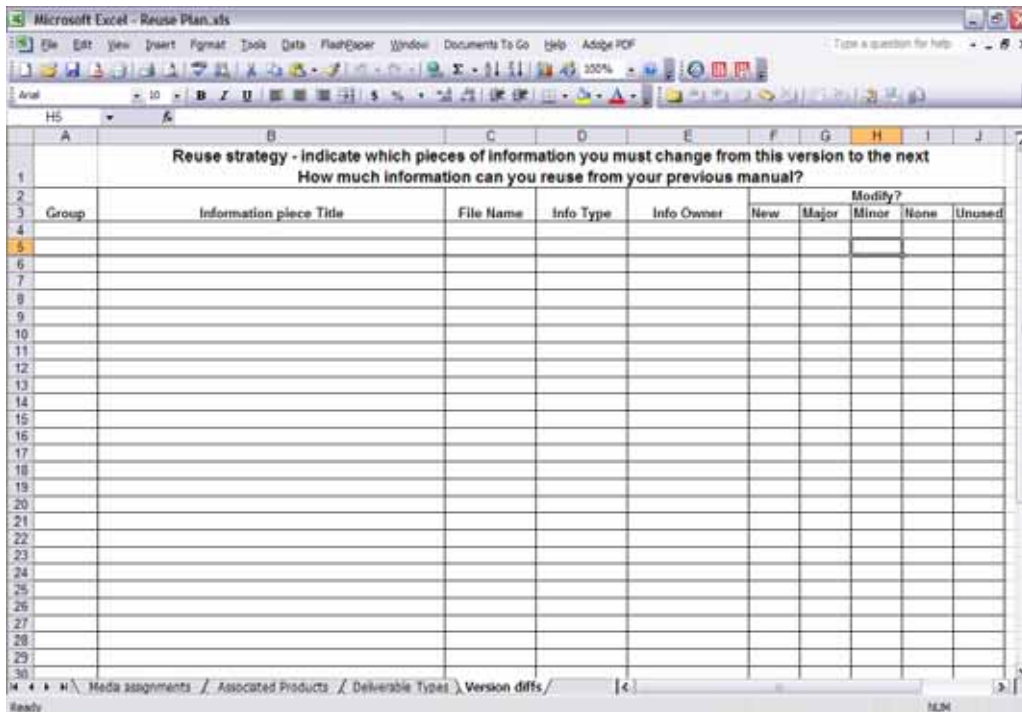


Figure 4. Spreadsheet for developing a reuse strategy based on which information can be used from the previous version of the information set

Developing an Information Model

An *information model*, in this context, is a document describing methods for creating standard information within your organization. This document is a way to formalize the analysis you conducted to identify the structures needed in your information. The information model directly supports the results found from your user study and information analysis. Your users help identify what information they need. As a result, you can pinpoint patterns and methods for delivering the information to best fit your users' needs.

The result of information modeling is a document—a guideline for each person who creates the information to follow so he creates consistent information from one instance to another. If you did not provide a guide to follow, each person will most likely create his own information inconsistently from one colleague to another. Also, each person is likely to create their own inconsistent information.

What Are the Benefits of an Information Model?

You can use an information model as a training tool to provide your content contributors with a guide for creating their information. You may find that each job role is responsible for creating different information. But, if each person doesn't create her information consistently, your users will have more and more questions.

An information model also provides a way to document best practices within your organization. If you provide a guideline for people to create information, you can ensure that a higher number of people are producing quality information that is more useful to your end users.

Creating an information model helps ensure standard practice, no matter what changes may come to your organization. Considering the constant practice of acquisitions and mergers currently, you have a better chance of staying consistent and even suggesting a consistent model for any future companies for whom you work.

Finally, your information model provides a documented method for the usability of your information products. Whether you create a web site, a user manual, a letter, a cookbook, or a log file, your information model is a way to represent a model your users will appreciate and find helpful. Users' need change. These changes can be represented in iterative corrections to your information model that keep you up to date with your users.

What Does an Information Model Provide?

An information model includes definitions for:

- Information types
- Content units
- Body elements
- Inline elements
- Documents and deliverable structures
- Metadata schema and file structure and navigation
- Naming conventions
- Information architecture terms

All of these components need documentation, though the documentation style and the process for creating the component will vary with the type of the component.

Information Types

Information typing is the process of discovering consistent and standard structures in your information deliverables. Information types are typically standalone pieces of information to which you can apply a standard structure. You can think of these as templates used every time you need to create a particular piece of information. Consider a user guide: as you analyze the guide, you may find particular pieces of information that help to answer your user's questions. For example, suppose a user

is looking for the answer to questions such as “How do I complete this task?” Or, “What does this term mean?” They may even be asking “What does this feature do?” In order to adequately represent and answer each question, you must create your information using a standard structure so your users can consistently find the answers to their questions.

In a typical organization, information types you work with may include parts of manuals, quick reference cards, log files, web pages, and more. In your personal life, however, you may reference different kinds of information. Perhaps you use a recipe to cook dinner, or a manual for a cell phone or a camera, or a travel web site to arrange your next vacation. Setting up your information types to follow a standard structure (i.e., a standard template), is the basis of your information model. Each piece of information you create should be consistent from one to another. If you lack consistency in what you create or view, the possibility of finding the information you need is reduced.

What Kind of Information Do You Include to Describe Your Information Types?

Information type definitions normally include the following:

- A definition of the information type’s purpose
- The XML element assigned
- A definition of where to use the information type
- The content units and other structural elements allowed in the information type
- A diagram of the information model
- The associated metadata
- A marked-up example demonstrating the XML structure
- A styled example of the information type

Each part of the information definition provides a guide for your information developers to follow when they begin to author information falling under this information type.

Consider again the sample user guide. Every user guide typically has three types of information: a task, a concept, and a reference structure. The example identifies a task information type definition based on your user studies and information analysis.

Example Information Type Definition

Part 1. The first part of the definition is a narrative definition of the type of information in your organization.

Notice how at this point you do not explain all of the details of the pieces that make up this information type. In the information type definitions, you describe only the full type of information and the overall structure it holds. You define the details in the content unit section that follows.

Part 1. Task information type

The task information type provides the user with a step-by-step approach to completing a task. A task helps the users answer the question “How do I?” Your task should neither contain background information about the product or features of the product, nor should it contain specific information about the product or the feature, such as specifications or parameters. The task is a simple structure containing only commands the user should follow to reach a result.

Each task must contain a title to identify the task subject. It must also contain the purpose or reason as a short summary of why the user wants to dive further into this task. The short summary must state what the end result of completing this task will be. After the short summary, you must provide prerequisites, steps to follow, a result, an example, and a post-requisite.

Part 2. The second part of the definition is to identify which XML element you will use to label your information type. Typically this element is the root element for your information type.

Part 2. Task XML element

```
<task></task>
```

This is the root element for the task information type, so it will be the first and last tag for each task you create. It is not an empty element—meaning, it must have an open and a close tag. This element can contain other content units as defined in Part 4.

Part 3. The third part of the definition describes when and where you would use this information type.

Part 3. Task information type location (contained by)

Use the task information type as part of user guides, maintenance guides, learning tutorials, and quick reference guides. Do not use tasks when creating product catalogs and sales and marketing brochures. A task may be used in a larger deliverable in print or on the Web, and it may be attached or linked to a concept or reference piece of information.

Part 4. The fourth part of the definition is to formalize the content flow. Identify the order of the content units. This step formalizes your structure for the information type. You should be able to take the information provided in this table and translate it directly to a schema or DTD definition. Each of the content units provided in Part 4 are described in detail in the section about content units in your information model.

When defining the occurrence of the elements in the information type consider the following rules:

required

Indicates that, in the DTD, you should identify this element with no indicator.

required (repeatable)

Indicates that, in the DTD, you should identify this element with a + sign.

'optional

Indicates that, in the DTD, you should identify this element with a ? sign.

optional (repeatable)

Indicates that, in the DTD, you should identify this element with a * sign.

When defining the sequence of the elements in the information type consider the following rules:

then

Indicates the defined element must be followed by the next element in the table. In a DTD this would be represented with a comma separator.

or

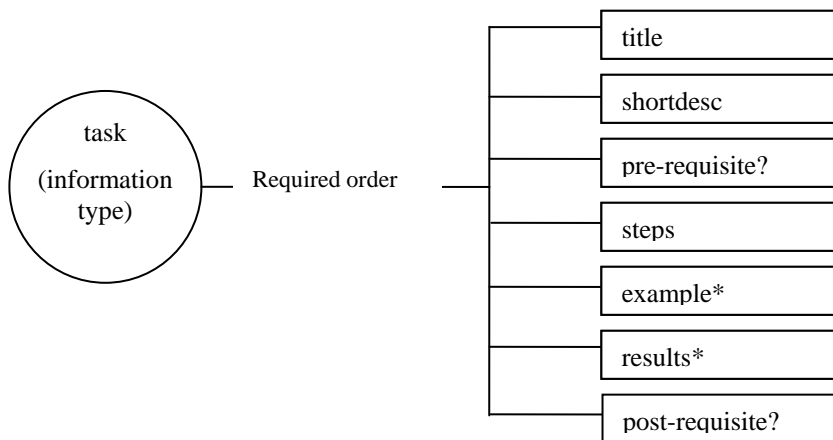
Indicates an option between the defined element and the next element in the table. In a DTD this would be represented with a pipe separator.

Part 4. The task contains the following elements:

Content unit label	Occurrence	Sequence
Title	Required	then
Short summary	Required	then
Prerequisite	Optional	then
Steps – multiple step elements are allowed within the steps element – steps is not a repeatable element	Required	then
Example	Optional (repeatable)	or
Result	Optional (repeatable)	or
Post-requisite	Optional	

Part 5. The fifth part of the definition is a graphical representation of what you defined in the content unit table. The graphical representation gives a quick-glance way to determine what content units are allowed in this information type. In the graphical representation, indicate the occurrence and sequence using the symbols described in Part 4.

Part 5. Task information type diagram



Part 6. *The sixth part of the definition is the metadata, or attributes, the information types allow. During your user study, you discover the majority of your metadata. However, you may need to conduct a separate study to identify exact terminology that both internal and external users understand. A process for metadata gathering is described later in the section “Metadata Attributes.”*

In your metadata attribute definition, include separate definitions for each attribute, the possible values you can use, the default value, and whether the attribute is required or not.

Part 6. Task information type metadata

Name	Description	Values (* = default)	Occurrence
product_name	The product name is an alphanumeric value that identifies the product related to this piece of information.	supercomputerseries* supercomputer210 supercomputer220 supercomputer500 supercomputer6000	Required
id	An anchor point. This ID is an alphanumeric value that indicates the virtual target for references by link.	An empty value. The information developer must use conventions as indicated in the naming conventions section for assigning a unique ID.	Required
user_skill_level	An optional metadata value to indicate the level of experience required of the end user. You may apply more than one user_skill_level to the information type delimited using a space.	all_users* beginner experienced internal_only external_only	Optional
language	The language in which the module is written.	English (EN)* French (FR) German (GE) Sspanish (SP)	Required

Part 7. The seventh part of the definition provides an example of an ideal representation of your information type. The purpose is to show the content creators which tags to use and in what sequence, as defined by Parts 4-6 of your definition.

Part 7. Task information type markup example

```
<task product_name="supercomputerseries" id="backingupcomputer"
user_skill_level="all_users" language="english">

  <title>Backing up your computer
</title>

  <shortdesc>It is important to back up your computer routinely once or
twice a year in order to not lose any of your files. You can select
which pieces you back up for each backup. Use this process as a guide
to back up files on your computer.</shortdesc>

  <prereq>In order to back up your computer, you must turn it on and log
in.</prereq>

  <steps>

    <step>Press start > All Programs > Accessories > System Tools >
Backup.
    </step>

    <step>In the Backup Wizard, press Next.
    </step>

    <step>Select "Backup files and settings" and press Next.
    </step>

    <step>Select "My documents and settings" and press Next.
    </step>

    <step>Specify a location to which you can save your backup.
    </step>

    <step>Press Finish.
    </step>

  </steps>

  <example>Please see the example screens for the Backup Wizard at
http://www.comstarcomputers.com. </example>

  <result>When you back up your documents, it will save a file that you
can use to restore from later.</result>

  <postreq>Once you complete backing up your information, you may want
to consider doing a system cleanup.</postreq>

</task>
```

Part 8. The eighth part of the definition provides a styled example of what your information type looks like, depending on what markup you use. Providing a styled example lets the authors know what to expect in output.

Part 8. Task information type styled example

Backing up your computer

It is important to back up your computer routinely once or twice a year in order to not lose any of your files. You can select which pieces you back up for each backup. Use this process as a guide to back up files on your computer.

Prerequisite: In order to back up your computer, you must turn it on and log in.

1. Press Start→All Programs→Accessories→System Tools→Backup.
2. In the Backup Wizard, press Next.
3. Select “Backup files and settings” and press Next.
4. Select “My documents and settings” and press Next.
5. Specify a location to which you can save your backup.
6. Press Finish.

Please see the example screens for the Backup Wizard at <http://www.supercomputers.com>.

When you back up your documents, it will save a file that you can use to restore from later.

Once you complete backing up your information, you may want to consider doing a system cleanup.

Content Units (Semantic)

Once you define each of your information types, you must define the pieces or building blocks that make up your information type. The larger pieces are content units. The smaller pieces are body and inline elements. The content units provide a semantic label to your information and to your structure. The semantic label allows you to search easier and style each piece differently, if needed.

The other elements that make up the smaller pieces of your content units provide more syntactic or stylistic functions. For example, a *post-requisite* is a semantic term indicating what a user must do following a set of steps. However, you may have multiple post-requisites that you divide up using paragraph elements to create a line break between each of the items.

Define content units using the same method you used to describe your information types. Including all parts of the definition ensures that your information developers have a guide or some sort of reference to create their structured information using the correct containers. If you don't provide your information developers with this guide, your information developers may begin to be creative with their XML markup. For example, your information developer may want the output to look a certain way, and therefore use a tag that they know will provide the right style they want. But, they are not describing the content correctly because they are using

incorrect markup. Your information model provides a guide and a standard way to create each information piece.

The following example shows how to create a definition for a content unit. Use the information type descriptions as a guide to create descriptions for each of your content units.

Example Content Unit Definition

Part 1. Post-requisite content unit

The post-requisite content unit is a statement or list of items that a user must follow after completing all the steps in a task. It must contain descriptive information, and it may contain unordered lists, hyperlinks, and options.

Part 2. Post-requisite XML element

```
<postreq></postreq>
```

The `<postreq>` element is not an empty element. You must provide an open and a close tag for well-formed XML. The `<postreq>` element can contain just text, or it can contain other body and inline elements.

When you define the contained-by part, indicate in which information types or other elements you can use this content unit. For this example, the post-requisite is only allowed as a child container element of `<task>`. For other content units, such as short description, it may be allowed in more than one information type.

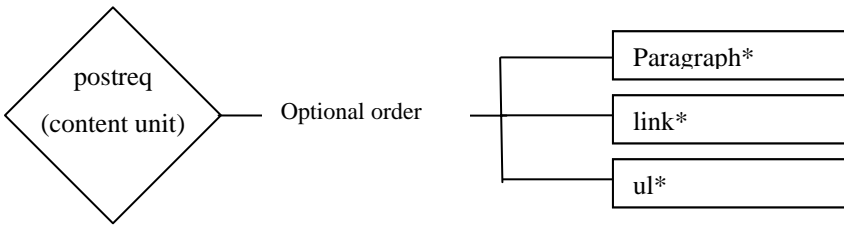
Part 3. Post-requisite content unit location (contained-by)

```
<task>
```

Part 4. Post-requisite contains the following elements:

Content unit label	Occurrence	Sequence (then/or)
paragraph	Optional (repeatable)	or
link	Optional (repeatable)	or
unordered list	Optional (repeatable)	

Part 5. Post-requisite content unit diagram



Part 6. Post-requisite metadata attributes

Name	Description	Values (* = default)	Occurrence
id	An anchor point. This ID is an alphanumeric value that indicates the virtual target for references by link.	An empty value. The information developer must use conventions as indicated in the naming conventions section for assigning a unique ID.	Optional

Part 7. Post-requisite markup example including surrounding context

```
<task product_name="supercomputerseries" id="backingupcomputer"
user_skill_level="all_users" language="english">

  <result>When you back up your documents, it will save a file that you
  can use to restore from later.</result>

  <postreq><p>Once you complete backing up your information, you may
  want to consider using other computer clean up options, including the
  following

  <ul>

    <li><link>Disk clean up</link></li>

    <li><link>Disk Defragmenter</link></li>

    <li><link>Desktop clean up</link></li>

  </ul>

  </p></postreq>

</task>
```

Part 8. Post-requisite styled example

Once you complete backing up your information, think about working on some other computer cleanup options, including the following:

- Disk cleanup: <http://www.comstarcomputers.com/support/diskcleanup.htm>
- Disk Defragmenter: <http://www.comstarcomputers.com/support/diskdefrag.htm>
- Desktop clean up: <http://www.comstarcomputers.com/support/desktopcleanup.htm>

Body Elements (Syntactic)

The body elements provide a way to further label your information to help with styling and search and retrieval. For example, you may use a `<p>` element to create line breaks between your paragraphs. The `<p>` element does not say anything about what kind of content is in the paragraph, but instead is used more just to create a formatting change. Inline elements can be purely formatting and typographic, such as bold, underline, italic, and so on. You can also create more specific labels, such as elements to help identify user interface buttons or keywords. Using specific labels allows you to easily search your information for the content. You can also style the labels using styles other than bold, italic, and underline, because they are not tied to a specific typographic style.

Body elements tend to have more complex definitions because you can use the elements in multiple locations.

Example Body Element Definition

Part 1. Paragraph element

Use the Paragraph element to indicate a separation of one paragraph from another. Although the `paragraph` element indicates separate paragraphs, it does not provide any semantic label as to what kind of content is in the paragraph. The paragraph body element's purpose is mostly to identify line breaks between paragraphs during styling.

Part 2. Paragraph XML element

```
<p></p>
```

The `<p>` element is not an empty element. You must provide an open and a close tag for well-formed XML. The `<p>` element can contain just text, or it can contain other body and inline elements.

Typically, a paragraph element is used in many different locations in any information, and it can use many other elements within it. Because it is so common, only a selection of the options is listed in the contained-by and contains sections. However, you may notice that there are other information types and content units in which you can use the paragraph element. In order to create a well-documented information model, create definitions for every element you have in your information.

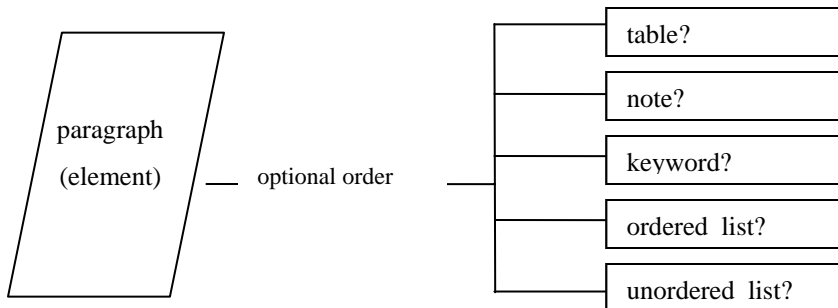
Part 3. Paragraph element location (contained-by)

<task>, <concept>, <reference>, <example>, <section>, <pre-requisite>, <result>, <pre-requisite>, <note>, <list_item> ...

Part 4. Paragraph contains the following elements:

Content unit label	Occurrence	Sequence (then/or)
table	Optional	or
note	Optional	or
keyword	Optional	or
ordered_list	Optional	or
unordered_list	Optional	or

Part 5. Paragraph element diagram



Part 6. Paragraph metadata attributes

Name	Description	Values (* = default)	Occurrence
id	An anchor point. This ID is an alphanumeric value that indicates the virtual target for references by link.	An empty value. The information developer must use conventions as indicated in the naming conventions section for assigning a unique ID.	Optional

Part 7. Paragraph markup example including surrounding context

```
<result><p>When you back up your documents, your super computer will save a file that you can use to restore from later.</p>
```

```
<p><note>The file extension for a backup when you are finished is .bak.</note></p></result>
```

```
<postreq><p>Once you complete backing up your information, you may want to consider using other computer cleanup options, including the following:
```

```
<ul>
```

```
  <li><link>Disk clean up</link></li>
```

```
  <li><link>Disk Defragmenter</link></li>
```

```
  <li><link>Desktop clean up</link></li>
```

```
</ul>
```

```
</p></postreq>
```

Part 8. post-requisite styled example

When you back up your documents, it will save a file that you can use to restore from later.

Note: The file extension for a back up when you are finished is *.bak*.

Once you complete backing up your information, you may want to consider using other computer clean up options, including the following:

- Disk clean up – <http://www.supercomputers.com/support/diskcleanup.htm>
- Disk Defragmenter – <http://www.supercomputers.com/support/diskdefrag.htm>
- Desktop clean up – <http://www.supercomputers.com/support/desktopcleanup.htm>

Inline Elements (Syntactic)

The inline elements can also provide a way to further label your information to help with styling, search, and retrieval. Inline elements can be purely formatting and typographic, such as bold, underline, italic, and so on. You can also create more specific labels, such as elements, to help identify user interface buttons or keywords. Inline elements provide support for semantic functionality but do not necessarily provide a structural purpose. Inline elements include `uicontrol`, `screen`, `bold`, and so on. Using specific labels allows you to easily search your information for the content. You can also style them using styles other than bold, italic, or underline because they are not tied to a specific typographic style.

Inline elements also tend to have more complex definitions because you can use them in multiple locations.

Each inline element may include other inline XML elements and some body XML elements. However, an inline element cannot contain an information type and usually does not contain a content unit.

Example Inline Element Definition

Part 1. User interface control element

Use the `uicontrol` element to specify button names, entry fields, menu items, or other user interface controls. You can use multiple `<uicontrol>` elements in a `<menucascade>` element to identify a sequence of menu choices in a nested menu.

Part 2. User interface control XML element

```
<uicontrol></uicontrol>
```

The `<uicontrol>` element is not an empty element. You must provide an open and a close tag for well-formed XML. The `<uicontrol>` element can contain just text and it can also contain other body and inline elements.

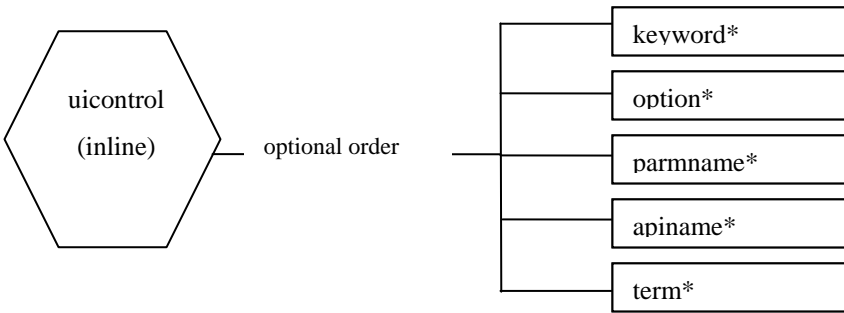
Part 3. User interface control element location (contained-by)

```
<title>, <shortdesc>, <section>, <example>, <desc>, <p>, <note>, <lq>, <q>, <slq>, <li>, <itemgroup>, <dthd>, <ddhd>, <dt>, <dd>, <figgroup>, <pre>, <lines>, <ph>, ...
```


Part 4. User Interface Control contains the following elements:

Element Label	Occurrence	Sequence
keyword	optional (repeatable)	or
option	optional (repeatable)	or
parmname	optional (repeatable)	or
apiname	optional (repeatable)	or
term	optional (repeatable)	or

Part 5. User interface control element diagram



Part 6. User interface control metadata attributes

Name	Description	Values (* = default)	Occurrence
id	An anchor point. This ID is an alphanumeric value that indicates the virtual target for references by link.	An empty value. The information developer must use conventions as indicated in the naming conventions section for assigning a unique ID.	Optional

Part 7. User interface control markup example, including surrounding context

```
<steps>
  <step>Press <menucascade><uicontrol>start</uicontrol> >
  <uicontrol>All Programs</uicontrol> >
  <uicontrol>Accessories</uicontrol> > <uicontrol>System
  Tools</uicontrol> >
  <uicontrol>Backup</uicontrol></menucascade>.</step>
  <step>In the Backup Wizard, press
  <uicontrol>Next</uicontrol>.</step>
  <step>Select "Backup files and settings" and
  press<uicontrol> Next</uicontrol>.</step>
  <step>Select "My documents and settings" and press
  <uicontrol>Next</uicontrol>.</step>
  <step>Specify a location to which you can save your
  backup.</step>
  <step>Press <uicontrol>Finish</uicontrol>.</step>
</steps>
```

Part 8. Post-requisite styled example

1. Press **start**→**All Programs**→**Accessories**→**System Tools**→**Backup**.
2. In the Backup Wizard, press **Next**.
3. Select "Backup files and settings" and press **Next**.
4. Select "My documents and settings" and press **Next**.
5. Specify a location to which you can save your backup.
6. Press **Finish**.

Documents and Deliverables Structures

Defining your documents and deliverables in the beginning of the project is often a hard task to complete because you don't actually know what content you have to include in your deliverables at the beginning planning stage. The purpose of defining a document and deliverable structure is to provide an initial standard for consistency that you can further refine.

Your document structure definitions provide an outline for what to include in the document deliverable. Your production coordinator is responsible for making sure each piece is in the final deliverable. The document structure definitions identify one or more information types needed to create the final deliverable.

Defining your document structure is very similar to how you define your information types, content units, and elements. Your document structure definition includes the following:

- Narrative definition
- Contents outline
- Diagram
- Metadata attributes

Example document structure definition

Part 1. *The document structure definition describes what a user would find in any user guide delivered from your company. It also describes who the intended user of this deliverable is.*

Part 1. User guide definition

User guides provide your users with a task-based approach to working with your product. Create your user guides with end users in mind, not technicians or sales staff. Some features of your product may be described in the user guide, but the features will directly support and provide background knowledge for a task the user needs to complete.

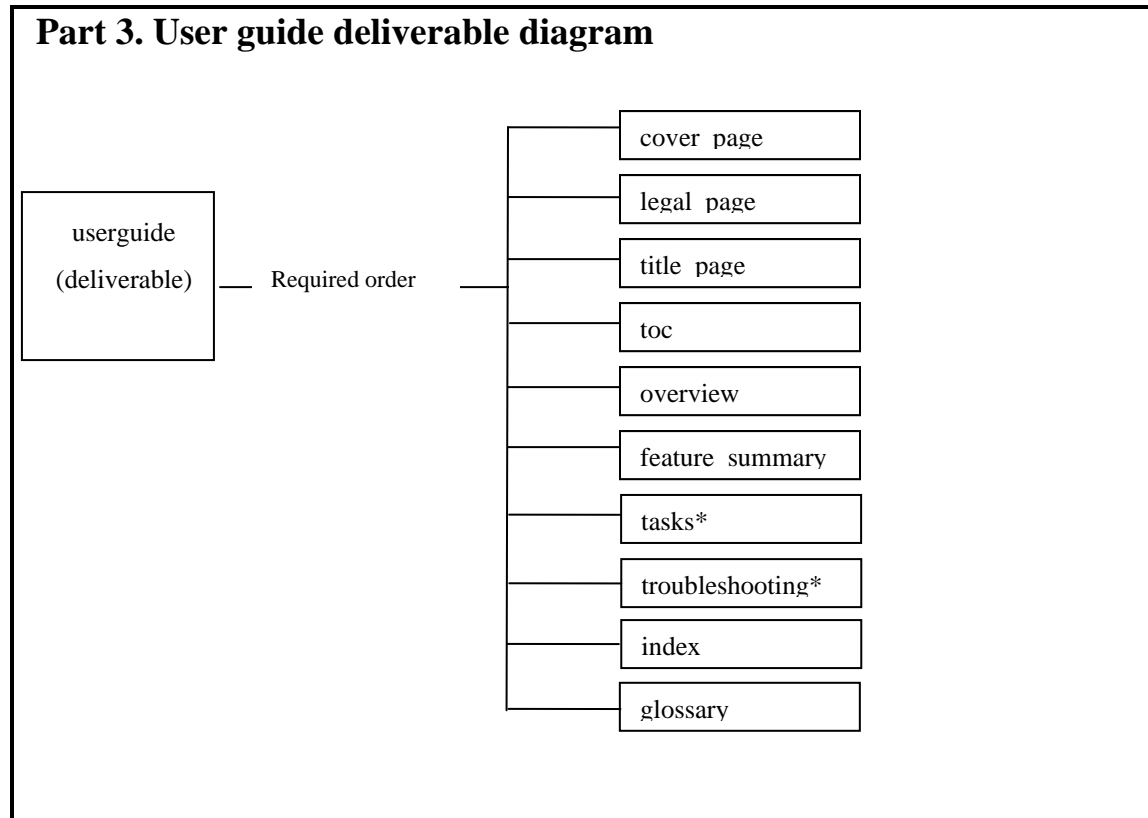
Part 2. *The second part of the deliverable structure defines the specific pieces that make up the deliverable and the order in which they should be arranged. If you are creating a web site, you may identify an outline for navigation at this point to describe how the users should maneuver through your site. The document structure outline is made up of a sequence of information types. Legal pages, overviews, product feature summaries, and so on are all considered different types of information used to present different information to your users.*

The document structure definition is not formally translated into an schema necessarily, but it does help to identify the occurrence and sequence any way. You may, after planning, decide you want to add elements in your schema to represent the full document structure as well as your information type structures.

Part 2. The User gGuide contains the following information types

Object name	Occurrence	Sequence (then/or)
Cover page	Required (not repeatable)	then
Legal page	Required (not repeatable)	then
Title page	Required (not repeatable)	then
Table of contents	Required (not repeatable)	then
Overview (concept)	Required (not repeatable)	then
Product feature summary (concept)	Required (not repeatable)	then
Task modules	Required (repeatable)	then
Troubleshooting modules	Required (repeatable)	then
Glossary	Required (not repeatable)	then
Index	Optional (not repeatable)	

Part 3. The third part of the document structure definition provides a visual representation of the deliverable structure similar to the diagrams created for the information types, content units, and elements.



Part 4. The fourth part of the document structure definition defines the metadata attributes associated with this type of information deliverable.

Part 4. The User guide metadata attributes

Name	Description	Values (* = default)	Occurrence
id	An anchor point. This ID is an alphanumeric value that indicates the virtual target for references by link.	Empty value – information developer must use conventions as indicated in the naming conventions section for assigning a unique ID	Optional
title	The title attribute allows you to include a document or deliverable title. You may use this attribute value in your stylesheets to automatically generate the cover page and title page	Empty value – information developer must use conventions as indicated in the naming conventions section for assigning a title	Required
product_name	The product name is an alphanumeric value that identifies the product related to this piece of information.	supercomputerseries* supercomputer210 supercomputer220 supercomputer500 supercomputer6000	Required
country	The country tag is used to specify the country to which this content is directed.	Canada North America* Italy Spain Portugal	Required
language	The language in which the module is written.	english (EN)* french (FR) german (GE) spanish (SP)	Required

Metadata Attributes

You can use metadata in many different aspects of your information deliverable. Even if you are not using XML, you may be assigning information labels to your information to categorize its importance. Because XML allows you to specify metadata at many levels of your information set, you can create extremely detailed information without actually writing anything for the user to view.

First, let's define the difference between metadata and attributes. *Metadata* is the information that describes your information, such as, what language it is in, what color it is, or for whom you are writing it. In XML, *attributes* are a way to express your metadata. The attributes are always represented as name/value pairs. You must use a standard syntax to address your attributes in XML. For example, `color="red"` is an example of an attribute. `color` is the attribute name and the value is `red`.

When you conduct a metadata study, your best results will again come directly from your users. Setting up a series of interviews to ask users to complete the exercise described later in this section will provide you with a starting point to your metadata categories. Other metadata you may discover from search and navigation (click Results) of your web site, or from internal resources such as your support desk.

Be careful when developing your metadata scheme. If you discover that you have too many categories, you may find your information developers become discouraged and frustrated while creating information because they have to enter in so many metadata attributes. If you have too few attribute categories, you may find you don't have enough information to provide adequate search and retrieval or specialized styles. Using the process below for defining your metadata attributes will ensure that you have the best metadata to fit your user's needs.

You can use your metadata at many levels of granularity within your document. You may find that you apply some metadata at the information type level or at the content unit level. Some examples of information type and content unit metadata are discussed in previous sections. Other metadata you discover may apply only to a particular paragraph or phrase. For example, you may find you need a metadata attribute to represent a product name. This way, if your company changes the product name, you need to change it only once, and it can populate each of the phrases tagged with that particular metadata value. This is one way to use variables in your XML deliverables.

The following steps provide a guide for working with your users and your information development team to discover your metadata scheme:

1. User persona development.

Creating user personas to identify each of your users is the first phase to identify your metadata scheme. In many cases, each persona searches for the same or very similar information, using different criteria for finding information.

Set up a spreadsheet to identify each of your users to include in your metadata discovery. You may consider triggering roles that tend to search more for information on a daily basis and develop descriptions for each of these.

Role/typical persona	Brief description of role	Specialized information
End user: beginner	Looks for information about products they own and simple tasks or tutorials to begin learning the features of the product.	Looks for help systems to guide them through tasks and for training modules, videos, or instructions about using the product features.
End user: advanced	Maintains the equipment and troubleshoots problems for other beginning end users.	Looks for troubleshooting checklists or help and for additional features to add on to what they have or to provide information on how to customize their current product environment.
Sales representative	Sells and supports products and services. Interacts with prospective and current customers.	Looks for information about product pricing, competitive information, sales materials, presentations, marketing brochures, etc., and for slide presentations, product logos, and trademarks.

2. Defining scenarios.

For each of your personas, develop about 5 to 10 scenarios of how each person in each role would look for the information they are interested in.

A typical scenario may look like the following:

Zoe is a beginning user (user) of the Super Computer 6000 who is from Russia.. She realizes the computer has more bells and whistles than she knows what to do with. She needs to find out how to use her new word-processing application. She knows that the computer comes with Microsoft Word Zenon, but is not sure how to get to the application to start or which icon to click on. She is also interested in which printers are compatible with her new Super Computer so that she can print out her pictures from her recent trip.

She begins by looking through your web site for the latest Super Computer 6000 documentation. At first, all she finds is the sales information such as feature descriptions and pricing quotes about the printers that she can use.

She navigates to the support part of the web site where she finds an online user guide for the Super Computer series. She also finds a tutorial of how to get started with the Super Computer 6000 features as a result of searching on Zenon.

She would like to print out the user guide so that she can take it with her and learn more about the features.

3. Identify metadata categories.

Once you or your users create scenarios for each of your job roles, break out the noun phrases from the scenarios and assign them to categories.

The following list uses the narrative scenario in step 2 to identify the categories:

- Zoe is a beginning user (user) of the Super Computer 6000 (product name), who is from Russia (country/location/language).
- She needs to find out how to use her new word-processing application (feature).
- She knows that the computer comes with Microsoft Word Zenon (feature name).
- She is also interested in which printers (product) are with her new Super Computer
- She begins by looking through your web site (media) for the latest (date/version) Super Computer 6000 documentation.
- At first all she finds is the sales information, such as feature descriptions (information type) and pricing quotes (information type)

- She navigates to the support part of the web site where she finds an online user guide (information type)
- She also finds a tutorial (information type)
- She would like to print (media) out the user guide

4. Defining metadata names

After you have identified the categories for each of your scenarios, group the categories together and create metadata names that your information developers can use to assign the metadata during content creation. The result of your scenarios should provide you with just the metadata you need to supply information to your users. You can use the metadata discovered through this process to apply to the metadata tables describing each of your information types, content units, and elements.

The examples below are names and definitions for the metadata categories identified in step 3.

information_type

A standard information structure with standard content, including tasks, concepts, and tutorials.

product_name

The name of the product, including Super Computer 6000, Super Computer 210, Super Computer 220, and Super Computer 500.

country_location

The country to which the documentation should be delivered, including Russia, United States, France, and Brazil.

language

The language the information is in or must be translated to such as Russian, Spanish, or French-Canadian.

feature

The topic or subject of the information such as a printer or a word-processing application.

media

The method in which you deliver the information to the users.

File Structure and Navigation

Your file structure and navigation are directly developed as a result of your metadata research and schema development. Your internal users will define metadata that describes how they find information within the company and which categories they most often use to file and maintain their information. External users

will probably use a completely different set of metadata to find and understand your information. Therefore, the external users will have a different navigation structure to work through to find the tasks or reference material they need.

As you develop your metadata schema, keep in mind your file structures and navigation for your information. During your interviews and metadata research, begin to define hierarchies that make the most sense to your users. If you have a lot of metadata to work with, you may find that you have to revisit the users to ask what metadata categories they would use and in what order they would assume that they would find them. Creating a hierarchy of metadata is a way of prioritizing it for your users.

Naming Conventions

Naming conventions are extremely important to the development of this and future information models. Without naming conventions, you may find yourself forgetting what a particular information type or content unit is called. You might also find that if you name each of your files differently, you are less likely to find the file you need again easily.

There are a variety of naming conventions you can specify, including:

- Filenames
- Element names
- Metadata names and values
- Unique IDs
- Titles

Filenaming Conventions

It is helpful to name files based on the information that is in the file. This standard will allow staff to quickly find what they are looking for or referencing.

The following contains some examples of some filenaming conventions:

Valid filename	Notes
Starts with a letter.	A–Z or a–z.
Consists of only letters and numbers (alphanumeric characters).	A–Z or a–z or 0–9.

Contains no special characters.	+ or – or * or ! or & or ? are not acceptable (/ is allowed).
Has <i>no more than</i> eight (alphanumeric) characters in all.	
Company name.	May start each filename with the company name “Comstar.”
Product domain.	After the company name, the product domain must be specified (e.g., Computers).
Product name.	After the product domain, the product name must be specified (e.g., SuperComputers).
Release number.	After the product name, the release number must be specified (e.g., 1.0).
Information type	Each filename starts with what kind of information type template the information developer used to structure the content.

Defining your filenames is the process of using the rules described in the table and identifying how to maintain each name using the same rules. For example, your filenames may look like:

task_backingupfiles_supercomputerseries.xml

Each filename uses the same syntax to describe and store your files consistently for easy navigation, search, and retrieval.

Element Names

Element names require consistent naming conventions so that your information developers have more of an idea of what element to use in their markup. If you create an information model and an schema with differently structured element names—such as abbreviations, spelled-out terms, or elements that use underscores, dashes, or periods to separate the names—you may end up with a mess of different element name types making content creation harder to distinguish from one to the other.

You may also take into account that if users are not using an XML editor, it’s possible your users will have to type in each element tag individually. If the names are long, the task of typing in each element becomes tedious and overwhelming. Also, there is more opportunity for human error. Try to consider your authoring team when determining element names. Ask them what they think of when they

are writing and how they would categorize and label their information for later retrieval. You may find that there are some element names that you can represent as acronyms or abbreviations, but remember to be consistent.

You may find that many of the element names match exactly what you created in your information type, content unit, and element definitions. Using the names you identify in your information model allows for quick referral to identify how and when to use the elements provided in your information architecture.

An element naming convention may look like the following:

Valid element naming rule	Notes
Lowercase	All elements must be lowercase. Camel case and initial caps should not be used to create element names.
Full names	All elements must be the full name of the content unit or element as indicated in the information model. For example, <code><paragraph></code> should not be shortened to <code><para></code> or <code><p></code> .
Alpha start	All element names must start with a letter of the alphabet (from a—z, or A—Z). A number, or any other symbol such as an underscore, should not be used.
Underscores	Used for elements with two or more words in the content unit name.

Metadata Names and Values

Metadata needs to be understandable for your authors to apply to the attributes. If you propose using misleading metadata names, abbreviations, or other metadata names that your information developers don't understand, your possibilities for incorrect metadata values increase.

Also, if you do not provide standard metadata values for your information developers to use when assigning metadata, it is possible that they may add incorrect values that make the styling, production, search, retrieval, and other maintenance of your content frustrating. You also face the possibility of losing content in your filesystem or repository—because you can't find it again based on the metadata your information developers apply to the content.

Valid metadata names naming rule	Notes
---	--------------

Lowercase	All metadata names must be lowercase. Camel case and initial caps should not be used to create element names.
-----------	---

Metadata values are defined in the information model under each information type, content unit, or element name definition. If future metadata values must be added to the information model, please send an email to the information architecture team at infoarchitectureteam@supercomputers.com.

Unique IDs

Unique IDs are often a part of an information model. They are also usually part of the metadata attributes, but you may want to specify rules for unique IDs separately. Many times, if you use a content management system, repository, or any other kind of database maintenance system, the system may assign IDs automatically for you. If you do use this feature, ask your system vendor to document how he generates the IDs. You can include those facts in your information model.

If you assign your own IDs, you may want to consider using a shortened version of the filename, content unit, or element name, and the purpose or subject of the content you are assigning the ID to. For example, consider a warning in a user guide you want to use in many different places for which you must refer to the unique ID to be able to refer to the guide's content. Your unique ID for the warning may be the value `generalreuse_warning_unplugcomputer`. This ID is long, but it is clear as to what content exactly you intend to reference.

Another example applies if you assign a unique ID to each piece of information you create using your information types. For this scenario, you may want to indicate the information type you used and the subject of the information you are writing about, such as `task_backingupcomputer`. This ID helps authors locate information they can use later on more easily.

The following example provides guidelines for creating unique IDs:

Valid ID naming rule	Notes
Lowercase	All unique IDs must be lowercase. Camel case and initial caps should not be used to create element names.
Information type, content unit, or element name	Each unique ID should indicate which information type, content unit, or element name that you are applying the attribute to, such as task, concept, warning, paragraph, etc.
Subject	After the information type, content unit, or element name, add an underscore (_) followed by the subject matter. The subject can be a phrase composed of two to three short words describing the content to which you are applying this ID.

Titles

Creating naming conventions for your titles is more of an authoring guideline rather than a technical one. Most information types include some sort of title as part of their structure. The titles can define the flow of your content and are incredibly important to your users for searching and finding the information they need quickly. If you give all of your concept pieces the title “Introduction,” then your users will not know what it is an introduction to and will have to read the content to determine whether they need to keep looking or if this concept piece is what they need. This kind of title creates a frustrating user experience because they have to read about parts of your product they may already know of and/or do not care about.

Creating rules for titling your content may look like some of the following:

Valid title rules	Notes
Task Information Type	Tasks must start with a gerund to indicate some sort of action the user will be doing during the task. An example title is “Backing up your computer.”
Policy Information Type	Policies must always contain the word “Policy” in the title.

Concept Information
Type

Concepts begin with the word “About” and then include a noun phrase referring to a feature or function that you are going to describe in the concept content.

Conducting a Small Pilot Project

The purpose of the pilot project is to implement the knowledge gained through your user studies and development of your reuse scheme and information model. This is your time to develop and test all of the research you have done. It’s best to set some metrics to measure during your pilot project.

Pilot projects sometimes seem like a daunting task. The intention of a pilot project is to not make sure you run through everything successfully, but to act as more of a test run to see what pieces of your model you missed or described incorrectly. The whole purpose of planning from the beginning of your project is to start with what you know from your users and with the tacit knowledge you have about creating information. As you work through the project, you will find that some of the assumptions you made during your planning are incorrect, and a new iteration of your information model needs to be developed.

It is important to pick a project that is significant so that you have some way to prove your model works, but that also doesn’t necessarily need to be held to a tight deadline. You will find that because your information model is not perfect the first time, you will have to allow for flex time to discover the errors and retest them in your pilot project.

Your pilot projects should test different parts of your plan, but not necessarily everything at once. Perhaps, for example, the first part of the pilot project tests your information developers (one or two of them) on how to create the information types you defined. They will not use all the elements defined in your information model at first, but they will probably find some they need but that haven’t been defined. The second part of the pilot project will test the metadata you have defined. Your project will identify the changes that need to be made to your schema and to your process so that you can refine your development needs. By keeping track of the metrics that you put in place for your pilot project, you will be able to use that as a basis for future projects to insure that they run smoothly.

Putting It All Together

By creating the information model first, your efficiency increases in the development of your schema. You are not approaching development with a build and fix method, but instead are using a much more iterative and logical approach to your programming. Your information model is to be used as a guide to develop your schema and it also can in turn be used as a guide and tutorial for those responsible for using the schema to develop the information.

Once you have finished the development of your schema and gone through an initial run through of it, you will most likely need to make some changes to either your schema or the process for developing the information to provide to your users. However, these changes are going to be minimal because of the thorough planning phase.

You may need to go through iterations of the information model as times change and new users come through the door. But, you can be positive that they are all following consistent methods for developing and delivering information to the users because they are all using the same information model as their guide. This consistency provides immeasurable benefits to you and your company over time.

References

Contributions by Comtech Services, Inc. (<http://www.comtech-serv.com>).

DITA Language Specification (<http://docs.oasis-open.org/dita/v1.0/dita-v1.0-spec-os-LanguageSpecification.pdf>).