Web Content Management with DITA and Documentum

Jaimala Bondre

EMC Proven Professional Knowledge Sharing 2010
Table of Contents

1 Introduction ................................................................................................................................. 3
2 An Object Oriented Approach .................................................................................................... 3
3 DITA – A structured approach ................................................................................................. 4
   3.1 A Brief History of DITA ...................................................................................................... 5
   3.2 DITA – A Paradigm Shift In Documentation .................................................................... 5
   3.3 DITA – Topic-based Authoring ....................................................................................... 6
   3.4 DITA Maps – The Assembly Information ........................................................................ 7
4 The Problem Context ................................................................................................................ 8
5 The Solution ............................................................................................................................. 9
   5.1 Step 1: Categorization of business data and identification of the content hierarchy .... 9
   5.2 Step 2: Identifying the metadata that needs to be captured for each of these topics .. 11
   5.3 Step 3: Building templates for different topics ............................................................... 11
   5.4 Step 4: Using authoring tool to create content based on the template defined .......... 12
   5.5 Step 5: Creating XML applications within the repository for chunking data .......... 13
   5.6 Step 6: Initiating the workflow for content review ....................................................... 14
   5.7 Step 7: Creation of DITA maps ..................................................................................... 15
   5.8 Step 8: Publishing the content through the DITA maps .............................................. 15
6 Conclusion .................................................................................................................................. 15
7 References.................................................................................................................................. 16

Disclaimer: The views, processes or methodologies published in this compilation are those of the authors. They do not necessarily reflect EMC Corporation’s views, processes, or methodologies.
1 Introduction
The Web has become one of the main channels of communication for most organizations. In fact, the content itself that is being published on the website has become more structured and has made it easier for both publishing and consumption. To cope with the complexity of content management most organizations use some form of a Web Content Management (WCM) system. Wikipedia defines a WCM system as “content management system software, implemented as a Web application, for creating and managing HTML content.” It is used to manage and control a large, dynamic collection of Web material. A WCM system facilitates content creation, content control, editing, and essential Web maintenance functions.”

The use of a WCM system allows organizations to publish huge and diverse data to the Web. However, publishing is just one part of the WCM process; it also needs to co-exist with an effective and easy content authoring process. Often, content in large organizations grows at a rapid pace, creating a need for a complete authoring and publishing solution in a structured fashion. DITA (Darwin Information Typing Architecture) takes a very structured approach towards content authoring, enabling content to be produced in a standard way for different publishing formats. This paper will attempt to combine the Documentum® content management workflow and publishing capabilities with the universally accepted DITA for content authoring. It will attempt to present a complete end to end structured and effective publishing solution – one where unstructured data is transformed to structured XML data, which is later passed through a workflow for review, and is finally published as per the needs of the target audience. Throughout this entire process the DITA structures are retained within Documentum to enable reuse, when required.

2 An Object Oriented Approach
As with most Web applications, WCM systems also adopt an Object Oriented approach to application architecture. In fact, the concepts and principles of Object Orientation can be applied to almost all kinds of applications. Such an approach allows each object to perform the task or activity that it is best meant for and also gives structure to the application. These same principles can also be applied for content publishing such as authoring books and printed material, publishing web content, or creating digital content. Taking an Object Oriented approach while structuring the content makes it easy to build, modify, and reuse the content.
To elaborate, take the example of a consumer durable company that is manufacturing and selling washing machines along with other products. This company would need to create user manuals for each of the individual models that it sells. The company can use an object oriented approach for content creation and publishing of these user manuals. This allows it to publish content to its website, create a printed copy and also a digital copy (i.e. a pdf format) in a structured manner rather than building each user manual from scratch. This results in greater efficiency and better time to delivery.

In the above example each user manual can have a set of sections, such as the installation section, the control panel section, a section on special features, etc. While designing the document structure for such a manual, one can have separate templates for each of these sections or can have templates at even more granular levels. Further, this same template can then be used across multiple models or one can choose to have different templates for different washing machine categories, for instance, a different template for fully automatic and semi-automatic machines. The choice of the appropriate number and levels of templates depends on the way an organization wants to structure its content. Depending on the kind of information that is being stored, to the way this information is being accessed and consumed, each organization needs to adopt the right kind of content structure. If there are too many templates, then each of the individual pieces of information would end up being too small, resulting in greater complexity of the content structure. On the other hand, if there are too few templates, the content structure might be simpler, but become difficult to reuse content as it is not granular enough. Thus, there is a trade-off between simplicity and reusability. One needs to be aware of the optimal solution that will suit the organization’s content needs.

3 DITA – A structured approach

Now that we understand the importance of structuring our content and how it can be reused, let us see how we can define an end to end solution that can create this structure and more effectively present this same information differently for different target audience. This is where Darwin Information Typing Architecture (DITA) comes into the picture to let one author content into concrete structures. DITA is an XML-based architecture that has been defined for creation and delivery of content.
3.1 A Brief History of DITA

At the start of the new millennium, there was a huge amount of interest in XML as a data exchange method. However, there were not many who were using it as a document markup language. DITA was introduced by IBM in March 2001 as a simplified version of XML for documentation, replacing an internal version of SGML (a fairly old and complex markup language for documents) that was being used by them. IBM’s introduction of DITA brought the key ideas of XML to documentation and enabled developers to use a much simpler XML starter set for documentation markup. This design allowed reuse of small content components, helped formalize information typing practices, and offered a standards-based information exchange. Such was the effectiveness and popularity of DITA that by April 2004, a Technical Committee set up by the Organization for the Advancement of Structured Information Standards (OASIS) – a global consortium that drives the development, convergence and adoption of e-business and web services standards – initiated the process for exploring a DITA standard that could be approved by them. By June 2005, DITA 1.0 was approved as an OASIS standard. Today, DITA’s topic-based authoring is viewed as a best practice for technical documentation.

3.2 DITA – A Paradigm Shift In Documentation

DITA is defined by a set of Document type Definitions (DTDs) and XML Schema maintained by OASIS. However, DITA is much more than just a schema – it represents a completely new paradigm for content creation and delivery. In the traditional approach to document creation, a document would be created sequentially, i.e., from beginning to end. This approach was both intuitive and congruent with the technology available during the early years of digitization. However, the introduction of XML to documentation through DITA principles enabled authors to create as well as publish documents in a modular, non-sequential form and yet bring a very structured approach to content creation and publishing.

DITA document components (called ‘topics’ in DITA terminology) are created individually and in a modular manner. When needed, the final document can be assembled using a mechanism called DITA maps. In fact, the final document could be altered for target audiences by using a different sequence – one appropriate for that particular audience. Going back to our washing machine example, we could have two different sections of the user manual which are created by different authors – the ‘technical specification section’ containing information the power source, the working conditions, etc. might be authored by a technician, while the ‘functional section’ that explains the various buttons on the panel and their functions might be authored by a functional
expert. While these two sections could be assembled together in a particular sequence when the user manual is being published, they could be assembled in a totally different manner when this information is being presented on the company website for customer support and troubleshooting purposes for example. The technical specification might be included in the set of problems and solutions for technical issues while the functional section might be included in the set of problems and solutions related to the functional usage. This flexibility in creating and publishing content is a very powerful and useful feature and is something that is offered through a DITA approach to content architecture.

3.3 **DITA – Topic-based Authoring**

The concept of a topic is core to the DITA approach, and its architecture is based on topics. In fact, one of the distinctive features of DITA is the topic-centric nature of its architecture. Topics are individual pieces of content that form small, standalone chunks. Each topic is a discrete piece of information that does not need a separate context for it to make sense. DITA defines topic as the smallest unit of information and these end up being the building blocks for the final document created.

DITA defines three basic topic types: Task, Concept, and Reference (see Figure 1.0). Each of these topic types are specializations of the generic Topic type and inherit a title element, a prolog element for metadata, and a body element. The body element is similar to HTML in structure and contains the paragraph, table, and list elements.

![Figure 1.0](image)

**Task**: A Task topic contains information on how to perform a particular task, e.g. how to set up a particular wash cycle through the washing machine control panel.
**Concept:** A Concept topic contains key conceptual information relevant to a task and gives an overview of the task. This kind of information is more objective in nature and might contain definitions and guidelines, e.g., the safety guidelines outlined before starting a washing machine.

**Reference:** A Reference topic provides detailed facts related to a task. It typically contains information such as programming instructions or factual material and is often presented in table or list form to give users quick access to information, e.g., the different settings that correspond to different wash cycles.

As explained earlier, these topics act as the building blocks for each document and during the creation of a document these are assembled in a particular manner. However, to assemble the document one needs to understand the sequence and the hierarchy of these topics. This information is present in DITA maps.

### 3.4 DITA Maps – The Assembly Information

DITA maps can be considered as a table of contents for a document. It gives information about the sequence in which the topics need to be assembled for a particular target audience. Apart from information on the assembly sequence, DITA maps may also contain relationship tables that give information regarding which topics are linked to each other. Thus, during the creation of a document using DITA concepts there are two steps involved as opposed to the creation of a single, monolithic document in the traditional approach. These steps are: (1) Creation of different topics by the authors and (2) Defining the sequence and hierarchy of these topics for different target audiences. This means that authors now need to think in a more modular form and need to be more aware of the relationships between topics. Thus, it becomes very important to first do an analysis of the content even before authoring topics so that one can understand the underlying structure of the content. Once this is done, one can identify the different types and levels of content before finally charting out their hierarchy and relationships – the first part of this information goes into the identification of the topics and the second part goes into the creation of DITA maps. Figure 2.0 shows how DITA maps can be used during the publishing process.
Right away we can see that following this kind of a structured approach provides an organization with a far more flexible publishing solution – one that enables greater reuse of content and minimizes redundant effort. For example, if a document has two versions for different target audiences with very little difference – e.g. a washing machine user manual for two countries which have different power socket designs (this might mean that there might be a slight change in the technical specifications part) – then authors can bypass the tedious task of publishing two different documents. Instead they could have different DITA maps defined for each of these documents and the relevant DITA map could be used to publish the required document. Similarly, for multi-channel publishing, i.e., publishing the same information to multiple channels with slight modifications, DITA maps provide a very powerful yet flexible mechanism to ease the publishing task. An example of multi-channel publishing is information that is published in a digital document (.pdf or .doc files) or published on the corporate website or on the technical support section. Each of these deliverables can have a different DITA map that identifies the sequence and hierarchy of topics and this information can be used to assemble topics to create the final output.

4 The Problem Context

Now that we have covered some of the basics of DITA, let us define the problem that we are attempting to solve. The consumer durable company mentioned earlier needs to define the
content structure that would help them create user manuals for all of their different models. The primary objective is to come up with a content architecture that allows the organization to reuse existing content and provides the flexibility to publish this information for different target audience in a customized manner.

## 5 The Solution

In this section we will demonstrate how we will be using the principles of DITA and the functionality of Documentum to translate unstructured data into a more structured XML form and provide a comprehensive publishing solution that will serve the objectives of the consumer durable company. The entire solution is detailed in eight steps.

### 5.1 Step 1: Categorization of business data and identification of the content hierarchy

Identifying how the data in the organization should be structured is of prime importance. One needs to decide how granular each information chunk should be since this will form the basis of content reuse. If the granularity of data is too high then maintainability becomes an issue. Similarly, if information chunks are too large, no two users will be able to reuse that piece of information.

To transform the organization’s legacy information to a DITA-based terminology, one needs to identify the non-homogeneous data and put it up as a separate category or topic, as we call it. Conceptual-based information should be segregated from task-based or reference information. Besides conventional DITA topics, one may also choose to have specialized topics as per the business needs. DITA offers complete flexibility and is fully extensible.

In our example, let us work with only the two identified sections for the manual; the ‘installation section’ and the ‘washing section’. Each of these pieces of information should be stored in the repository so that they can be reused in the future. Let us define the object types for these two sections. The various object types are shown in Figure 3.0.

The object types created are exact matches to the topic categories as defined in DITA, i.e., concept_type, task_type, and reference_type. For each of these base types there are two types; one for washing, the other for installation. These types inherit from their parent type. Thus, attributes common across concepts, tasks, or references can be kept in the base type, where, as specialized attributes, they can be kept in the child classes. In our example, the concept and
the task types have common attributes —‘description’ and ‘title’ — which are inherited by their children. These fields are mandatory and must be entered by the author while creating concept or task content. Similarly, prerequisite_id[] is a repeating attribute which is mandatory for an installation task. This attribute indicates references to prerequisites which need to be done before the installation task. The prerequisite_id attribute could be a subset of keys to prerequisite values of a registered table.

![Diagram of concept_type, task_type, and reference_type](image)

**Figure 3.0**

Now let us define a folder structure in the repository where all the individual content created by the author will be stored. This structure is detailed in Figure 4.

![Folder structure](image)

**Figure 4.0**
5.2 Step 2: Identifying the metadata that needs to be captured for each of these topics

Once we have identified the various object types – which enable us to carry forward the DITA hierarchy into the CMS repository – we need to have a mechanism to uniquely identify these topics. Once these topics have a unique identifier and sufficient metadata associated with them, it becomes easy to search them within the repository and enable reuse.

Good naming conventions for each topic will ensure maintainability and effective reuse. Once the volume of documents and topics in the repository increase, it can become increasingly difficult for the authors to search and reuse the topics if the naming convention is not intuitive. This may lead to redundancy and duplication of topics and can defeat the purpose of using DITA.

Whenever an author creates a topic, metadata for these topics must also be entered. In our example, every concept and task has a mandatory field, i.e. ‘description’ and ‘title’. Also, beside these two custom attributes every topic when checked into the Documentum repository will be saved with an unique r_object_id and r_version_label. Also, various custom attributes can be made mandatory by using templates for creation of topics.

5.3 Step 3: Building templates for different topics

It is important to associate correct and useful metadata with the topics in order to make them retrievable. Thus, it is mandatory for content authors to enter this information correctly. Templates can be created in the repository that enforces filling mandatory metadata associated with any topic. Templates are saved as XML files and also serve as placeholders for data that must be entered for that topic. In our example, content authors who choose to author an installation task are prompted to fill prerequisite_ids. Moreover, when the author saves the topic or checks it into the repository, certain business validations can also be configured besides syntactical validations which are provided by the authoring tool. Thus, users that do not enter a prerequisite_id for an installation task will not be able to save that topic. This ensures that the necessary metadata for the topic is captured by the author while creating it.
5.4 Step 4: Using authoring tool to create content based on the template defined

Authoring tools enable authors and content writers to concentrate on creating content rather than being involved with the technicalities of how this content needs to be published or printed. Content authors with different skill sets can write about their subject topics and leave assembly of the various topics to the technical person. DITA-based authoring tools help content creators use DITA with minimal knowledge of DITA technicalities. Thus, content authors simply need to be trained to create topics using the right template and should be able to save content with the appropriate metadata. The process of naming various topics should be streamlined to have a standard way for referencing topics.

Authoring tools can also be integrated with a Content Management system. In cases where Documentum is used, Xmetal authoring tool by Just Systems and QuarkXML are two commonly used authoring tools that integrate with Documentum.

Authoring tools integrated with the repository give authors greater flexibility in performing their tasks. Authors can save metadata for the content they prepare, check-in or check-out the content, and create versions from within the authoring tool. Also, most of these tools provide drag-and-drop functionality to add the different XML tags or references to other topics without really having to know XML or DITA syntax. Additionally, validation features ensure that the content created within these tools is free from syntax errors. Some tools also provide for adding various business related validations at various integration points between the repository and the tool. Search for topics from within the repository is also a useful feature which lets authors search for topics and add references from within the repository. One must seek most of the above features while selecting a suitable authoring tool. Content can also be created from within a Webtop application and appropriate metadata can be associated with it. For instance, users can create a concept task directly in Webtop and set the appropriate metadata for it. In our example, assume the author creates a washing_reference topic in which he/she adds a reference to an image. Also, the author creates a washing_task which contains a content reference.

All this is made easy with the help of authoring tools. The author need not know the syntax for <conref>. Simply select the appropriate topic, content, or image from the repository, place it within the topic being created, and then check-in the document.
5.5 Step 5: Creating XML applications within the repository for chunking data

XML applications in Documentum can define how the imported or checked-in XML files can be chunked into smaller units of information. A repository can have various XML applications to chunk the XML files in different ways. These applications can also deal with different XML structured files. Chunking can be triggered depending on the element in the XML file, or on its parent or on any of its attributes. Each XML application has its own configuration file which is core to the entire chunking process. This file conforms to a particular DTD or schema and contains various XML tags which define the rules for the XML document.

The <app-pattern> tag determines which XML application should be applied to the checked-in XML file. Similarly, the <dds-validation> and <validation> tags can enforce validations on the checked-in XML file. The <map_rules>, <xml_content_rule> and <link_rule> tags determine how the XML file will be chunked. With the help of the configuration information within the file, each chunked object can be stored as a separate object of a particular object type or it can be stored at a particular repository location. Apart from this, the chunked object can also be attached to a business lifecycle. Also, various attributes such as permissions, owners, etc. can be set.

![Figure 5.0]
In the washing machine example, washing_reference topic contains image references. Similarly, the washing_task topic contains a content reference to another topic or a content document. The XML application can be configured in Documentum to create the washing_reference object with appropriate metadata (from the topic XML) when the topic file is checked in to the repository. Also, we can configure that the washing_reference topic will be stored under the path Cabinets/washing/reference. The chunked data of washing_reference will be stored as a virtual document with the image file as its child. Thus, any change to this image file in the repository will be reflected in all the topics where the reference to this image has been made. Now the folder structure that was shown in Figure 4.0 would look like the structure shown in Figure 5.0.

Let us also consider that the XML application associates a Lifecycle to the checked-in document. The initial Lifecycle state is ‘draft’. Once the author is satisfied with the content, he/she promotes it to the review state for approval. From here, the workflow is initiated for approval of the created content (see Figure 6.0).

![LIFECYCLE STATES FOR CONTENT](image)

**Figure 6.0**

## 5.6 Step 6: Initiating the workflow for content review

Each topic content that is created must be approved by the approving authority. Thus, if the washing_reference task is completed by the author, he will push that content to the ‘review’ state. ‘Review’ state indicates that the document is ready for review or is currently being reviewed.

From this state, a workflow for content approval is initiated. Renditions may be created for each topic using the XTS transformation services as part of the workflow. If the organization deals in multi-lingual content availability, translations may also be applied on these topics to generate the same topic in multiple languages. In the workflow, the topic may be rejected and the author may need to re-initiate the review process after making the required modifications. Once all the
topic contents for a particular manual are approved, a DITA map can be used for assembling the topic contents.

5.7 Step 7: Creation of DITA maps
Various DITA maps may be created for different audiences. This is done by altering the sequence or hierarchy of the topics as defined in the DITA maps. DITA maps are created by the inclusion of topic references <topicref>. Thus, manuals for different washing machine models will be created by including different topicrefs to other topics. In fact, we can have two sets of DITA maps, one for a print copy of the manual which can be more concise, and one for the Web. Thus, the DITA map for printing a copy of the manual might exclude the installation_reference topic altogether, while the Web version might have references to most of the other models that are currently being sold. Thus, without much effort the same information is assembled in different ways for different audiences.

5.8 Step 8: Publishing the content through the DITA maps
The DITA map created for the Web can be again passed through a workflow for publishing approval. Once it runs through the workflow cycle and is approved as a whole it can be pushed to the appropriate publishing folder. The Site Caching services based on publishing configurations will push the content to the target server and the content will be published.

6 Conclusion
The above illustration is just one simplified approach to walk one through all the steps of an end-to-end content management solution using DITA framework along with Documentum. While this solution may not be the best or the only solution, it does help to illustrate the various steps to follow while using DITA and Documentum for WCM. Any organization that wants to use an XML-based approach to its content management needs — especially one where the same content needs to be reused but with slight changes for different target audiences — can follow the steps outlined above as a guideline.
7 References

http://www.comtech-serv.com/dita.shtml
http://www.adobe.com/devnet/framemaker/articles/intro_dita_fm8_04.html
http://sourceforge.net/projects/dita-ot/
http://mylearn.emc.com