Managing content for information centers and large repositories using the DITA framework

Alice S. Etim
Winston Salem State University

ABSTRACT

DITA is an acronym for Darwin Information Typing Architecture. It is an information architecture and standard specification for representing different information types. There are several flavors of DITA in the market today and each industry can customize the open source DITA to suit the specific needs of that industry. The Open DITA specification is one of the releases of the OASIS TC (Organization for the Advancement of Structural Information Standards Technical Committee) for DITA that serves as the approved standard for organizations to use in creating, representing, discovering and managing their product information, digital repositories and information centers. Although developers and providers of large information systems in many organizations have integrated DITA and its associated metadata schema into development, discovery, management and use of product information, very little has been written about DITA in academic literature. This paper is written primarily to integrate DITA into academic and business literature as an important new information architecture and standard for creating, representing, discovering and managing large digital information types. The paper examines the general framework or open source architecture of DITA that is often referred to as the *OASIS DITA*. The organization of the content begins with a definition of DITA, followed by a background or origin of the DITA framework. The core metadata elements are identified and discussed with appropriate examples. The paper concludes with a discussion on the benefits of using DITA by business entities.

Keywords: Information architecture, DITA framework, content encoding, document delivery standards, XML-based standards and schemas.

INTRODUCTION

The author is in the position to write about the open source DITA or the OASIS DITA because of her research while at IBM Software Group and her participation in OASIS TC for a new technology project that led to the development of the first DITA specification, Specification 1.1. The OASIS DITA TC is made up of representatives from different firms and industries that have adopted DITA. Some of the firms include: Arbortext, BMC Software, Comet Communication, Comtech Services, Inc., Intel Corporation, IBM, Nokia and Sun Microsystems (OASIS DITA TC, 2007).

The Darwin Information Typing Architecture or DITA is an eXtended Markup Language (XML) based standard for information architecture. It has a rich metadata schema that supports modular content creation, delivery, publishing and discovery. Day et al (2001:1), in a document that serves as the roadmap for DITA, defines DITA as an XML-based, end-to-end architecture for authoring, producing, and delivering technical information. This architecture consists of a set of design principles for creating "information-typed" modules at a topic level and for using that content in delivery modes such as online help and product support portals on the Web.

The definition highlights several things including one important attribute of DITA and that is the use of a syntax/language for content encoding. Elements or properties of DITA are closely related to the ones used by existing information systems' metadata schemes such as the DUBLIN CORE (DC), Metadata Object Description Schema (MODS) and Metadata Encoding and Transmission Standard (METS). The structure and elements of DITA is discussed in a later section of this paper.

EVOLUTION OF DITA

DITA is the brain child of several IBM Corporation's engineers and information systems experts. At the forefront of the DITA effort in the late 1990s and early 2000s were: Don Day, Michael Priestley, Gretchen Hargis and David Schell (Day, Priestley and Schell, 2001). DITA was discovered to address an important information problem at IBM in the 1990s: the need for an effective means of collaborative creation of topic-based, modular technical and information guides for the myriad of IBM products as well as the representation of the information in digital repositories, portals or Web-based information centers to allow for ease of discovery, retrieval, search, update, republish and reuse.

Prior to DITA, many businesses and organizations, including IBM, delivered hard-bound documentation and shipped them to their clients. Employees of IBM for example, were able to access these resources in the individual shelves or physical libraries that were often at centralized locations within the company's physical facilities. The bound resources were often shipped in large boxes with product CDs by regular mail to clients who used them to read concepts, readmes, or install and configure products as well as perform a myriad of other tasks. The reference resources and handbooks were exceptional large and could span five volumes per shipment to a client. The shift came when in the 1990s the bound resources were replaced with the CD version. This practice of shipping product information either as bound copies or in CDs with the actual products is still being done today but in limited cases such as shipping products to customers in less developed countries where technology is minimally used. In this digital era, the common approach is that clients have access to product information in digital repositories, online information centers or portals.

The road to the creation and delivery of product information in digital formats can be traced back to several initiatives and one of such was an important advancement by way of the IBM *DocBook*. The DocBook was primarily designed to allow a single, continuous technical narrative about an article, book or multiple-volume resource (Day, Priestley & Hargis, 2005). Through code transforms, DocBook provided a means to chunk the technical narrative into topics that could be viewed via a web page using earlier browsers like Netscape. DocBook provided a document type definition (DTD) and the goal of the DocBook DTD was to handle all standard requirements for creating technical documentation electronically. The usage model for DocBook as well as other digital information architecture tools of that era, however, was difficult to grasp. Although DocBook encouraged some form of customization, it discouraged local extensions due to the potential for unknown new elements that can break the tool support and interoperability (Day, Priestley & Hargis, 2005).

Another disadvantage in the use of continuous technical narrated document portal was with task-based activities. Usage was very error-prone and Rich (2009) observes that the use of non-task based information over the years has led to a usability crisis because of task complexity and inconsistencies of intuitively deducing what next activity or step to perform. The usability crisis is costly for businesses and the global economic system. Rich (2009) cites a study in the Netherlands in consumer electronics. In the study, half of all reported malfunctioning of consumer electronics and related products that were returned to manufacturers and retailers were working well. The problem was that consumers could not figure out how to operate the devices or the product information was not very helpful. The DITA framework is now the prescribed solution for this type of problem as well as other problems that businesses have for creating, transforming, and discovering product information. DITA can help to overcome some of the product information issues because it allows information to be created and organized in topical and modular format that are further grouped into types – *concepts, tasks* and *reference* information.

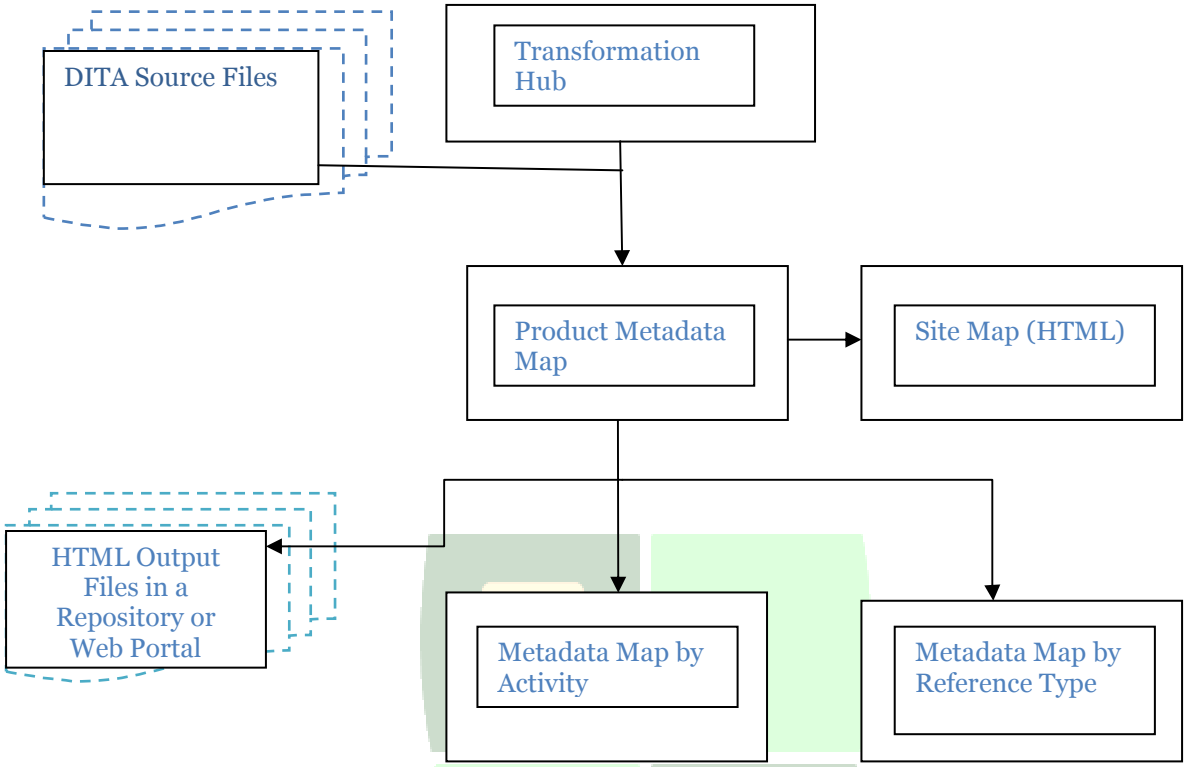
THE DITA FRAMEWORK

DITA unifies and integrates several features into a framework. Figure 1 shows a simplified DITA framework with the following features:

- DITA source files
- Transformation hub (driver file, processor such as Xalan XSL transformer)
- Product metadata map
- Site map
- HTML articles as output files
- Metadata map by activity
- Metadata map by reference type

The DITA Open Toolkit, an open source implementation of the OASIS DITA TC specification for DITA DTDs and Schemas, has a transformation hub (Xalan or related parser). The Toolkit transforms DITA content (maps and topics) into deliverable formats such as HTML and PDF formats.

Figure 1: The DITA Framework



Content Providers

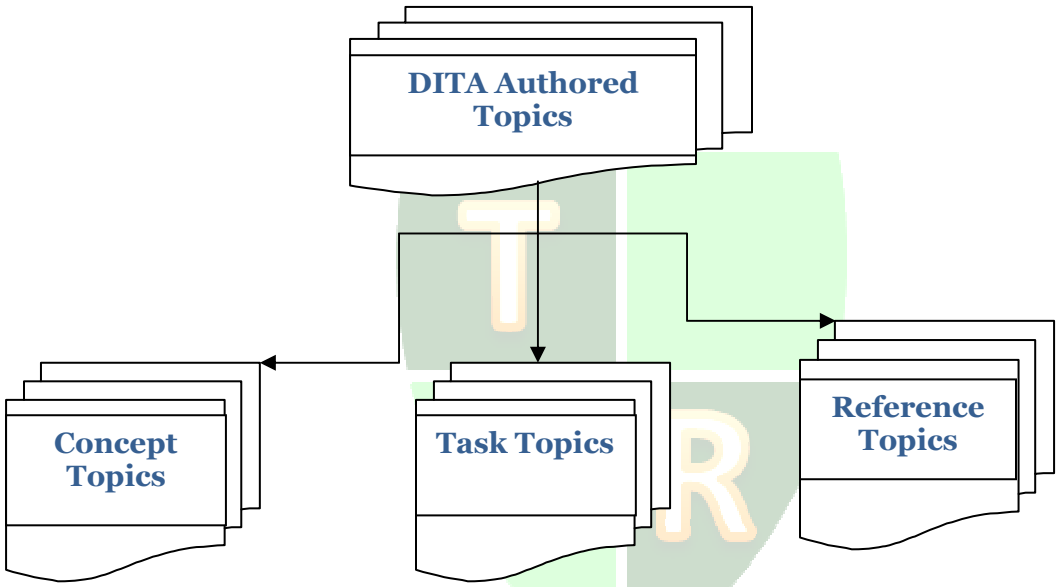
The DITA source files have the technical contents that are developed by content providers - different individuals such as information architects, software developers, engineers and technical writers. The term "content providers" is used loosely to describe generic authors of contents for the web portal or digital repository. In a project team environment for example, anyone that share in the task of helping to produce the product can generate content. One of the main advantages of DITA is that it allows for collaboration among content providers across an organization regardless of their geographical location.

The process for generating metadata in DITA framework is similar to what is already established about metadata generation in the literature – manual, automated or both (Greenberg, 2003; NISO, 2004). As pointed out by Greenberg (2003), there are several factors that influence the quality of the generated metadata – the type of objects such as an online article, environment that the metadata is hosted, who is the creator of the metadata, availability of financial or human resources and the complexity or intellectual requirements for the underlying metadata scheme. In the case of DITA framework, metadata quality is guarded by information architects or IS (information systems) professionals who oversee the finished articles with the generated metadata. These professionals provide to content providers necessary guidelines, authoring templates/tips, training sessions, editing and reviews. Content providers also depend on the information architects to provide them with tools, templates and overall guidance for developing content development using XML encoding and syntax.

The DITA Source Files

The DITA source files are categorized into three document types: concept, task and reference types. The DITA source files contain the *metadata elements* and as such I will devote most of my discussion to this feature in the DITA framework. The categorization of content is what makes DITA well accepted in the industry because it is different from the continuous narrative content like those generated from DocBook. Contents are easy to create and specialization is possible. Task-based contents can easily be separated from non-task content. An article that is authored cannot be generic; it has to fit into one of the three categories: concept, task and reference as shown in Figure 2.

Figure 2: Categorization of DITA Topics



The first task is for content providers to agree on the topics to be written and then draft an outline for the topics. Once that is done, the information architect can help to group the outline into concept, task and reference topics. Each content provider uses appropriate templates and tools to author the content.

The information architect or the IS professional is also responsible for ensuring that templates are developed to match the document types as described in the DITA specification. Content providers can follow guidelines that are laid down in the DITA specification. The common and recommended best practice is to provide in-house developed templates to content providers as well as XML tools rather than ask content providers to code using the DITA specification directly. This effort helps to ensure high metadata quality. The DITA specification has the structure, guidelines and information for all rendering of DITA and some of the information includes the following (OASIS DITA TC, 2007):

- A <bookmap> specialization for encoding book-specific information in a DITA map
- A <glossentry> specialization for glossary entries
- Indexing specializations for page ranges, and sort order
- Graphic scaling capability

- Short description with flexibility through a new <abstract> element
- Specialization support for all attributes including global attributes, such as conditional processing attributes
- Support for integration of existing content structures through the <foreign> element
- Support for new kinds of information and structures through the <data> and <unknown> elements
- Formalization of conditional processing profiles

Figure 3 is an example of a concept topic with embedded metadata elements. The XML encoding was done using an Arbortext editor as shown in the comment line. The metadata elements are in the angle brackets "<>". The metadata elements in the top section (header) are descriptive in nature and they include concept id, language, title, titlealts, navtitle, author and short description. The required elements in the header for a concept topic are "<concept id>", "<title>" and "<shortdesc>" for concept Id, title and the short description for the concept document. If any of the required elements for this concept topic is left out, the processing of the file by the transformation hub will generate errors. The header section is followed by the prolog section. The "<prolog>" tag groups all the administrative, technical (example, product information) and rights management metadata. There are many metadata elements that one can add in the prolog. Some of the elements that are embedded within the prolog metadata that are shown in figure 3 are listed again here to help in the explanation. Elements such as product information <prodinfo> can be nested. The list include: copy right, copy year, copy holder, audience, product category, product name, version management list, component, and platform.

```
 <copyright><copyryear year="2011"/><copyrholder>XYZ Corporation</copyrholder>
</copyright>
<audience>Internet users </audience>
<category>Add at least one product category with supported values</category>
<prodinfo><prodname>XYZ Web Server</prodname>
<vrmlist><vrm version="v2.0"/></vrmlist>
<component>a1 a2 a3 a4</component>
<platform>windows linux</platform>
</prodinfo>
```

The concept body section is called "<conbody>". For a task topic, it would be "<taskbody>". The body section is used by content providers to write their specialized content for publishing to the digital repository. The body content can be arranged neatly into paragraphs. Both basic and advanced formatting (boldfacing, listing, linking, etc.) can be introduced within the body section by authors of content. Most of the header and prolog metadata would have been written into the template that the information architect or professional delivers to the content providers to use in authoring content.

The last section that is shown in figure 3 is the related links "<related-links>" section. This section allows authors of content to link to articles, books, technical papers, user guides, learning objects and other web resources including documents within and outside an organization's intranet. Related linking capability in DITA makes it to be favorably received by Web users and enthusiasts. Jovanovic et al (2005), for example, uses DITA as one of the information standards in defining ontology learning objects for the Web. DITA has

generalization, specialization and inheritance rules that define how new information objects should be created and linked to the existing ones, like the “parent” or “friend” information roles.

Figure 3: Illustrating the DITA metadata elements for Concept Document Type

```
<?xml version="1.0" encoding="utf-8"?>
<!--Arbortext, Inc., 2008-20011, v.4002-->
<!DOCTYPE concept PUBLIC "-//DTD DITA Concept//EN"
 "../dtd/concept.dtd">
<concept id="concept_topic" xml:lang="en-us">
<title>Template for new concept</title>
<titlealts>
<navtitle>Sample concept</navtitle>
</titlealts>
<author>Alice Etim</author>
<shortdesc>One- to three-sentence description will appear as the first paragraph of the finished
article.</shortdesc>
<prolog>
<copyright><copyryear year="2011"/><copyrholder>XYZ Corporation</copyrholder>
</copyright>
<audience>Internet users<audience/>
<category>Add at least one product component with supported values</category>
<prodinfo><prodname>XYZ Web Server</prodname>
<vrmlist><vrm version="v2.0"/></vrmlist>
<component>a1 a2 a3 a4</component>
<platform>windows linux</platform>
</prodinfo>
</metadata></prolog>
<conbody>
<p>The concept body is used to write about the concept one has in mind. Say for example, if I
wanted to write about Metadata, I will discuss the Metadata concept here.</p>
</conbody>
<related-links>
<link href="metadata1.dita" otherprops="version_goes_here" role="friend" type="reference">
</link>
<link format="html" href="http://www.samplemetadata.com/index.html" role="external">
<linktext>Sample metadata site</linktext></link>
</related-links>
</concept>
```

It is known fact that there is no single international standard for all metadata types ((Burnett, 1999 and Greenberg, 2003). The effort in showing the syntax and displaying the DITA elements in Figure 3 is to help explain DITA to those who are familiar with other metadata schemas or objects.

BENEFITS OF USING DITA

The benefits of DITA are discussed in this final and concluding section. The first benefit is that DITA uses XML syntax. With the rapid evolution of the Internet and Web-based technologies, XML and other tooling that DITA uses are quickly gaining popularity and they support IS (information systems) professionals in their jobs. In this paper, the DITA framework was introduced with examples that use XML as the language for encoding the document types. The reason for using XML is that it is efficient for the creation, transformation, delivery, discovery, extension, and reuse of technical/product information. Another reason behind the XML document type popularity is its universality and the support that it provides for transforming documents to meet the needs of information-processing systems as well as humans who desire to collaborate across many geographical locations. As the DITA framework in Figure 1 showed, DITA has a primary function of helping to convert XML documents into HTML, PDF, and other presentation formats (Leslie, 2001). DITA has been adopted by many organizations who value these characteristics as well as its open standard structure. New activities and specifications are governed by an open standard body, OASIS DITA TC and DITA XML.org (http://dita.xml.org/), which is the official community and gathering place for DITA OASIS users. DITA provides many other benefits including the following:

- Modular representation of topic information
- Human metadata creators are not restricted to library and information science professionals but a wider group of people who are experts in their discipline
- Adaptability or extendibility to new areas of information and knowledge quickly
- Allows ease of collaborative authoring of information.
- Addresses translation problems through localization of information (Harrison, 2005).

REFERENCES

Burnett, K., Ng, K. B., Park, S. (1999). A comparison of the two traditions of metadata development. *Journal of the American Society for Information Science* 50.13, 1209-1217.

Day, D., Priestley, M. and Schell, D. (2001). Introduction to Darwin Information Typing Architecture. *IBM developerWorks.* Retrieved, 20 November 2009 from <http://www.ibm.com/developerworks/xml/library/x-dita1/>.

Day, D., Priestley, M. and Hargis, G. (2005). Frequently Asked Questions about the Darwin Information Typing Architecture. *IBM developerWorks.* Retrieved 14 November 2009 from <http://www.ibm.com/developerworks/xml/library/x-dita3/#N210>.

Greenberg, J. (2003). Metadata and the World Wide Web. *Encyclopedia of Library and Information Science*, N.Y.: Marcel Dekker, Inc., 1876-1888.

Harrison, N. (2005). The Darwin Information Typing Architecture (DITA): Applications for globalization. *Professional Communication Conference, 2005. IPCC 2005. Proceedings, IEEE Xplore,* 115-121.

Jovanovic, J., Gasevic, D., Duval E. (2005). Ontology of learning object content structure in Looi, C., McCalla, G., Bredeweg, B. & Breuker, J. (eds.) *Artificial intelligence in education: Supporting learning through intelligent and socially informed technology.* Amsterdam: IOS Press, 322-329.

Leslie, D. M. (2001). Transforming documentation from the XML doctypes used for the Apache Web sites. *Proceedings of the 19th annual international conference on computer documentation.* Sante Fe, New Mexico, USA: *portal.acm.org*, 157 - 164.

NISO. *Understanding metadata*. Bethesda, MD: NISO Press. 1-12.
OASIS DITA TC. (2007). *About the DITA 1.1 Specification,* http://docs.oasis-open.org/dita/v1.1/OS/archspec/ditaspec.html.

Rich, C. (2009). Building task-based user interfaces with ANSI/CEA-2018. *IEEE Computer Society*, 42.8 (2009): 20-27.