Darwin Information Typing Architecture (DITA)

a research note by Namahn

Introduction

The purpose of this research note is to introduce the Darwin Information Typing Architecture (DITA) and highlight its relationship to other information architectures like DocBook and Information Mapping™.

Definition

DITA is an XML-based information architecture for authoring, producing, and delivering modular technical documents that are easy to reuse with varied display and delivery methods, such as helpsets, manuals, hierarchical summaries for small-screen devices, and more.

The architecture consists of a set of design principles, defined in a DTD.
(No schema yet. Documents that are intended to work across systems and have a long life cycle should be defined in the most standard format available, which today is still the DTD.)

The architecture and DTD were designed by a cross-company workgroup representing teams from IBM, Lotus and Tivoli.

DITA building blocks

In DITA a document is made up of a number of topics, each with its own information type.

Topical writing

A **topic** is a chunk of information organized around a single subject. Structurally, it is a title followed by text and images, optionally divided into sections.
Below a typical topic structure:

```
<!ELEMENT topic (title, prolog?, body, %info-types;*>
```

One simple guideline is that a topic is a chunk of information that is small enough to be assimilated by the reader in one go. Generally that's no more than a screen or two of info, online.

The DITA forum provides a lot of information on:

- defining the right topic size
- how to chunk information into topics
- how to nest topics, build relationships between topics

Topical writing is opposed to the historical alternative 'Writing in sections'.

- Writing in sections is the traditional way of handling information. Consider the routine DocBook implies: you have to start with a hierarchical three structure (that represents sections and nested subsections), and then store your information in the fixed tree-structure. The three structure on itself, acting as a TOC, gives access to the sections and subsections. When the information has to be published, you have to stick to the tree-structure.
- Topical writing does not imply any structure at start. Nor have authors to consider how all sections will be nested properly. Topics can later be nested and linked in a separate navigation

map, describing the context of all topics. (This is very much the way W3C promises XLINK will work.).

Scenario **separating content from context**:

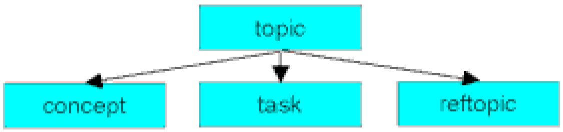
- Write topics, without any relationship to other topics.
- Write a "navigation map" (could be a topic map, an RDF document, something proprietary or - in its simplest form – one or more TOCs) that describes the relationship between a particular set of topics.
- Apply the map (either dynamically, with a map-aware server, or statically, using a publishing transform) to instantiate the relationships in each of the participating topics.

- Another benefit is that topical writing allows for **progressive disclosure**: Because each topic has a title and short description in addition to its full content, the information can be accessed in a variety of viewing contexts (full-screen browsers, integrated help panes, infopops, PDA screens, a.s.o.)

Information typing

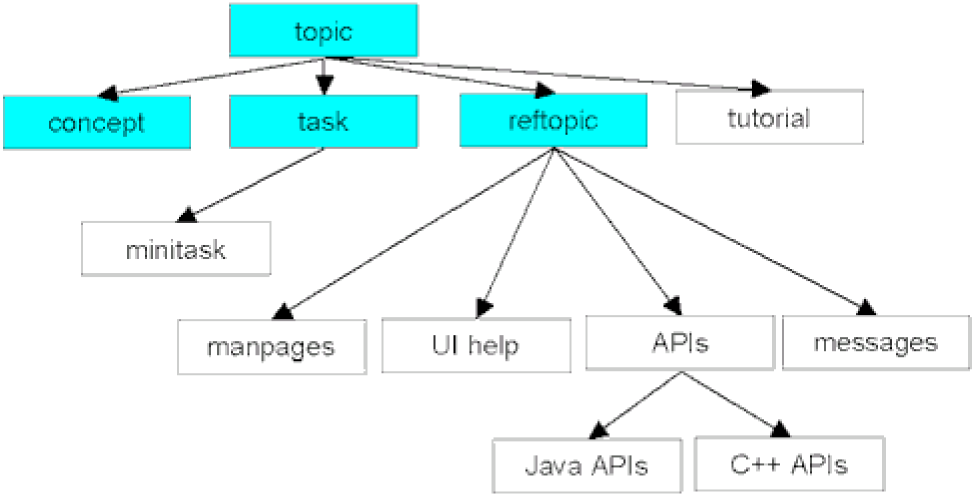
Top level 'core' information types

Each topic has its own **information type**. The information type describes the content of a topic. DITA has four types of primary information types: a generic topic and three core information types: concept, task and reference topic.
Typically, different information types support different kinds of content. For example, a task typically has a set of steps, whereas a reference topic has a set of customary sections, such as syntax, properties, and usage.



Lower level 'specialized' information types

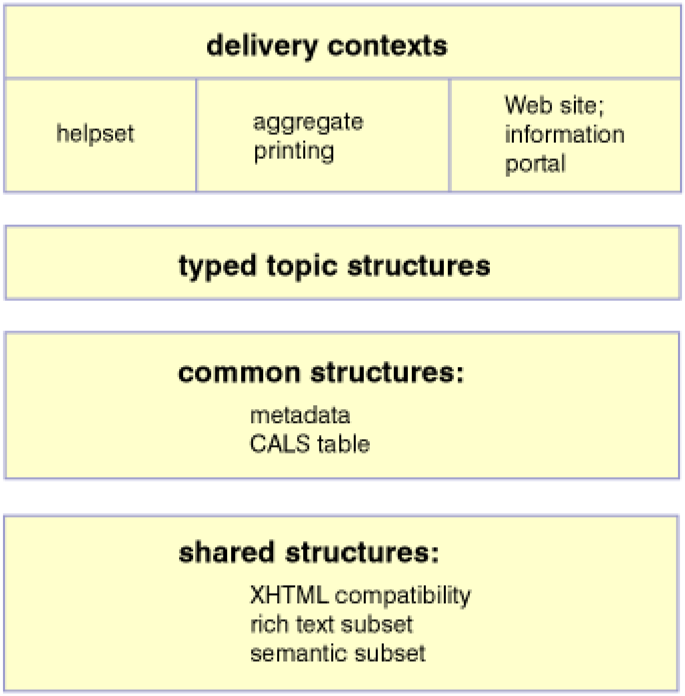
Authors/communities can extend or define additional **specialized information types**, using the existing ones as a base. Rules govern how to specialize safely: e.g. each new information type must map to an existing one, and must be more restrictive in the content that it allows.



.

DITA layers

As shown in figure below, the DITA has four layers that relate to specific design points.



- **Delivery contexts** represent the processing applications for topical information. Also include document management systems, authoring units, translators, and more.
- **Typed topic structures** are standalone units of publishable info.
- **Common and shared structures** are content models that build the basis for the DTDs. Common structures are unchanging structures that can be used within any topic. Shared structures provide elements and content models that can be used in technical documentation. Element names have been borrowed from the HTML-DTD and DocBook-DTD (a.o.).

DITA status

- As DITA stands now, it consists of:
  - A core set of DTD modules to allow authoring in the core topic types and provide a base for creating more specialized topic types.
  - A set of guidelines and samples for creating content, transforms, stylesheets, and new topic types

    See: DITA DTDs, style sheets, sample documents
- In March of 2004, IBM donated DITA to the OASIS standards organization, and updated the developerWorks content about DITA to point to more current information. Please review the information on the "DITA downloads" page.
- Although several companies are testing DITA principles; it (still) is a prototype. There is plenty of work left to be done:
  - Navigation (or topic) maps, for defining topics and how they relate to each other
  - Content management, to provide for centralized tracking and versioning of topics in source repositories

- Tools to allow custom authoring environments
- Specilizations to allow for interchange at common content areas as contextual help and API documentation
- Etc.

DITA versus other information architectures

DITA versus DocBook

DITA advantages over DocBook

- DocBook's solution is primarily focused on delivering books, whereas DITA focuses on delivering topics, which can be assembled into books or linked together into helpsets, websites, hierarchical summaries for small-screen devices, a.s.o.
- DocBook is strictly hierarchical, not providing any mechanism for separating content from context. DITA is much more flexible by storing contextual information in a different document. DITA can process topics at whatever level of any defined hierarchy.
- DocBook is a fixed set of elements and attributes. DITA is extensible and can be tailored to the specific needs of a community: specialization rules define how new, custom information types can be defined, using the existing ones as a base.
- Don Day on DocBook:
  "DITA has the ability to simultaneously accomodate semantic specificity (markup that describes information) and interchangeability (markup that is shared by a large reuse community). The core information types (concept, task, reference, and their common archetype "topic") provide the base of an infinitely extensible type hierarchy. Topics can be processed at whatever level of the hierarchy a particular transform is aware of. More specialized types do not require more specialized processes, and remain reusable by and interchangeable with a much larger reuse community. This is not something DocBook does."

DocBook advantages over DITA

- DocBook, actively developed and refined for more than ten years, has basically become the de facto standard for structured authoring of technical documents, and many other types of documents as well. DITA, though promising and more flexible, is still a prototype.
- DocBook is thoroughly documented. Feedback from the large and loyal user community is widely available. DITA documentation is still lean (e.g. a description of the semantic intent of the elements and attributes is still lacking).
- DocBook provides off-the-shelf support for transforming, formatting, and publishing content as HTML pages, PDF files and other formats (like RTF, FrameMaker MIF, JavaHelp or Microsoft HTML Help).

DITA versus in Information Mapping™

DITA is using some principles that are already common pratice within Information Mapping™: namely the topic as the basic unit of information and the use of information types.

Since DITA clearly uses Information Mapping™ as an inspiration, how do they relate to one another:

- whereas Information Mapping™ has a generic architecture with a fixed set of information types, DITA encourages authors to tailor specialized information types to the specific needs of their audience.
- whereas Information Mapping™, like Docbook, is primarily focused on delivering books, DITA allows for a more flexible presentation of information.

- whereas Information Mapping™ is a product; DITA (and Docbook) are firmly rooted in open source ideals .

Conclusion

DocBook is a robust, proven architecture, widely implemented and well documented. It pays off right away. That is, you get immediate benefits.

DITA is the contender. Challenging and promising, by taking the best out of DocBook (the semantic names and attributes) and Information Mapping™ (topical writing an information typing), and pushing it one stage further by adding flexibility. Major drawback: there is still a lot to be build and documented.

Information Mapping™ has a hard time competing with DocBook and DITA. Not only because it is proprietary, but also because it is mainly an authoring environment (focusing on Word-templates and writing principles), rather than an overall information architecture (like DocBook and Dita) dealing with a much broader scope of issues (like linking, search, transforming and publicing information).

Overview

	Since	**Delivery context**	**Key features**	**Tools**	**Open source**	**Online community**	**Documentation and tutorials**
Doc Book	Early 90's, started as an SGML-DTD	Book oriented, HTML-Help	Large set of elements and attributes that correspond to the general notion of what constitutes a "book"	Off-the-shelf support for transforming, formatting, and publishing content to PDF, HTML, HTML-help	Firmly rooted in the open source movement	Well supported by the XML community	Well-documented
DITA	2000	Broad delivery context	Topic based, separation of content and context, flexibility	Announced	Open source	Small, slowly growing community	Announced
Info Map	History dates back to the seventies and eighties (did not evolve very much since)	Book oriented	Authoring environment (Word-templates) and writing principles, DTD is limited compared to DocBook	Supported by a set of tools, tools are not state of the art	Proprietary	No online community	Well-documented

Resources

DITA

- Introduction to the Darwin Information Typing Architecture
- Specialization in the Darwin Information Typing Architecture
- DITA FAQ
- DITA forum (moderated by Don Day and Michael Priestley)

DOCBOOK

- The official DocBook homepage
- Introduction to DocBook by Norman Walsh
- OASIS, the maintenance organisation for DocBook
- Michael Smith, moderator of the XML-doc mailing list, recommending DocBook
- DocBook schemas, stylesheets and related resources on SourceForge
  - Website customization layer
    http://prdownloads.sourceforge.net/docbook/website-2.0b1.zip
  - Slides customization layer
    http://prdownloads.sourceforge.net/docbook/slides-2.0a2.zip
  - Jirka Kosek's XSLT system for generating HTMLhelp and Norm's original system for generating JavaHelp -- both part of the standard DocBook XSLT stylesheet distribution
    http://prdownloads.sourceforge.net/docbook/docbook-xsl-1.45.zip
  - Steve Cheng's DocBook2X for generating man pages, Texinfo files
    http://prdownloads.sourceforge.net/docbook2x/docbook2X-0.7.0.tar.gz

INFORMATION MAPPING

- IMAP research note

Namahn bvba
Grenstraat 21
B-1210 Brussels
www.namahn.com
Tel. +32 (0)2 209 08 83