A Style Guide for Authoring Documents using DITA

An Exegesis submitted for the fulfillment of the requirements for the degree of Doctor of Philosophy
2011

Anthony Self

SWINBURNE UNIVERSITY OF TECHNOLOGY

Table of Contents

Abstract

The Darwin Information Typing Architecture (DITA) is an XML-based architecture for writing and delivering information using structured authoring techniques. It is aimed at technical communicators, authors, editors, and creators of textual content. The DITA Technical Committee of OASIS guides the ongoing development of the standard.

The approach enabled by DITA represents a major shift in the way in which technical communicators and other writers approach a documentation project. DITA effectively separates the content from the presentation; technical communicators using DITA can no longer use traditional *style-based* techniques and authoring tools to write manuals and other documents. This is because DITA is based on semantic mark-up and integrated metadata, both of which are new concepts for most technical communicators.

DITA may become the accepted approach and *de facto* standard for documentation projects. If this occurs, best practices for DITA authoring will need to be devised to allow writers to take advantage of the new approach. Early adopters of DITA are in a strong position to find these best practices, either through logical re-application of traditional approaches or by trial and error.

The research in this project addresses the question of **what constitutes best practice in DITA authoring**, using an *Action Research* methodology, and captures those best practices into a *Style Manual for Document Authoring with DITA*. *The DITA Style Guide* is an original synthesis that finds new knowledge through the analysis and development of existing disparate material together with my own practical experience.

Rather than a traditional dissertation thesis, this project is a “PhD by Artefact and Exegesis”. The primary product of the research is an Artefact (*The DITA Style Guide: Best Practices for Authors*), published as a substantial book. *The DITA Style Guide* is itself written using DITA, and is accompanied by an Exegesis reflecting on the process of writing the book within the DITA approach. In fact, both the Artefact and the Exegesis have been written entirely within a DITA framework, incorporating the very practices they recommend. The project thus extends the understanding of how DITA can be practically implemented.

The DITA Style Guide is designed to serve as an supra-organisational set of guidelines for DITA authors world-wide. In the medium term, it is planned that *The DITA Style Guide* will be migrated to a Wiki format and donated to the community as an open source project, to become a collaboratively maintained reference work.

Document Conventions

This document features a number of formatting and presentation conventions.

Some conventions are by virtue of the document's origins as DITA source, some are implementations of the best practice recommendations of *The DITA Style Guide*, and some are deliberate design decisions.

This document is written in the first person, primarily to make it clear whether I am writing about myself or DITA authors in general, but also because as an Exegesis this document is a reflective piece of writing.

This document was created following a topic-based approach, where individual standalone topics were written independently and later assembled into a coherent published work.

Each topic has a short description (or `shortdesc` element in DITA semantics) which summarises the content of the topic. In the printed and PDF versions of this document, the short description is presented at the top of the topic in italics, with a grey vertical bar at its left. In the HTML-based versions of this document, the short description is presented in grey Roman text with a book icon at its left.

Although this document has been written to the rules of *The DITA Style Guide*, some topics pre-date some rule decisions, and therefore may not be entirely conformant!

Sentence case has been used for topic titles, except for chapter titles which are in title case.

In the printed and PDF versions of this document, the first topic in each chapter is indented and presented in a smaller text size. The indent is to provide for a mini Chapter Table of Contents on the left of the page. The first topic is an introduction to the chapter.

In the printed and PDF versions of this document, block quotations are presented in indented italics, with a single quotation mark icon at its left. In the HTML-based versions of this document, block quotations are in indented grey Roman text with a double quotation mark icon at its left. Inline quotations are presented within double quotation marks.

The publishing process which converts the DITA source to reading formats (such as paper and CHM) is automated, using the DITA Open Toolkit and WinANT Echidna. There are currently a number of limitations that prevent some sophisticated page layout features with that tool set. Those limitations include the positioning of page breaks, text wrapping, and table formatting. I considered it more important to write this document in DITA than to achieve particular formatting outcomes.

References are listed in a Bibliography at the end of the chapter content but before the Appendixes. In-text citations are formatted in italics. To make it easier to reference cited works, references relating to some essays or academic papers are also included at the end of the essay or paper. These references are *transcluded* from the master reference list at the end of the chapter content using the DITA *conref* mechanism.

In the printed and PDF versions of this document, illustrations are numbered as figures if they are standalone in nature. If an illustration is used in context of the flow of the text (where the illustration cannot be used outside that context), then it will not be semantically identified as a figure. In the HTML-based versions of this document, figures are not numbered.

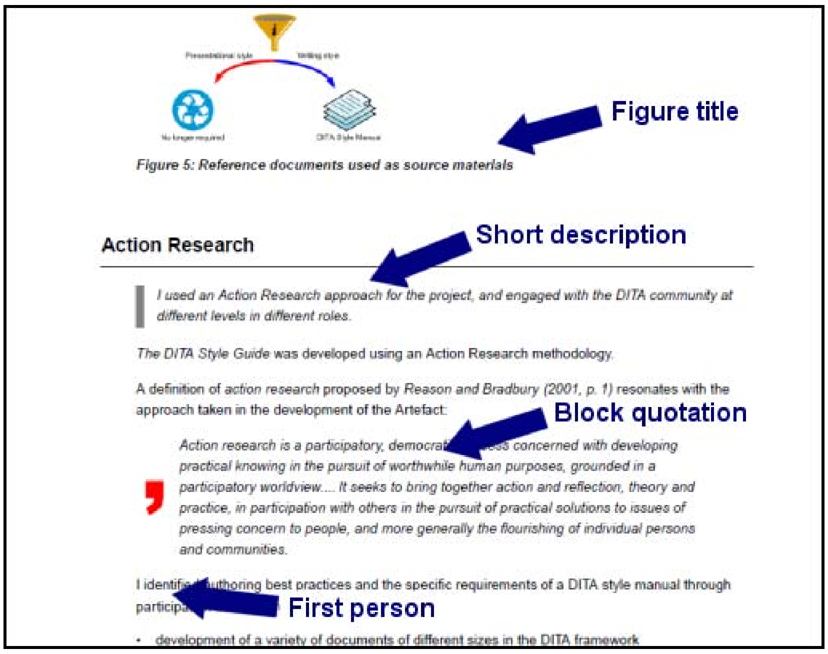


Figure 5: Reference documents used as source materials

Action Research

I used an Action Research approach for the project, and engaged with the DITA community at different levels in different roles.

The DITA Style Guide was developed using an Action Research methodology.

A definition of *action research* proposed by *Reason and Bradbury (2001, p. 1)* resonates with the approach taken in the development of the Artefact:

> *Action research is a participatory, democrat[illegible]ess concerned with developing practical knowing in the pursuit of worthwhile human purposes, grounded in a participatory worldview.... It seeks to bring together action and reflection, theory and practice, in participation with others in the pursuit of practical solutions to issues of pressing concern to people, and more generally the flourishing of individual persons and communities.*

I identifi[illegible]uthoring best practices and the specific requirements of a DITA style manual through particip[illegible]

- development of a variety of documents of different sizes in the DITA framework

In the printed and PDF versions of this document, notes, tips and postscripts (admonishments) are identified with a pencil icon at the left of the paragraph. In the HTML-based versions of this document, admonishments are identified with a bold prefix label.

In the printed and PDF versions of this document, double brackets may occur in cross-references, where a page number reference is generated automatically in the publishing process. That page number reference is not generated in the HTML-based versions of the document. Any greater reworking of the publishing process to circumvent this issue is beyond the scope of the research project.

The language used in the Exegesis is Australian English.

Student Declaration

This Exegesis contains no material which has been accepted for the award to the candidate of any other degree or diploma.

To the best of my knowledge, this Exegesis contains no material previously published or written by another person except where due reference is made in the text of the examinable outcome; and where the work is based on joint research or publications, discloses the relative contributions of the respective workers or authors.

[illegible]

Anthony Self

14 July 2011

Chapter 1

Introduction to the Exegesis

Topics:

The *Darwin Information Typing Architecture (DITA)* is an emerging approach to writing manuals, online Help systems and other publications, where XML is used to separate content from form. The DITA method of semantically identifying elements of a document contrasts with traditional *style-based* approaches to authoring, where the author creates a document by reference to how it will look in one particular output format (such as print). DITA enables flexible *single-source* publishing, where the same content can be published to different delivery formats, including Help, Web, paper and eBook.

DITA is an innovation with the potential to change habits of composition, expression and presentation for a broad range of writers and in so doing enhance productivity across a number of industries and occupations. Establishing best practice in this emerging field will enable users to maximise the productivity dividend.

The challenge of semantically identifying document elements confronts every new DITA author. When starting to work with DITA in 2005, I observed that there were no guidelines for best practice application of the DITA standard, and indeed very few resources to help a new author adopt consistent mark-up practices. Further, I learned that people (or teams) adopting DITA were making their own interpretations and approaches. This ad-hoc practice would render interchange of documents between DITA authors impractical. I formed the impression that DITA authors were using the *Yahoo! DITA Users Group* as a way of determining what common practices were.

My own experience, when starting authoring in DITA in 2006, of the need for a guide to best practices led to his research question of “what constitutes best practice in DITA?”. An *action research* approach to the project was adopted, with the result being an Artefact and Exegesis. The Artefact is a *The DITA Style Guide*, which aims to encapsulate a set of best practices for DITA authoring. The best practices were derived from analysis of the Yahoo! DITA Users Group and OASIS DITA Technical Committee mailing lists, and through my own experience as a practitioner immersed in the DITA community.

My *action research* activities included participating in the OASIS DITA Technical Committee and OASIS DITA Adoption Technical Committee work, developing an open source DITA publishing software utility named WinANT Echidna, participating in the DITA Open Toolkit open source project, developing the curriculum and materials for a post-graduate level *Structured Authoring with DITA* university subject, running DITA training workshops, and presenting at industry and academic conferences on DITA-related subjects.

The development of the Artefact itself, and this Exegesis, also formed part of my *action research*, as I wrote all content in DITA using the best practices formulated in the Artefact.

The Artefact was published by Scriptorium Press in March 2011, and I have committed to donating the work to the DITA community as an open source resource within two years.

The nature of the Exegesis

> *The Exegesis is a first person reflection on the process of developing The DITA Style Guide.*

As a published work for general publication, the Artefact (*The DITA Style Guide: Best Practices for Authors*) only contains best practice recommendations for semantic authoring in DITA. By contrast, the Exegesis is a record of the methods, sources, decision-making process and discoveries made during the writing of the Artefact.

The Exegesis is written in the first person, reflecting the nature of the work as a primarily reflective piece of writing. The Exegesis constitutes an original and unpublished component of the PhD project.

As the project follows an *action research* approach, the Exegesis draws together “action and reflection, theory and practice”*(Reason and Bradbury, 2001, p. 1)*. It “records the process of turning information into knowledge” *(Arnold, 2005)*.

What are style guides?

> *Style guides are documents used as authoritative standards to record and arbitrate on issues relating to style.*

The term *style guide* or *style manual* describes a set of rules that govern an authoring team's use of language, the writing and publishing processes, and the presentation of textual content in documents of different types. Most style manuals were developed for the specific needs of a particular publishing house, but a small number were made publicly available, and were consequently adopted more broadly outside the of originating organisation. Style manuals tend to be categorised as in-house, organisational (private) or general use (public).

During a literature review, I analysed a large number of style guides, and deduced that the seven key functions of style guides are:

- to create consistency
- to promote a professional image
- to improve the quality of writing
- to serve as an induction or training guide for new writers
- to document publication processes
- to make it easier to localise documents

- to create the conditions for effective interchange of documents

The most overriding function is to create consistency. A style manual is in essence a tool used by writers and editors to create consistency in spelling, punctuation, presentational style, use of language, terminology, and writing style. However, the term *style guide* is also sometimes applied to a tool used to create consistency in graphical presentation and aesthetics; these documents should be more correctly referred to as *corporate image guides*.

When style manuals make recommendations, or set rules, they are generally not identifying correct or incorrect use, but rather documenting the standard or agreed usage from the many possible usage options. In that way, style manuals document style decisions, so that the same issues are not re-visited.

A widely used style guide in the English speaking world is *The Chicago Manual of Style*.

The need for a DITA style guide

> *The change from style-based authoring to structured, semantic DITA authoring has created a need for a style manual appropriate to this new authoring approach.*

Existent conventional style manuals do not provide any guidelines for semantic mark-up, and in general devote a large proportion of their content to styling conventions. DITA is a style-agnostic authoring environment, a result of the DITA standard's embrace of the philosophy of the separation of content and form. As a result, existent style manuals do not provide sufficient guidelines to promote consistency in DITA authoring, and the need for a style guide specifically focussed on semantic authoring in DITA is apparent.

The shift from print publishing to Web publishing exposed a similar need for a style manual for the new medium that the shift from style-based to structured authoring has.

> *The web is still a young medium with no standards organizations to canonize existing typical page layout practices. Until we have a Chicago Manual of Style for the web, we can at least combine current mainstream web design practice, user interface research, and classic page composition to form recommendations for the location of identity, content, navigation, and other standard elements of pages in text-dominant, information-oriented web sites.*
>
> ***Lynch and Horton, 2009***

In *The DITA Style Guide*, I followed *Lynch and Horton's* suggested approach for Web style guides of "canonising" existing practices. Those existing practices were distilled from an analysis of Yahoo! DITA Users Group and DITA Technical Committee peer support correspondence.

Content re-use and re-purposing in a modular, structured authoring environment requires a greater degree of consistency than in traditional publishing. Likewise, the opportunities to interchange content with other parties are greater in structured authoring. Style manuals are therefore more important in such an environment than in non-structured, style-based regimes.

Style guides often specify which option to use when several options exist.... The specific content in the style guide is not usually a matter of "correct" or "incorrect" grammar or style, but rather the decisions you or your employer or client have made from among the many possibilities.

Hollis Weber, 2009

A style guide creates consistency by specifying which option to use, even though each option may be grammatically and technically valid.

According to *Hart (2000)*, style guides “fill the gap between the need for consistency and the means of being consistent”. *The DITA Style Guide* is designed to fill that gap.

Types of style guides

Style guides can be grouped into two categories: house, and general use.

There are two types of style guides: organisational house style guides, and general use (supra-organisational) style guides.

House style guides are usually developed inside an organisation, for the organisation's internal use. Their scope tends to be limited to the specific writing needs and contexts of the organisation.

General use, or *generic*, style guides are often commercially-available, though many are free access, Web-based documents. Some Web-based style guides, such as the Associated Press *AP Stylebook*, are made available on a subscription basis.

According to *Hart (2000)*, generic style guides (or a “published guide”, as he puts it) covers 90% of the communication needs of a publishing organisation, so house style guides cover the remaining 10% of organisation-specific needs.

Is *style manual* the correct term?

The product of my research into what constitutes best practice in DITA is a style manual for DITA adopters. This topic addresses the question of whether the term Style Manual is even valid in an authoring environment where style is separate from form.

The term *style guide* or *style manual* describes a set of rules that govern an authoring team's use of language, the writing and publishing processes, and the presentation of textual content in documents of different types. Most style manuals were developed for the specific needs of a particular publishing house, but a small number were made publicly available, and were consequently adopted more broadly outside the of originating organisation. Style manuals tend to be categorised as in-house, organisational (private) or general use (public).

Style manuals are approached in different ways by their authors. *Barker (1998)* defines the style guide as a "collection of conventions of grammar, punctuation, spelling, format and other matters...". *Damrau (2005)* identifies the two purposes of a style guide as to show "the basic concepts for producing... material that has a uniform look", and to embody the less tangible "corporate culture and values". *Price and Korman (1993)* see a style guide's role as setting standards for mechanics and voice, and keeping a record of made decisions (on matters of contention). The different breadths of style guide definitions seem to reflect the variability of what constitutes the documentation process. As *Jones (1997)* states in describing the confusion of what constitutes *style*, "[some] style guides leave us with the impression that everything in the documentation process - from planning to production - is a matter of style".

DITA separates content (the words) from form (the presentation). The DITA standard is carefully distanced from issues of presentation and format. The transformation of raw DITA content into a readable format for delivery (such as paper, RTF or HTML), is referred to as "processing", and the DITA standard does not cover the processing process. DITA is an authoring standard, not a processing or presentational standard.

The word "style" infers presentational or formatting style, but it may also refer to writing style. A DITA style manual would not address issues of presentational style, but may be able to guide matters of writing style. To avoid confusion, a guide to the effective use of the DITA language for authors of documents would more accurately named a *DITA Authoring Standards Manual*, but it would still serve a similar purpose to traditional style manuals.

Example of the challenges

> *Understanding some of the difficulties for writers working consistently within an XML structure is aided by a simple example.*

DITA is a set of structure rules. Those rules, however, still require interpretation to be effectively and consistently used. For example, DITA includes a number of block element structures, including paragraph (`p`) and unordered list (`ul`). DITA permits a paragraph to contain a list, and it permits an item in a list (`li`) to contain a paragraph. It also permits paragraphs and lists to be standalone block elements. A sentence introducing a list could therefore be coded in DITA using any of three structures:

```
<p>There are two groups of resources that can be drawn upon to
determine what constitutes DITA best practice:
```

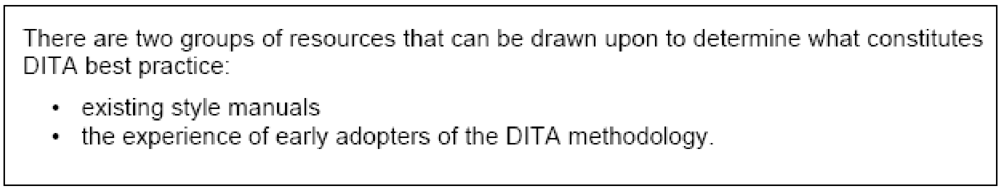
```
</p>
<ul>
 <li>existing style manuals</li>
 <li>the experience of early adopters of the DITA methodology.</li>
</ul>
```

```
<p>There are two groups of resources that can be drawn upon to
determine what constitutes DITA best practice:
<ul>
 <li>existing style manuals</li>
 <li>the experience of early adopters of the DITA methodology.</li>
</ul>
</p>
```

```
<p>There are two groups of resources that can be drawn upon to
determine what constitutes DITA best practice:
</p>
<ul>
 <li><p>existing style manuals</p></li>
 <li><p>the experience of early adopters of the DITA
methodology.</p></li>
</ul>
```

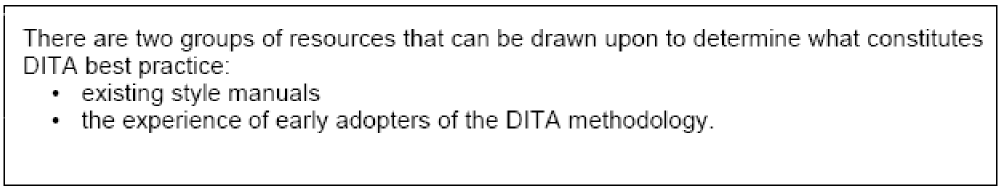
All of these examples are technically valid DITA. One of the benefits of DITA is meant to be consistency, or less ambiguity. Collaboratively-developed documents, where different authors create different parts of the same document, is also a central feature of DITA. If three different authors choose three different structuring approaches for one simple block of text, where is the consistency?

Authors used to style-based authoring may make a decision on which form to use based on the result of the structure in the output. For example, an author might experiment with the first example, and finds that the PDF output (from a common DITA publishing tool) looks like:

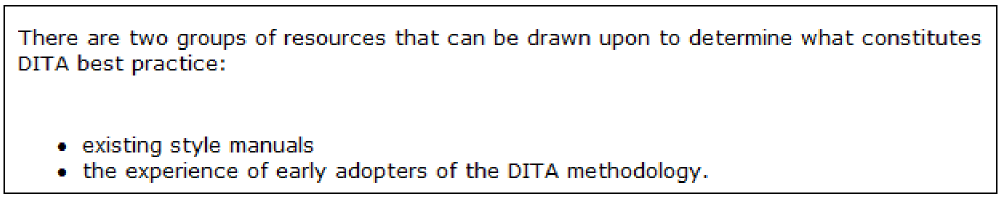

There are two groups of resources that can be drawn upon to determine what constitutes DITA best practice:

- existing style manuals
- the experience of early adopters of the DITA methodology.

The author may therefore choose this version over the second example, which renders into PDF in a less aesthetically-pleasing form:


There are two groups of resources that can be drawn upon to determine what constitutes DITA best practice:
- existing style manuals
- the experience of early adopters of the DITA methodology.

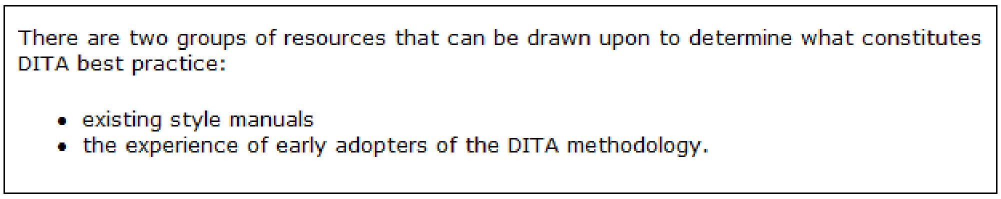
The resulting choice of this authoring decision may or may not be the correct one, but the decision process used is certainly wrong. A DITA fundamental is the separation of content from form, so content decisions cannot be governed by matters of form. The form of the output is entirely separate from DITA authoring, and can be easily modified in the publishing process. For example, the *correct* first example would render in HTML, using a common DITA publishing tool, as:



There are two groups of resources that can be drawn upon to determine what constitutes DITA best practice:

- existing style manuals
- the experience of early adopters of the DITA methodology.

This is an aesthetically poorer form than would have the second example, rendered in HTML with a smaller gap between the lead-in sentence and the list:



There are two groups of resources that can be drawn upon to determine what constitutes DITA best practice:

- existing style manuals
- the experience of early adopters of the DITA methodology.

The correct decision process is to work out which of the three examples is the one that is semantically correct. This is often a difficult decision, sometimes contentious, and nearly always arguable.

This dilemma crystallises an outstanding challenge in writing in DITA, and one that can be resolved by a common *style guide*. Even if the style guide suggests the wrong answer, at least it will provide a consistency that is, in practice, more important than correctness!

Postscript: To me, the second example is the more semantically correct, as the sentence belongs to the list, and the sentence and the list only make sense when they are a whole.

Audit trail of rationale behind rules

> *The product of the Action Research project is an Artefact and Exegesis. The Artefact is published as a conventional book entitled The DITA Style Guide. Although not published*

> *in The DITA Style Guide, the source content of the Artefact contains rationale for the best practice recommendations. This rationale provides an audit trail for the decisions made.*

Style Manuals offer rules on issues of contention in the writing process. They do not provide a rationale for a ruling. Style manuals speak with the authority of the publisher; a ruling in *The Chicago Manual of Style* carries the authority of the University of Chicago Press.

For example, the *Microsoft Manual of Style for Technical Publications*, carrying the authority of Microsoft and Microsoft Press, makes the following edict:

> *garbage collection*
>
> *Even though it's jargon, this is a commonly used and acceptable term in programming documentation. It refers to the automatic deletion of objects that the environment perceives are no longer being used or to the automatic recovery of heap memory. A more formal but much less understandable synonym is "unpredictable finalization."*

The Manual provides no rationale or explanation of why the term is “acceptable”.

Likewise, *The Chicago Manual of Style* states:

> *A period should be omitted at the end of a sentence that is included within another sentence.*

Again, no rationale is provided; the Style Manual is a rules book, not a discussion document.

The DITA Style Guide does not carry the authority of a publishing house, so the need for providing a rationale for its recommendations is important. (Although to a certain extent, providing any set of rules, rational or not, will at a minimum provide a framework for consistency.) *The DITA Style Guide* will, in the medium term, be donated to the DITA community and migrated to a Wiki format, where the rationale and background is an important part of the consensus process. As *Hyttinen (2009, p. 59)* succinctly put it, “participation bonds the writers to the style guide”.

The rationale for rules and recommendations in *The DITA Style Guide* have been recorded within the source DITA mark-up of each topic, but excluded from the commercially published work. This approach provides an *audit trail* for the decision-making process, which will become useful when the work is donated to the DITA community to become a collaboratively developed resource.

Note: A version of the Artefact that includes the rationale hidden in the published work is provided on the CD-ROM accompanying this Exegesis.

The Exegesis also provides a record of some of the decision-making processes, providing longer discussions of writing style issues that are not appropriate for *The DITA Style Guide* itself. However, the primary purpose of the Exegesis is to document the process of creating a large publication in DITA, highlighting and reflecting on the problems uncovered, the solutions and workarounds found.

Whether to include rationale in *The DITA Style Guide*

> *Some style manuals include not only rules and guidelines on styling decisions, but the rationale behind such edicts. The question of whether to include rationale in The DITA Style Guide was considered, and I decided to omit rationale from the published work.*

Historically, style manuals have included a wide range of subject matter. The separation of content and form requires a different approach for *The DITA Style Guide*. Further, the generic (supra-organisational) nature of *The DITA Style Guide* requires a different approach than that for a corporate style guide.

Hollis Weber (2007) suggests that a style manual should **not** contain:

- process information (how we do things in this company or this department)
- design information (what our documents should look like)
- grammar and writing tutorials
- rationale for decisions

While the Artefact does not cover those areas, the Exegesis provides some of the rationale for upper level decisions, and conditional text within the Artefact source topics (but not published into *The DITA Style Guide*) provides some specific rationale for later use when the work is donated to the DITA authoring community as a Wiki resource.

Chapter 2

Literature Review

Topics:

The opportunity for literature and Web review as a tool in answering the research question of what constitutes DITA best practice in DITA is somewhat limited. To date, publication in the field of DITA has been very thin. Most literature in the field is informal articles published on a variety of Web sites. The largest repository of articles is found on the IBM Web site; DITA was originally invented and developed within IBM before being donated to the open source community. When the project began, only one book (of two editions) on the subject of DITA had been published *(Linton and Bruski, 2006)*. During the course of the project, just two other books, Practical DITA *(Vazquez, 2009)* and DITA 101 *(Rockley, Manning, and Cooper, 2009)* were published. The majority of knowledge of DITA is embedded within forums and user community mailing list archives, user guides and language reference documents provided with the DITA Open Toolkit, and within the OASIS DITA XML Web infrastructure. It is difficult for a new user of DITA to find and filter these disparate information sources.

However, numerous style manuals have been published, including some for Web sites. The availability of Web style manuals is particularly useful, because the Web adopts a partial separation of content and form. (The separation of content and form is a key difference between DITA authoring and earlier style-based authoring techniques.) The approach taken by style manuals for newspaper journalists, where journalists play no part in the bulk of the presentational format, is also instructive.

An analysis has been made of the make-up of a variety of style manuals, identifying whether the style rules concern editorial (writing) style or presentational style.

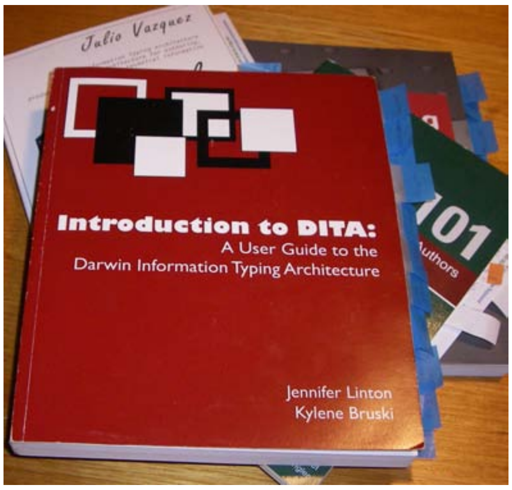


Figure 1: Use of Bookmarks in Reference Books During Literature Review

What style guides cover

A style guide provides advice on the optimal use of approaches, procedures and processes.

The primary purpose of a *style manual or style guide* is to promote consistency. Style guides also aim to codify best practice. (What the term *best practice* actually means is not universally understood or agreed. My interpretation is that best practice is the most effective technique or method used to accomplish a task.)

A *style manual or style guide* is a set of standards governing the design and writing of documents, and usually takes the form of a printed manual. Publishing organisations, standards bodies, government agencies and publication departments within an organisation are the typical originators of style manuals. Technical publication style manuals used by technical communicators, such as the *AGPS Manual of Style* and *The Chicago Manual of Style*, devote a number of chapters to the publishing process, which, in the DITA model, is removed from the technical communicator's domain. Conversely, very little space is devoted to the semantic identification of content elements, something that is core in DITA. The major purpose of a style manual is to promote consistency, and one of the difficulties of DITA adoption is working without a style guide appropriate to the DITA paradigm.

Kim Lockwood, author of a number of News Corporation style manuals, described the role of style guides in an interview with Jill Kitson *(The Newspaper Style Manual, 1999)* for the ABC radio programme *Lingua Franca*:

> *Style manuals serve two purposes. First, they are intended to help achieve consistency through the paper. They show the choices the newspaper has made in how to spell and use certain words, how to use punctuation and how to use figures. We can argue endlessly about etymology, popular usage and the credibility of this or that dictionary, but finally the newspaper must simply choose how it will spell "glamour", with or without a "u".... A style book is therefore a compilation of choices made by its editor to be followed by all staff.... Second, newspaper style manuals aim to be reference works tailored to the needs of journalists.*
>
> **Kim Lockwood**

A literature review that focusses on the areas of the publishing process that different style guides cover will provide an understanding of the type of guidelines and subject areas that should be provided in a DITA style guide for semantic authoring.

Content and form in style guides

> *When analysed to identify the relative emphasis of individual style guides was focussed on content and form, wide variations were found.*

The contents of four typical style guides were analysed in the context of the separation of content and form. This concept is key to DITA, and is one of the main differences between semantic authoring and conventional style-based authoring.

The analysis involved identifying the nature of page content as being most focussed on content issues or presentational (form) issues, and comparing the relative page counts for each type. The raw results were as follows.

AGPS Manual of Style **(Snooks and Co, 2002)**

- Part 1: Mixed 184
  - Part 1 Presentational: 42
  - Part 1 Content: 142
- Part 2: Presentational 95
- Part 3: Presentational 35

Total: 172 - 55% (Presentational) 142 - 45% (Content)

NZ Style Book (Wallace and Hughes, 1995)

- Part 1: Mixed 110
  - Part 1 Presentational: 20
  - Part 1 Content: 90
- Part 2: Presentational 45

Total: 65 - 42% (Presentational) 90 - 58% (Content)

The Chicago Manual of Style **(1982)**

- Part 1: Presentational 104
- Part 2: Mixed 380
  - Part 2 Presentational: 231
  - Part 2 Content: 149
- Part 3: Presentational 25

Total: 149 - 30% (Content), 360 - 70% (Presentational)

News Limited Style (Lockwood, 2005)

- Part 1 (Grammar): Content

- Part 2 (Punctuation): Content
- Part 3 (Spelling): Content
- Part 4 (Names): Content
- Part 5 (Capital Letters): Content
- Part 6 (Titles): Content
- Part 7 (Keeping it Clean): Content
- Part 8 (Style Specifics): Content
- Part 9 (Legal): Content

Total: 100% Content

Separation of content and form

The use of semantic mark-up in DITA, where text elements are marked up based on their meaning, allows the content to be completely separated from its rendition and display to the reader. For example, a term is marked up as a `<term>` and a citation as a `<cite>`, and no information about how those elements will be displayed is stored in the content. Stylistic (display) rules are applied when the DITA content is *transformed* into a reading format, such as HTML or paper. In a DITA workflow, documents are created as collections of modular, re-usable topic files, and mechanisms allow not only the format to be separated from the content, but also the context. The same topic may be a section in the context of one publication, but a sub-section in the context of another. The intermingling of content, format and context in a style-based document workflow essentially eliminates the possibility of re-use. Once a paragraph is styled as having a 13 cm left margin, it cannot be used on paper 12 cm wide. A phrase marked up in italic won't render as italic on a reading device that doesn't support italic. But a citation identified as a citation in a DITA topic can be processed to italic by one transformation process, to bold red by a different transformation process, and to synthesised voice by another transformation process.

Most popular styles guides mix content (or *editorial*) style rules (eg, “use active voice for instructions”) with presentational (*form*) style instructions (eg, “use italics for product names”). The pie-charts in *Figure 2: Breakdown of Content (Editorial) and Presentational Style Rules in Four Style Guides* (see page 190) (*Snooks and Co, 2002*, *The Chicago Manual of Style, 1982*, *Wallace and Hughes, 1995*, *Lockwood, 2005*) show a simplistic calculation of the proportion of content and presentation style rules in three common publishing industry style guides, and a contrasting newspaper style guide, based on the number of pages devoted to each type of rule.

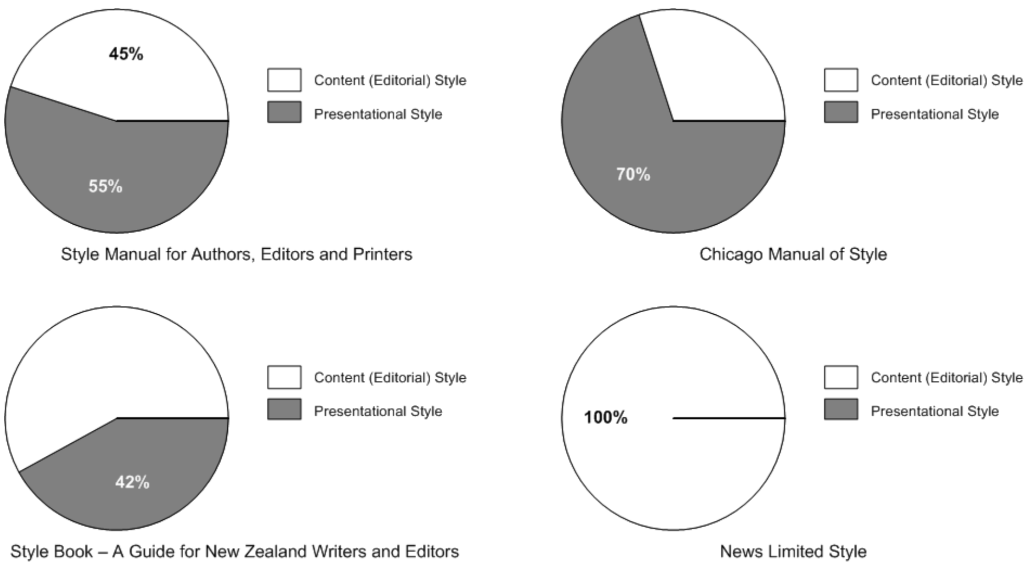


Figure 2: Breakdown of Content (Editorial) and Presentational Style Rules in Four Style Guides

Unlike article publishing organisations such as STC, the separation of content and form in the newspaper publishing process, particularly at News Limited, is complete. Journalists submit stories to the news content management system without any presentational style information. Those stories (which are modular in nature) are single-sourced to Web editions of the paper, traditional paper editions, syndicated newspapers, database archives, and RSS XML feeds.

Style guides

> *The style guides reviewed can be categorised as conventional style guides, writing style guides, or Web publishing style guides.*

In reviewing a large number of publications that could be defined as "style guides", it was found that the works could be grouped into categories of:

conventional style guides
writing style guides
Web publishing style guides

Conventional style guides tended to have a larger proportion of their content devoted to presentational issues. For example, 70% of *The Chicago Manual of Style (1982)* and 55%

of the *AGPS Manual of Style* relate to presentational issues. (Refer to *Content and form in style guides* (see page 23).)

Writing style guides are those aimed at teaching writing and language skills, and do not include much information on presentational form.

Web publishing style guides are those produced to promote consistency in Web-based publications and Web sites. These guides often include Web authoring instructions. For example, the Monash University *Web Style Guide (n.d.)* contains sections explaining HTML and CSS. (Refer to *Monash University Web Style Guide* (see page 46).)

Conventional style guides

A review of a sample of conventional style guides (*conventional* meaning style guides designed primarily for printed documents) was conducted. The style guides ranged from those first published in 1906 (The Chicago Manual of Style, 1982) to new works such as Wired Style *(Hale, 1996)*.

The review focussed on identifying the high level structure of the information, and special characteristics of the publication.

Chicago Manual of Style

> *The Chicago Manual of Style is arguably the predominant English language style manual. It is made up of 18 parts covering language and publishing issues.*

The dominant pre-Web publishing style guide worldwide is arguably *The Chicago Manual of Style (1982)*. The top level table of contents for its 15th edition is as follows.

- Preface
- Acknowledgments
- 1 - The Parts of a Published Work
- 2 - Manuscript Preparation and Manuscript Editing
- 3 - Proofs
- 4 - Rights and Permissions William S. Strong
- 5 - Grammar and Usage Bryan A. Garner
- 6 - Punctuation
- 7 - Spelling, Distinctive Treatment of Words, and Compounds
- 8 - Names and Terms

- 9 - Numbers
- 10 - Foreign Languages
- 11 - Quotations and Dialogue
- 12 - Illustrations and Captions
- 13 - Tables
- 14 - Mathematics in Type
- 15 - Abbreviations
- 16 - Documentation I: Basic Patterns 593
- 17 - Documentation II: Specific Content 641
- 18 - Indexes 755
- App. A - Design and Production - Basic Procedures and Key Terms 803
- App. B - The Publishing Process for Books and Journals 857
- Bibliography 863
- Index 881
- Terms of Use

The majority of the content covers presentational matters, which are not relevant to a DITA style guide for authoring where content (authoring) is separate from form. For the content that relates to authoring, most rules are concerned with the crafting and standardisation of words. Some content, such as that concerning punctuation, may overlap and even conflict with recommendations in a DITA style guide. For example, punctuation rules in lists where list items may be re-used or re-ordered may need to be changed for DITA authoring.

Style (News Limited)

> *Style: The essential guide, as the in-house style guide for New Limited's Australian news organisations, is used as a reference by more than 120 newspapers.*

Style: The essential guide for journalists and professional writers (Lockwood, 2005) is an unusual style manual because it has been written specifically for an industry (newspaper journalism) where content and form have already been separated. It covers spelling conventions, grammar and syntax rules, punctuation rules, and legal requirements.

It reads in some parts like an English text book, explaining how to use apostrophes, the question of split infinitives. Its guidelines on the use of dashes is particularly good, because it doesn't mention typographical dashes (em dashes, en dashes), etc.

Style does not contain any guidelines about presentational form. The book is divided into nine sections:

- Part 1 (Grammar): Content
- Part 2 (Punctuation): Content
- Part 3 (Spelling): Content
- Part 4 (Names): Content
- Part 5 (Capital Letters): Content
- Part 6 (Titles): Content
- Part 7 (Keeping it Clean): Content
- Part 8 (Style Specifics): Content
- Part 9 (Legal): Content

Style is an excellent resource, but it does not address any issues relating to semantic mark-up generally, and understandably does not contain any specific DITA style content.

IBM Style

> *Many information technology companies have their own style guides, and some of these find their way into the public sphere. IBM® made available their internal IBM Style manual in PDF format, via the IBM Design Web site.*

IBM Style is designed as a paper document, but written and managed within a Lotus Notes database, and was made available for public download as a PDF file. The document is organised into topics by style issues, arranged into a simple, flat, alphabetical sequence. This alphabetical topic approach is also used by other style guides, including the *Microsoft Manual of Style for Technical Publications*.

Examples of the topics in *IBM Style (2009)* are:

- Abbreviations
- Anthropomorphism
- Colons
- Company names
- Mouse buttons
- Plurals
- Times of the day
- Verbs, tense

Based on the contents of *IBM Style*, the topics that may be appropriate to a public DITA style manual are:

- Abbreviations
- Active voice or passive voice

- Agent names
- Bits, bytes and words
- Capitalisation of headings
- Captions and legends
- Citing
- Code
- Colons
- Command syntax
- Commands and subcommands
- Company names
- Contractions
- Copyright notices
- Cross-reference punctuation
- Dashes
- Dates
- Definition lists
- Engineering notation
- File names, directory names, and object types
- Footnotes
- Glossaries and definitions
- Graphical User Interface elements
- Headings and subdivisions
- Highlighting
- Indexes
- Interface and device labels
- International currency codes
- Keyboard keys
- Latin terms and abbreviations
- Lists
- Mathematical equations
- Menu instructions and navigation
- Message help
- Message text
- Messages
- Mouse buttons
- Notes in text
- Notices: attention, warning and danger
- Parentheses
- Percentage

- Prepositions with interface elements
- Product names and service names
- Programming statements
- Quotation marks
- References to licensed programs
- Referring to other publications and Web information
- Republishing from other companies
- Revision indicators
- Semicolons
- Tables
- Telephone numbers
- Temperature
- Times of the day
- Trademarks
- Version, release and modification levels
- Web addresses and IP addresses

These specific guide topics may be logically presented in a **Specifics** section with *The DITA Style Guide*, perhaps within a broader **Syntax** usage area.

The content model adopted for the *IBM Style* tends to be:

- Overview
- Details or Usage Advice
  - Translation Considerations
  - Tagging Considerations
- Examples
- See Also

This content model is considered to have a compatible structure to that needed for *The DITA Style Guide*.

Read Me First! A Style Guide for the Computer Industry

> *A style guide that started as an internal document at Sun Microsystems was ultimately released as a commercial publication through Prentice Hall. It was written by technical communicators, and is a comprehensive reference.*

Read Me First! *(Sun Technical Publications, 2003)* is aimed at information technology documentation. It covers a broad range of issues including grammar, typography, indexing, glossary, project management, and even recruitment. It therefore also serves as a handbook for technical communicators in the workplace.

1. Mechanics of Writing
    - Capitalization. Contractions. Gerunds and Participles. Numbers and Numerals. Pronouns. Technical Abbreviations, Acronyms, and Units of Measurement. Punctuation.
2. Constructing Text
    - Headings. Lists. Tables. Code Examples. Error Messages. Cross-References. Endnotes, Footnotes, and Bibliographies. Notes, Cautions, and Tips. Part Dividers. Typographic Conventions. Key Name Conventions.
3. Writing Style
    - Why Is Style Important? Stylistic Principles. Some Basic Elements of Style. Writing for the Reader. Style That Could Offend the Reader. Common Writing Problems to Avoid. Ways to Improve Your Style.
4. Online Writing Style
    - About These Guidelines. Solving Online Writing Problems. Creating an Effective Document Structure. Writing Short, Self-Contained Topics. Constructing Scannable Paragraphs, Headings, and Lists. Preserving Context in Online Documents.
5. Constructing Links
    - About These Guidelines. Where to Place Links. General Linking Strategies. Guidelines for Writing Link Text.
6. Writing Tasks, Procedures and Steps
    - Understanding the Relationship Among Tasks, Procedures, and Steps. Developing Task Information. Writing Procedures. Writing Steps.
7. Writing for an International Audience
    - General Guidelines for Writing for Translation. Cultural and Geographic Sensitivity. Definitions and Word Choice. Grammar and Word Usage. Numbers, Symbols, and Punctuation. Illustrations and Screen Captures.
8. Legal Guidelines

- Copyrights. Trademarks. Third-Party Web Site References. Protection of Proprietary/Confidential Information.

9. Types of Technical Documents

    - What Is a Documentation Set? Documentation Plans. Document Plan. Abstracts. Structure of Manuals. Descriptions of the Manual Parts. Types of Hardware Manuals. Types of Software Manuals. Other Product Documents. Training Documents.

10. Working with an Editor

    - Technical Editor's Role. Editor's Role in Producing Online Documents. Types of Editing. Edit Schedules. Document Submission. Editing Marks. Edit Style Sheet.

11. Working with Illustrations

    - Working With an Illustrator. Illustration Formats, Styles, and Types. Examples of Illustrations. Placing Illustrations. Writing Captions for Illustrations. Writing Callouts for Illustrations. Creating Quality Screen Captures. Creating Leader Lines. Simplifying Online Illustrations.

12. Writing About Graphical User Interfaces

    - Using GUI Terminology. Writing About Windows, Dialog Boxes, and Menus. Writing About the Web.

13. Glossary Guidelines

    - Glossary Content. Terms for an International Audience. When to Include a Glossary. Writing Good Glossary Entries.

14. Indexing

    - What Is an Index? Style and Format. Creating an Index. Refining and Checking an Index. Bad Page and Column Breaks. Checking the Size of the Index. Global Index. Online Index.

15. Appendix - Developing a Publications Department

- Establishment of a Publications Department. Scheduling. Documentation Process. Internationalization and Localization. Online Documentation Considerations. Final Print Production. Post-Production Considerations.

16. Appendix - Checklists and Forms

- Manuscript Tracking Chart. Request for Editing Form. Artwork Request Form. Technical Review Cover Letter. Authorization to Produce Document. Print Specification.

17. Correct Usage of Terms
18. Recommended Reading

Some aspects of the structure of *Read Me First!* would be compatible to the requirements of *The DITA Style Guide* (such as the grouping of information about constructing text, constructing links, and writing tasks, but a great deal of the information is too generalised for a specific DITA style guide. For example, chapters on legal guidelines and working with an editor will not have any DITA-specific implications.

Wired Style

> *Wired is an innovative and respected monthly technology magazine, founded in 1993. The magazine spawned an electronic magazine, HotWired, which later changed name to Wired News. Wired published a style guide in 1996.*

Wired Style - Principles of English Usage in the Digital Age (Hale, 1996), as the title implies, concentrates on English usage, and has little to say on presentational aspects of style. In particular, it focusses on writing in the Digital Age, and provides standards for use of technology terms that are not yet broadly used outside the *digerati*.

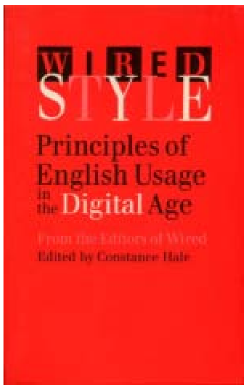

WIRED
STYLE
Principles of English Usage in the Digital Age
From the Editors of Wired
Edited by Constance Hale

Figure 3: Cover of Wired Style - Principles of English Usage in the Digital Age

In some ways, *Wired Style* resembles a dictionary of technology terms, arranged into logical groupings of:

- voice (“Voice is Paramount”)
- historical concepts of cultural literacy (“Be Elite”)
- jargon (“Transcend the Technical”)
- colloquial language (“Capture the Colloquial”)
- creating new terms (“Anticipate the Future”)
- provocative writing (“Screw the Rules”)
- citation (“Grok the Media”)
- international English (“Go Global”)
- acronyms (“Acronyms, FWIW”)
- contentious spelling, formatting and grammatical issues (“Style FAQ”)

Wired Style is a very useful resource, but its structure and content is not a close fit for the requirements of a DITA style guide.

AP Stylebook Online

> *The Associated Press Stylebook has been used by American journalists since its first edition in 1953. It has continually evolved, and is now made available as a printed book and as a Web resource.*

Over the years, the AP Stylebook *(The Associated Press, 2009)* has grown from 62 pages to over 400 pages, and is sometimes referred to as “the journalist's bible”.

Figure 4: Cover of the AP Stylebook

The journalistic focus of the *Stylebook* is evident from its structure, where the bulk of the content is organised into an alphabetical dictionary of usage rules, with little formatting or presentational content. The alphabetical list of usage rules is not dissimilar to the approach used in the *Microsoft Manual of Style for Technical Publications*.

After the usage rules, the *Stylebook* is broken down into chapters detailing the requirements of the sporting and business specialised styles of journalism (also arranged as an alphabetical list of usage rules), then a chapter on punctuation (again, organised alphabetically), and followed by a series of process-related chapters.

The Web version of the *Stylebook* allows subscriber to add personal notes and rule entries, through a feature known as “My Custom Stylebook”.

AP Stylebook Contents

- Stylebook Key
- Alphabetical Usage Guide
- Sports Guidelines
    - Alphabetical Usage Guide
- Business Guidelines
    - Alphabetical Usage Guide
- Guide to Punctuation
    - Alphabetical Usage Guide
- Briefing on Media Law
- Graphics
- Photo Captions
- Editing Marks
- Filing Practices
- Filing the Wire
- Bibliography
- About the AP

The alphabetical arrangement of content is an interesting approach, and works particularly effectively when used as a look-up reference. This architecture also suits a Web-based reference.

ISO/IEC 26514: Requirements for Designers and Developers of User Documentation

> *The ISO/IEC 26514 Systems and Software Engineering - Requirements for Designers and Developers of User Documentation standard was released in 2008, replacing the earlier 2004 standard ISO/IEC 18019 Systems and Software Engineering - Guidelines for the Design and Preparation of User Documentation for Application Software. Although not widely used, its status as an international standard makes it an authoritative source.*

As an international standard, *ISO/IEC 26514 Systems and Software Engineering - Requirements for Designers and Developers of User Documentation (2008)* has some similarity in function with a supra-organisational style manual such as *The DITA Style Guide*. The *ISO/IEC 26514* standard covers the breadth of the documentation life cycle, but dedicates most attention to the processes involved in the documentation project itself.

The standard is broken up into 12 sections and 7 annexes as follows:

- Introductory Sections (10 pages)
  - Scope
  - Conformance
  - Normative references
  - Terms and definitions
- Main Sections (77 pages)
  - User documentation process
  - Project requirements, objectives, and constraints
  - Analysis and design
  - Development and review
  - Production
  - Structure of documentation
  - Information content of user documentation
  - Presentational format of documentation
- Annexes (53 pages)
  - User documentation style guide content
  - Writing style and techniques
  - User documentation style for translation and localization
  - Design, development, and production of printed information
  - Checklists for user documentation
  - Checklists for the documentation process
  - Checklists for documentation products

The majority of the main sections are entirely generic in nature, in that they relate to user documentation very broadly and are independent on any particular writing methodology or technology such as DITA.

Section 10 covers information types and document structure, but discusses these only in relation to audience analysis and organising of information.

Annex A recommends the use of documentation project style guides covering writing style, language, spelling, and grammar and usage.

Annex B provides some writing style guidelines, covering the topics of:

- writing style in general
- style for sentences
    - hanging participles
    - tautologies and redundant phrases
    - articles and pronouns
    - positive and negative constructions
    - style for conditions
    - active and passive voice
    - tenses
    - single and plural verbs
    - punctuation
- style for paragraphs
    - hyphenation
    - infinitives
    - capital letters and lower case
    - anthropomorphisms
    - analogies and metaphors
- style for quick reference information
- style for installation instructions
- style for tutorial and task instructions
- style for describing user interface elements
- style for descriptions and explanations
- style for onscreen information
- style for lists

Some of these topics will most likely overlap with topics in *The DITA Style Guide*, so *The DITA Style Guide* will need to be reviewed to ensure that its recommendations do not conflict (without due cause) with the requirements of ISO/IEC 26514, and that the standard forms part of the rationale for rules in *The DITA Style Guide*. Consideration should also be given to cross-referencing rules in *The DITA Style Guide* with related ISO/IEC 26514 clauses.

Microsoft Manual of Style for Technical Publications

The Microsoft Manual of Style for Technical Publications is widely used by technical communicators when documenting software applications, particularly Windows applications.

The *Microsoft Manual of Style for Technical Publications (2004)* was first published in paperback form by Microsoft Press in 1995, and the most recent third edition was published in 2004. The third edition was also made available in electronic form.

The primary aim of the style manual, as described in the introduction, is:

...to help Microsoft writers and editors maintain consistency within and across products. MSTP is not a set of rules, but a set of guidelines that have been discussed and reviewed by experienced writers and editors across the company.

Microsoft Manual of Style, 2004

The book is made up of two parts: *General Topics* and *Usage Dictionary*.

The General Topics part provides writing and formatting conventions and advice for a broad range of subjects, spanning authorial style, legal issues, grammar and punctuation, attributions, and style troubleshooting. It is made up of 11 sections:

- 1. Documenting the User Interface
- 2. Content Formatting and Layout
- 3. Global Content
- 4. Content for Software Developers
- 5. Web Content
- 6. Legal Content and Front Matter
- 7. Indexing and Attributing
- 8. Tone and Rhetoric
- 9. Common Style Problems.
- 10. Grammatical Elements
- 11. Punctuation

The Usage Dictionary part is an alphabetical list of terms, with terms explained and cross-referenced. This part is intended as a quick reference source, particularly for word usage and abbreviations.

The content of the *Microsoft Manual of Style for Technical Publications* is likely to have few overlaps with the content required in a DITA style guide, so the two references are likely to be used together in a DITA authoring team working to the *Microsoft Manual of Style for Technical Publications*.

As summarised in a review of the *Microsoft Manual of Style for Technical Publications (DeLoach, 1998)*, “it's an invaluable resource for anyone documenting Windows software”.

Creating a Style Guide (Pro Gradu Thesis)

> *The Masters thesis of Laura Hyttinen of University of Tampere (Finland) described a case study of the development of an organisational house style guide.*

In analysing the development of a house style guide for Sandvik Mining and Construction, *Hyttinen (2008)* reported the structure of the first version as comprising:

- using and processing pictures
- organising information
- writing body text
- writing procedures
- using headings
- using tables and lists
- creating the table of contents
- reviewing the instructions
- getting feedback

The majority of these sections would not be applicable to a DITA writing style guide, but the structure does suggest that *The DITA Style Guide* should include sections on:

- writing concept, task and reference topics
- working with block elements
- working with inline elements

Writing guides

A review of a small sample of English writing guides was conducted.

The purpose of the review was to confirm that information in such guides was outside the scope of the Artefact, intended only to cover issues specific to semantic authoring. The review focussed on identifying the high level structure of the information, and special characteristics of the publication.

The Elements of International English Style

> *The Elements of International English Style is probably not a "style guide" proper, but is a guide to English writing for a global audience.*

The Elements of International English Style (*Weiss, 2005*), provides instructions and guidelines for writing in the English language for an international audience. As such, it isn't designed to provide the rules normally found in a conventional *style guide*.

The book has the following major sections.

- Preface
- Chapter 1 The Language of Global Business Is International English
- Chapter 2 Principles of Simplicity
- Chapter 3 Principles of Clarity
- Chapter 4 Reducing Burdens
- Chapter 5 Writing for Translation
- Chapter 6 Principles of Correspondence
- Chapter 7 Principles of Cultural Adaptation

In terms of separation of content and form, the book covers very few form issues, being focussed on English usage and writing style. It covers different ground to that of *The DITA Style Guide*, and is likely to be used as an educational resource rather than a day-to-day reference.

Basic English: a General Introduction with Rules and Grammar

> *Controlled vocabularies are used as a writing methodology within some organisations, particularly those where writers or readers are not native speakers of the publication language. The idea of controlled vocabularies was developed by Charles Ogden, who published his work in Basic English: a General Introduction with Rules and Grammar.*

In the 1930s, Charles Ogden developed the idea of a global English language, with a controlled vocabulary and simplified spelling. Ogden suggested the English vocabulary could be reduced to 850 words. (There are approximately 25,000 words in the Oxford English Dictionary.)

Ogden's ideas inspired later *controlled vocabulary* forms of English, including Simplified English, Simplified Technical English, Caterpillar English, IBM Easy English, and General Motors CASL.

This work *(Ogden, 1930)* does not have much relevance to *The DITA Style Guide*, but would be a useful reference for organisations also working with controlled vocabularies.

Web style guides

A review of a sample of Web style guides (style guides designed primarily for Web sites and Web publications) was conducted. The style guides included those for commercial, government and academic Web publications.

The review focussed on identifying the high level structure of the information, and special characteristics of the publication. As the Artefact is intended to be donated as an open source project in a Wiki format in the medium term, the review of open source style guides such as Wikipedia's helped to form the feasibility of this idea.

Wiki-based style guides provided comprehensive sets of rules, and typically provided features to allow contributors to explain their rationale or to discuss contentious issues. When the Artefact a Wiki resource, similar features to expose the rationale behind decisions will be necessary. Even though rationale may not be published in *The DITA Style Guide*, it is important that it is documented and retained within the DITA source of the Artefact.

Wikipedia Manual of Style

> *It is quite a challenge to have a consistent mechanical style for a collaboratively written work such as the Wikipedia online encyclopaedia. One of the tools used to encourage this consistency is the Wikipedia Manual of Style, itself written collaboratively as a Wiki.*

The *Wikipedia Manual of Style* is a style guide for Wikipedia articles. The Manual of Style addresses issues of writing style, presentational formatting, and culture. Associated *talk pages* allow authors to discuss usage and other issues not specifically addressed in the Manual of Style. Although the technical architecture of Wikipedia features a partial separation of content and form, the separation is not as complete as for DITA.

Of particular interest is the way the Wikipedia Manual of Style attempts to provide rules that can be cited by editors and authors in a multiple-author environment, which is where there is a close parallel to a typical DITA topic-based, multiple-author implementation.

Wikipedia Manual of Style Contents

Retrieved 30 March 2009

1. General principles
   - Internal consistency
   - Stability of articles
2. Article titles, headings, and sections
   - Article titles
   - Section headings

- Main article link
- Section management

3. Capital letters
    - Titles of people
    - Religions, deities, philosophies, doctrines, and their adherents
    - Calendar items
    - Animals, plants, and other organisms
    - Celestial bodies
    - Directions and regions
    - Institutions
4. Acronyms and abbreviations
5. Italics
6. Non-breaking spaces
7. Quotations
8. Punctuation
    - Apostrophes
    - Quotation marks
    - Brackets and parentheses
    - Ellipses
    - Commas
    - Colons
    - Semicolons
    - Hyphens
    - Dashes
    - Slashes
    - Punctuation at the end of a sentence
    - Punctuation and inline citations
    - Punctuation after formulae
9. Geographical items
10. Chronological items
    - Precise language for dates
    - Times
    - Dates
    - Longer periods
11. Numbers
    - Numbers as figures or words
    - Large numbers
    - Decimal points
    - Percentages
12. Units of measurement
    - Which system to use
    - Conversions
    - Unit symbols and abbreviations

- Clarify ambiguous units
- Unnecessary vagueness

13. Currencies
14. Common mathematical symbols
15. Simple tabulation
16. Grammar

- Possessives
- First-person pronouns
- Second-person pronouns
- Contested vocabulary
- Contractions
- Instructional and presumptuous language
- Subset terms
- Plurals
- Ampersand
- National varieties of English
- Foreign terms
- Identity
- Gender-neutral language

17. Images
18. Captions

- Usage
- Formatting

19. Bulleted and numbered lists
20. Links

- Wikilinks
- External links

21. Miscellaneous

- Keep mark-up simple
- Formatting issues
- Scrolling lists
- Invisible comments
- Pronunciation

MythTV Manual of Style

> *MythTV is an open source software project for personal video recorders. Its style manual was unashamedly based on Wikipedia's Manual of Style.*

The *MythTV Manual of Style* is a style guide for information and documentation for the open source MythTV software, which is being developed collaboratively. The style guide is based largely on the *Wikipedia Manual of Style* (see page 41). The MythTV Manual of Style therefore has an approach similar to Wikipedia, focussing on issues of writing style and presentational formatting.

1. Article titles
   - Singular vs. Plural
   - Upper case/lower case/mixed case
2. Headings
3. Capital letters
   - Titles
   - Calendar items
   - As Of dates and versions
4. Italics
   - Titles
   - Words as words
   - Quotations
5. Punctuation
   - Quotation marks
     - Look of quotation marks and apostrophes
   - Colons
   - Contractions
6. Acronyms and abbreviations
7. Simple tabulation
8. Usage and spelling
   - Usage
   - Avoid self-referential pronouns
   - Avoid the second person
9. Pictures

10. Captions
11. Bulleted items
12. Wiki-linking
13. External linking
14. Miscellaneous notes

    - When all else fails
    - Keep mark-up simple
    - Formatting issues
    - Make comments invisible
    - Legibility

ABC Radio National Online Style

> *Like many journalism style guides, the ABC Radio National online style guide is primarily concerned with consistent term usage. However, the summary topic provides a great synopsis of the macro editorial style rules.*

The main section of the *ABC Radio National online style* guide provides an alphabetical list of words and phrases, with explanations of their correct use. In some cases, consistent spelling is the aim of the style rule. In other cases, rationale for the recommended use, and/or an extended definition of the term, is provided.

For example, “pejorative” is explained as “deprecatory ... nothing to do with perjury”.

Some terms in the alphabetical section are guidelines for writing, such as the entry for “dangling participles”, which is explained as:

> *Dangling participles, or dangling modifiers, become a problem if a reader has to pause to work out how a sentence should be understood. For example, 'Driving up to the house, her dog always barks loudly.' That split-second hesitation while you work out what's going on can be avoided by writing 'Her dog always barks loudly when she drives up to the house.' We still don't know if the dog's in the car or in the house, but at least it's not driving.*

A short topic entitled “At-a-glance online style points” provides a succinct summary of macro-level (that is, document level, rather than sentence level) editorial style rules. Specifically, these rules cover:

- capitalisation in headings

- capitalisation in body text
- abbreviations and acronyms
- titles of published works
- quotations

The online style does not cover formatting style rules, with the exception of use of italics and inverted commas.

It does not cover any issues relating to semantic authoring, so is unlikely to be a useful resource during the development of the Artefact.

Monash University Web Style Guide

> *Monash University publish a Web Style Guide to manage the quality of contributions to the University's various official public and internal Web sites.*

The *Monash University Web Style Guide* is a Style Guide for contributors to the University's Web sites. In addition to the specific Monash University rules, the Style Guide contains links to many external style guide resources for generic Web writing guidelines. The bulk of the Monash-created content is focussed on design and branding guidelines.

Monash University Web Style Guide (Version 3) Table of Contents

- Designing for our users
- Quality, usability, and accessibility
    - Quality assurance
    - Creating usable websites
    - Accessibility standards
- Legal requirements
    - Australian Government legislation for international students - content requirements
    - Copyright
    - Disability Discrimination Act
    - University privacy policy
    - Caution and Australian Business Number (Australian campuses only)
- Branding and visual identity
    - Masterbrand websites
    - Faculty websites
    - Commercial entities
    - Other entities
    - Dual-branded sites
    - Use of other logos/brands
    - Personal pages
    - Use of images
- Navigation and site structure

    - Website structure
    - Website navigation
    - Accessible navigation
    - Navigation labels
    - Text links
- Creating web pages
    - Page layout
    - Page titles
    - File and directory names
- Content (text and images)
    - Writing for the web
    - Images and multimedia
    - Making images and multimedia accessible
    - Publishing in formats other than HTML
    - Using HTML tags properly
- Tables and forms
- Use of colour and styles
    - Using colour properly
    - Using fonts
    - Applying styles to content
- Metadata and search
    - Metadata and search engines
    - Monash-specific metadata
    - Filtering intranet pages from Monash search engine results
    - Guidelines on use of specific metadata elements
    - Metadata and the content management system
- Technical standards and implementation
    - HTML mark-up
    - Style sheets (CSS)
    - Error messages
    - Technical implementation of the web templates
- Web templates

This style guide is very focussed on Web publishing, and although the section on metadata has some relevance to semantic authoring in DITA, much of the content is incompatible with the requirements of a DITA style guide.

Web Style Guide

> *The Web Style Guide takes a user-centred design approach, and focusses more on the process of Web site and content creation, rather than prescribing rules at a lower paragraph or word level.*

Very little of the *Web Style Guide* *(Lynch and Horton, 2009)* relates to writing content; in fact, in a recommendation of members of a Web development team, a technical writer or content developer is not mentioned. Responsibility for content is deemed to be the role of the Site editor, the last of the eight key roles. It follows that much of the *Web Style Guide* relates to site organisation, information architecture, and presentational style.

Web Style Guide (Lynch and Horton, 2009) Table of Contents

- 1 Process
    - The site development team
    - Sidebar: Web teams
    - Initial planning
    - A list of reminders
    - Types of web sites and documents
    - The site development process
    - Developing a project charter
    - General advice about running web projects
- 2 Universal Usability
    - A basis for universal usability
    - Sidebar: Universal design principles
    - Universal usability guidelines
    - Universal usability in the design process
    - Sidebar: The development cycle
- 3 Information Architecture
    - Organizing Your Information
    - Site Structure
    - Presenting Information Architecture
- 4 Interface Design
    - Navigation and wayfinding
    - Interface design
    - Information design
    - The enterprise interface
- 5 Site Structure
    - Semantic content mark-up
    - Site File Structure
    - Search Engine Optimization
- 6 Page Structure
    - Site design in context
    - Page structure and site design
    - Page templates
- 7 Page Design

  - Document design
  - Visual design
  - Sidebar: Visual design principles
  - Page frameworks
  - Page width and line length
  - Design grids for web pages

- 8 Typography
  - Characteristics of type on the web
  - Legibility
  - Typefaces
  - Sidebar: Type for comfortable reading
  - Emphasis
  - Display typography with graphics
  - Sidebar: Signal to noise ratio

- 9 Editorial Style
  - Structuring your prose
  - Online style
  - Sidebar: Rhetoric and web design
  - Test formatting for web documents
  - Links

- 10 Forms & Applications
  - Technologies that support interaction
  - Designing web applications
  - The design process

- 11 Graphics
  - Graphics as content
  - Sidebar: The origins of information graphics
  - Characteristics of web graphics
  - Graphics file formats
  - Imaging strategies
  - Images on the screen
  - Graphics mark-up
  - Sidebar: Color terminology

- 12 Multimedia
  - Considerations for multimedia
  - Web multimedia strategies
  - Preparing multimedia
  - Design and multimedia

- References
- Abbreviations

This style guide is very focussed on Web publishing, and is largely incompatible with the requirements of a DITA style guide.

W3C Quality Assurance Web Site

> *The World Wide Web Consortium's Quality Assurance Interest Group maintain a Web site aimed at improving the quality of HTML and CSS standards implementation.*

The *W3C Quality Assurance Web Site (n.d.)* is a collection of topics and documents relating to the best practice implementation of HTML, CSS and other W3C standards. One of the principles described is the separation of semantic and presentational mark-up, addressed by a topic on using semantic CSS class names, and by an online presentation about how to separate semantic and presentational mark-up.

The site includes a related *Using Style Sheets* style guide *(Berners-Lee, 2002)*, which is a simplistic, upper level generic guide to using cascading style sheets effectively.

Rather than being a comprehensive and authoritative style manual, the W3C Quality Assurance Web site is a small repository of collected snippets of useful information. Within the library of assorted articles, documents, external references and tutorials, there is a user-centred blog, where issues of quality and standards can be queried and commented on by site users.

The site tends to focus on marketing the use of standards for Web site development, by providing details of benefits of a standards-based approach, and advice on how to specify or change to standards-based sites. Unlike Web development, where technically non-standard HTML code can still be practically used, DITA can only be technically valid. *The DITA Style Guide* may, however, incorporate information aimed at marketing the principles of best practice approaches. Best practice is different to technically valid; many alternative techniques may be technically permissible, but only one will be the better technique.

Apache Derby Writing Guidelines

> *Although quite brief, the Apache Derby project Writing Guidelines is a rare example of style guidelines for authors working in a DITA environment.*

Apache Derby is an open source relational database application, developed around Java and SQL. Derby is a project of the Apache Software Foundation, a non-profit organisation that fosters numerous innovative open source projects. Derby project documentation is collaboratively developed by contributors to the project, using DITA as the source format.

The challenge of collaboratively developing documentation in DITA in an open source environment with geographically separated authors, all of whom are volunteers, is partly addressed through the *Apache Derby Writing Guidelines (n.d.)* online document. It is rare to find a style guide for authoring in DITA in the public domain, so the Apache Derby Writing Guidelines, although brief, is particularly instructive.

The Guidelines comprise approximately eight short core topics, with approximately 14 related supporting topics. The core topics are:

- New topics: Choosing the correct topic type
- New topics: Using a template to create a new topic
- New topics: Selecting the best filename
- Understanding DITA tags
- Tagging examples - structural tags
- Tagging examples - textual tags
- Indexing guidelines
- Customizing tables in the Derby documentation

Rules in the Writing Guidelines should be cross-checked against *The DITA Style Guide* to ensure there is no logic or rationale gap between the two.

Derby documentation uses the DITA base content model, which is also what *The DITA Style Guide* will cover. (It will not cover specialised information types.) However, the Derby Writing Guidelines are often low level XML coding instructions (such as how to manually mark up a table). *The DITA Style Guide* will assume the use of a DITA authoring tool, so XML coding instructions are less important. Readers of *The DITA Style Guide* will also be generally far less technically competent than Derby document contributors.

References - Literature Review

- Apache Derby: Writing guidelines. (n.d.). Apache Derby. Retrieved October 10, 2009, from http://db.apache.org/derby/manuals/guidelines.html.
- Berners-Lee, T. (2002, April 17). Using Style Sheets - Style Guide. World Wide Web Consortium (W3C). Retrieved October 9, 2009, from http://www.w3.org/Provider/Style/StyleSheets.html.
- DeLoach, S. (1998). Microsoft Manual of Style (Review). Technical Communication: Journal of the Society for Technical Communication, 55(3), 285-306.
- Hale, C. (1996). Wired Style - Principles of English Usage in the Digital Age (1st ed.). San Francisco: Publishers Group West.

- Hyttinen, L. (2008, May). Creating a Style Guide - A Case Study of Sandvik Mining and Construction Hard Rock Mining Project. Pro Gradu Thesis, University of Tampere. Retrieved from http://tutkielmat.uta.fi/pdf/gradu02978.pdf
- IBM Style. (2009). IBM UT Style and Word Usage Council.
- International Standard ISO/IEC 26514: Systems and software engineering - Requirements for designers and developers of user documentation. (2008, June 15). International Standards Organisation.
- Lockwood, K. (2005). Style: The essential guide for journalists and professional writers (3rd ed.). Melbourne: News Custom Publishing.
- Lynch, P. L., & Horton, S. (2009). Web Style Guide: Basic Design Principles for Creating Web Sites, Web Style Guide (3rd ed.). New Haven, USA: Yale University Press. Retrieved from http://webstyleguide.com/index.html
- Lockwood, K. (2005). Style: The essential guide for journalists and professional writers (3rd ed.). Melbourne: News Custom Publishing.
- Microsoft® Manual of Style for Technical Publications, Third Edition, 2004, ISBN 0-7356-1746-5
- MythTV: Manual of Style. (n.d.). MythTV. Retrieved January 27, 2011, from http://www.mythtv.org/wiki/MythTV:Manual_of_Style.
- Ogden, C. (1930). Basic English: A General Introduction with Rules and Grammar. London: Paul Treber and Company.
- Radio National style guide. (2004, March 1). Australian Broadcasting Corporation. Commercial, . Retrieved January 27, 2011, from http://style.radionational.net.au/glossary.
- Snooks and Co. (2002). Style manual: for authors, editors and printers (6th ed.). Brisbane: Wiley Australia.
- Sun Technical Publications. (2003). Read Me First! A Style Guide for the Computer Industry (2nd ed.). Mountain View, California: Prentice Hall PTR.
- The Associated Press. (2009). The Associated Press Stylebook 2009 Fully Revised and Updated (43rd ed.). New York: Basic Books. Retrieved from http://www.apstylebook.com/
- The Chicago Manual of Style. (1982). (13th ed.). Chicago: University of Chicago Press.

- The Newspaper Style Manual... (1999, December 4). Lingua Franca. Australian Broadcasting Corporation. Retrieved January 26, 2011, from http://www.abc.net.au/rn/arts/ling/stories/s70685.htm.
- Wallace, D., & Hughes, J. (1995). Style Book - A Guide for New Zealand Writers and Editors (5th ed.). Wellington: GP Publications.
- Web Style Guide (Version 4). (n.d.). Monash University. Academic, . Retrieved May 1, 2009, from http://www.monash.edu.au/staff/web/.
- Weiss, E. H. (2005). The Elements of International English Style : A Guide to Writing Letters, Reports, Technical Documents, and Internet Pages for a Global Audience. New York: M.E. Sharpe.
- Wikipedia: Manual of Style. (n.d.). Wikipedia. Retrieved January 27, 2011, from http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style.

Chapter

3

Research Approach

Topics:

The primary product of the research project was intended to be a published work (an Artefact) that provided a useful contribution to the DITA community. A secondary product was to be an Exegesis documenting my discoveries and reflections along the process of writing both Artefact and Exegesis within the DITA approach.

The research project is not a traditional PhD dissertation thesis, but a *PhD by Artefact and Exegesis*.

The Artefact was published commercially in March 2011 as *The DITA Style Guide: Best Practices for Authors*.

I adopted an *action research* approach to the project, which was ideally suited to a project that aimed to develop practical knowledge from participatory practice, and contribute to the further enhancement of the resources available to the DITA community.

PhD by Artefact and Exegesis

The outcome of the research project is an Artefact accompanied by this Exegesis.

Rather than a traditional dissertation thesis, the primary product of the research described in this Exegesis is an Artefact (*The DITA Style Guide: Best Practices for Authors*), published commercially in March 2011. The Artefact itself was written using DITA, as was this Exegesis.

The role of the Exegesis in a PhD in writing accompanied by an Exegesis, was summarised by *Arnold (2005)* as:

> *The exegesis is not a critique of the work, but sits alongside it. Some see it as a more academic way of accompanying, writing about, exploring further and in a different way ideas in the non-academic writing, thus supporting the process of the non-academic writing which is a more public piece with a wider and/or different audience in mind. However, it is also evolving into a more reflective piece of writing in which the contribution to knowledge becomes insights into the individual creative process with reference to ideas in the relevant literature.*

The purpose of this Exegesis, therefore, is to document my reflection on the process of writing the Artefact (and Exegesis) within the DITA approach. In particular, the Exegesis reflects on the issues and dilemmas I faced in writing *The DITA Style Guide* (or any other DITA-based publication), and the decision-making processes I employed. These are matters not addressed in the Artefact itself, and constitute an original and unpublished component of the PhD project.

Summary of approach to source materials and reference works

In the process of creating The DITA Style Guide, I drew upon the existent authoritative style guides and related standards, and the experience of early DITA adopters.

What the term *best practice* actually means is not universally understood or agreed. Researching the subject of what constitutes best practice in DITA is made more difficult by the difficulty in finding agreed practices in this field, let alone best practice. Something as simple as whether paragraphs should exist within list items is not clear-cut.

The product of this research is a style manual for technical communicators working in a DITA environment. DITA represents a fundamental shift in the approach to writing and publishing, and to a certain extent, the carefully accumulated practice of hundreds of years of publication projects has to be reviewed and redefined.

There are two groups of resources that can be drawn upon to determine what constitutes DITA best practice:

- existing style manuals
- the experience of early adopters of the DITA methodology.

The recommendations and guidelines in existing style manuals were filtered to separate the presentational rules from the writing rules, and used as a starting point for *The DITA Style Guide*. The relevant ISO standards for documentation, and the DITA Language Reference, were also drawn upon as source materials.

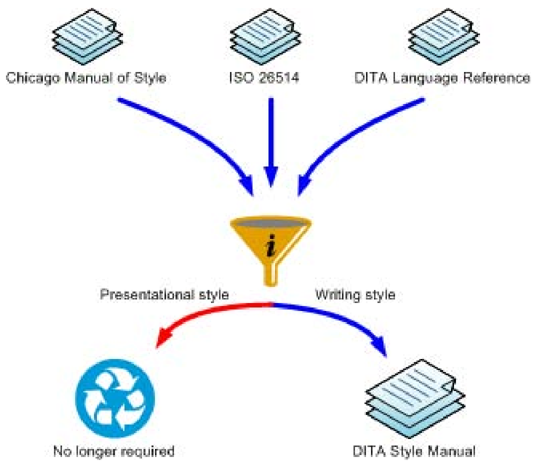


Figure 5: Reference documents used as source materials

Action Research

I used an Action Research approach for the project, and engaged with the DITA community at different levels in different roles.

The DITA Style Guide was developed using an Action Research methodology.

A definition of *action research* proposed by *Reason and Bradbury (2001, p. 1)* resonates with the approach taken in the development of the Artefact:

Action research is a participatory, democratic process concerned with developing practical knowing in the pursuit of worthwhile human purposes, grounded in a participatory worldview.... It seeks to bring together action and reflection, theory and practice, in participation with others in the pursuit of *practical solutions to issues of pressing concern to people, and more generally the flourishing of individual persons and communities.*

I identified authoring best practices and the specific requirements of a DITA style manual through participation in three areas:

- development of a variety of documents of different sizes in the DITA framework
- analysis of the requirements of other DITA adopters expressed in the Yahoo! DITA Users Group
- work within the OASIS DITA Technical Committee (the organisational entity responsible for the development and maintenance of the DITA standard) and the OASIS DITA Adoption Technical Committee (the entity responsible for promoting the adoption of DITA).

My extensive industry experience as a practising technical communicator, as a manager of technical communication projects, and as a user (and author) of traditional style manuals, was drawn upon during the research project.

Reason and Bradbury (2001, p. 2) suggested that the primary purpose of action research is “to produce practical knowledge that is useful to people in the everyday conduct of their lives”. The Artefact is intended to be a resource containing such practical knowledge that is useful to the technical communication professional community in their everyday working lives.

Development of DITA documents as part of Action Research

> *Some of the challenges encountered by technical communicators working on DITA-based documentation projects were discovered by my own work in developing DITA-based documentation.*

I wrote a number of documents entirely within the DITA framework as part of the action research approach. Specifically, he developed the following documents or document parts:

- Help system for the WinANT Echidna software
- The Artefact (*The DITA Style Guide*) itself
- Topics within the OASIS DITA Help Subcommittee's *DITA Help Technologies Guide*
- The TACTICS Web site (unpublished)
- OASIS Whitepaper *(Improved Glossary and Terminology Handling Features in DITA 1.2)*
- Two published academic conference papers (*DITA and the Challenges of Single-Source Article Publishing* and *Context-Agnostic Writing in Modular Documents*)
- Magazine articles (including *What if Readers Can't Read?*, *Writing to STOP*, *Is Rhetorical Writing Our New Destiny?*, and *Can you explain that again from the beginning? What is DITA?*).

The range of projects provided a good sample of not only typical documentation topics, but also of the different range of deliverable reading formats. The practical experience of working as a DITA author provided many insights into how best practice guidelines would make authoring easier.

Further, I was involved in a number of consultancy projects for commercial clients. One project involved my designing a file and folder naming convention for DITA topic, map, and supporting files.

Analysis of Yahoo! DITA Users Group

A detailed analysis of a sample of 212 message threads from the Yahoo! DITA Users Group revealed the types of questions DITA authors need answered in order to write effectively in a DITA environment.

There are a number of online communities for DITA authors, including:

- Bob Doyle's DITA Users group
- LinkedIn DITA Awareness Group
- Yahoo! Adobe FrameMaker DITA Development Group
- OASIS DITA Community (dita.xml.org)
- DITA Facebook Community
- Yahoo! DITA Users Group

The most active of these groups, measured by the number of messages exchanged, is the Yahoo! DITA Users Group.

The Yahoo! DITA Users Group was established in June 2004, and by the end of 2010, 21,000 messages had been exchanged amongst members.

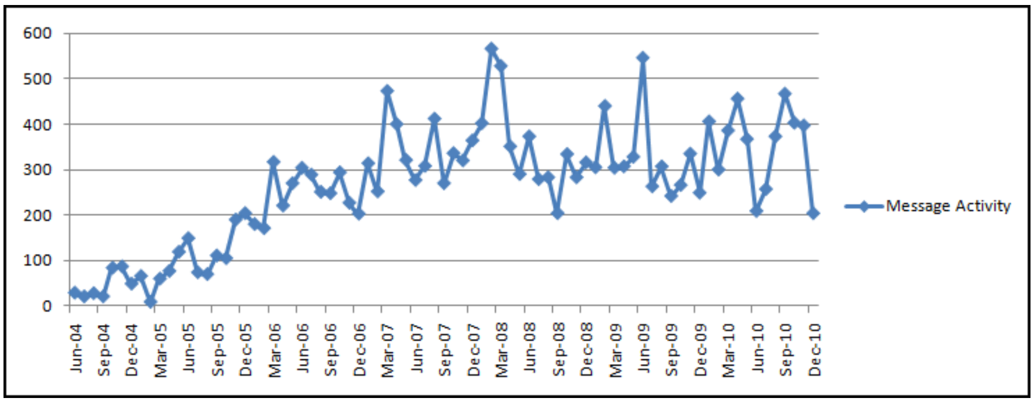


Figure 6: Yahoo! DITA Users Group activity (2004 to 2010)

Most messages exchanged were found to be either questions to other members of the community, or answers to those questions. The type of questions asked provided a good insight into the practical problems typically encountered by DITA authors.

I conducted an analysis of the questions asked to categorise the nature of the information sought, and to obtain an insight into what would be require in a DITA style manual.

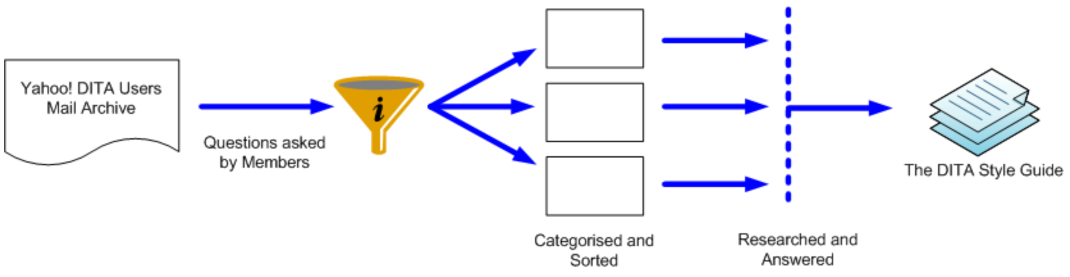


Figure 7: Schematic showing use of Yahoo! DITA Users messages

A sample of 211 messages posing questions was analysed, and the subject matter of the questions was identified, based on the subject line. (A description of the data collection method is provided in *Data collection using Yahoo! Group Downloader* (see page 60).)

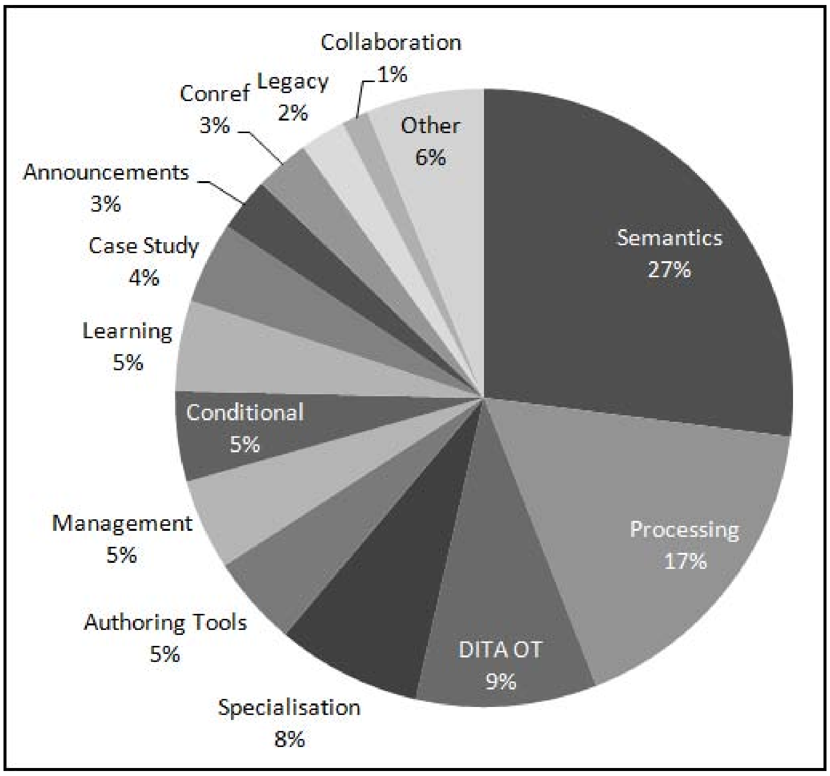


Figure 8: Pie Chart: Breakdown of message subjects

Table 1: Categories Message Subjects of Sample

Category	Message Count
Semantics	57
Processing	36
DITA OT	20
Specialisation	16
Authoring Tools	10
Management	10
Conditional	10
Learning	10
Case Study	9
Announcements	6
Conref	6
Legacy	5
Collaboration	3
Other	13

Data collection using Yahoo! Group Downloader

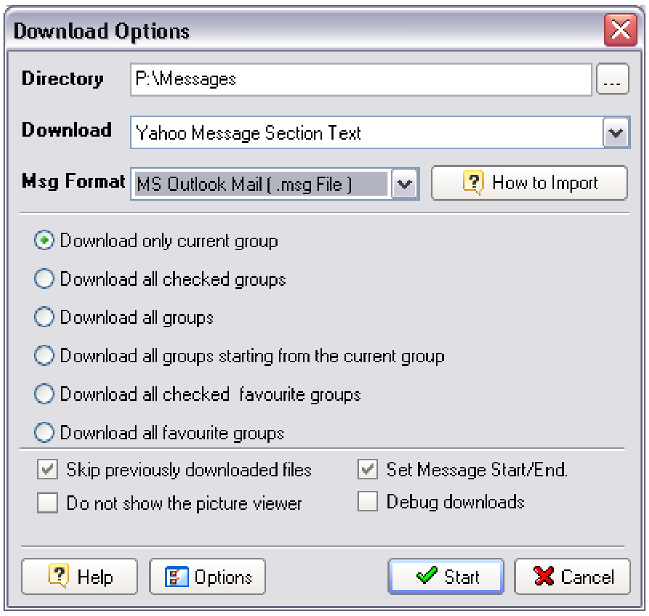
> *Over 13,000 messages had been posted to the Yahoo! DITA Users Group by April 2009. All these messages were downloaded using a software tool, and their subject lines then extracted. The subject matter was then analysed and categorised.*

The community of DITA authors around the world is not as fragmented as many other similar communities. There are only two large online DITA groups: Yahoo! DITA Users, and Bob Doyle's DITA Users. The latter group is semi-commercial, and does not have an active mailing list. The majority of the members of Bob Doyle's group are also members of the Yahoo! group. In May 2009, the Yahoo! group had 2,382 members.

Membership in the Yahoo! group has grown, since its formation in June 2004, at a rate of around 40 users per month. The questions asked on the list are a good representative sample of the questions that new DITA users need answered in order to more effectively use the methodology. The Yahoo! group is therefore a good source for researching what questions a DITA Style Manual needs to answer.

I conducted an analysis of a sample of messages from the Yahoo! Group.

All messages from June 2004 to March 2009 were downloaded in Outlook `.msg` format to a repository (refer to *Figure 9: guru's Yahoo! Group Downloader - Options Window* (see page 61)), using a software tool called guru's Yahoo! Group Downloader 3.0™ (refer to *Figure 10: guru's Yahoo! Group Downloader - Status Window* (see page 62)).


Download Options

Directory: P:\Messages

Download: Yahoo Message Section Text

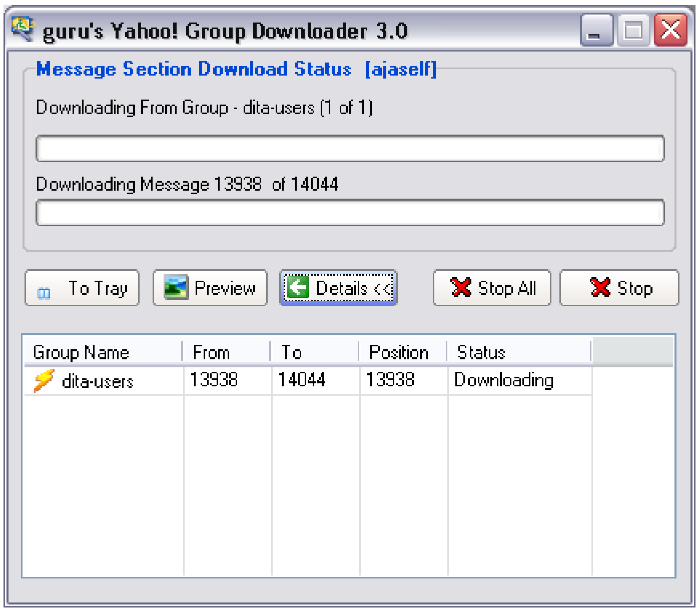
Msg Format: MS Outlook Mail (.msg File)

[✓ Download only current group]
Download all checked groups
Download all groups
Download all groups starting from the current group
Download all checked favourite groups
Download all favourite groups

[✓ Skip previously downloaded files]
[✓ Set Message Start/End.]
Do not show the picture viewer
Debug downloads

Figure 9: guru's Yahoo! Group Downloader - Options Window

The message files were stored on a network share, with each message in its own `.msg` file.


guru's Yahoo! Group Downloader 3.0

Message Section Download Status [ajaself]

Downloading From Group - dita-users (1 of 1)

Downloading Message 13938 of 14044

Group Name	From	To	Position	Status
dita-users	13938	14044	13938	Downloading

Figure 10: guru's Yahoo! Group Downloader - Status Window

After 13,500 messages were downloaded, representing four and a half years of messages, the messages were imported into the Windows Mail program. The messages were then transferred to Microsoft Outlook®, and then exported to Microsoft Access® format. Approximately 14%[1] of the messages could not be imported. Due to the way the messages were stored in Yahoo!, the saved messages did not have a consistent format. Therefore, the message Subject fields in the overwhelming majority[2] of the message subject fields were empty, and the subject appeared in the message body. This made categorisation of those messages more difficult.

The 1,239 messages with subject lines included many responses to a previous mail, and responses (all with a subject line prefix of `Re:`) were filtered out. This left 329 messages with a *start of thread* subject line. On the basis that those messages with subject lines represented a random sample of all messages, the assumption was made that for every one question message, an average of three[3] responses were made. This figure was extrapolated to deduce that of the 13,500 original messages, 3580 were probably questions (*starts of thread*). The sample of 329 messages with *start-of-thread* subject lines therefore represented 9% of all questions asked over the four and a half year period, which was large enough to be representative.

I categorised the questions through an additional Category field in the Access table, and then used Queries to analyse the information. The results of the analysis were summarised into graphs and tables.

[1] 1,884 messages from 13,500
[2] 10,377 from 11,616
[3] 2.77

The questions were continually referred to during the development of *The DITA Style Guide* to ensure that the document provided answers.

Work with OASIS Technical Committees

> *I is an active participant in the OASIS standards body which is responsible for the ongoing development of the DITA standard.*

The *Organization for the Advancement of Structured Information Standards (OASIS)* is a not-for-profit standards body, funded by its own membership. DITA is controlled (or perhaps a better word is "guided") by the OASIS DITA *Technical Committee (TC)*. The DITA intellectual property is commonly-owned, with the trustee being OASIS.

The OASIS *DITA Adoption TC* was formed in 2008 with the aim of "educating the global marketplace on the value of DITA for document creation and management" (OASIS DITA Adoption TC Web site, n.d.).

I joined OASIS in 2006, and in 2007, became the inaugural chairperson of the *DITA Help Subcommittee* of the DITA TC. In 2010, I became the inaugural chairperson of the *DITA Adoption Help Subcommittee* of the DITA Adoption TC.

I contributed to the DITA Help Technologies Guide, produced by the DITA Help Subcommittee, and also created whitepapers and transformation customizations for the DITA Adoption TC.

I also presented at industry conferences and published papers in industry publications either individually or in my role as an OASIS member. (Refer to *Industry Conferences and Publications* (see page 248).) In particular, participation in industry conferences provided me with the opportunity to discuss concerns and ideas with other DITA authors.

My work with the OASIS TCs and subcommittees has complemented the development of the Artefact and ancillary projects (such as the WinANT Echidna software), and the deliberations of and discussions within the TCs and subcommittees has informed the best practice recommendations embodied in the Artefact.

History of the Artefact and candidature

> *The Artefact started in 2006 as a collection of topics created for a university post-graduate DITA subject. The project was formalised into a PhD research proposal, and the main product of the research, The DITA Style Guide, was published in March 2011.*

This is a first person account of the history of the development of the Artefact.

The plan to create the Artefact began forming around April 2006, when approval was granted for the development of the Structured Authoring (HATC424) subject within the Swinburne University Technical Communication programme.

As the Structured Authoring subject was the first university-level offering focussing on DITA, and as DITA was a new field, there were no immediately suitable text books around which learning materials could be built. In consultation with the Technical Communication course convenor, Dr Kathy Betts, I started to pursue the idea of a PhD by publication, or a PhD by artefact and exegesis.

I started looking for suitable supervisors at Swinburne in late 2006. I tried Prof Doug Grant, Dr Judy McKay, Prof Jun Han, Dr Peter Ling and Dr Belinda Barnett, all of whom either felt their field of expertise was incompatible or were unable to take on the primary supervision role. The problem of finding a supervisor with expertise in the field of DITA, or more broadly structured authoring, or even more broadly technical communication, proved to be a major challenge, as there are very few academics working in those fields. Attempts to find supervisors outside Swinburne also failed, with Julie Fisher at Monash University being one of those approached.

Eventually, I found that Dr Andrew Stapleton from Swinburne was able to act as my principal supervisor, as his experience with games design and science communication had sufficient relevance to structured authoring and DITA. However, before the candidature documents were prepared, Dr Stapleton left Swinburne.

Throughout the period from mid-2006 to mid-2007, I had been preparing the materials that would eventually form part of the Artefact. These materials were prepared following a modular approach, so that materials could be re-used in different publications. In August 2007, the materials were used for the 12 part learning resources for the first intake of the HATC424 Structured Authoring subject.

The same repository of content was drawn upon to create the *Structured Writing in XML* workbook for TACTICS Consulting, published in October 2007. I continued adding to the repository of topics, from which materials were produced to create workshop materials for my DITA classes and seminars in Helsinki (October 2006), Los Angeles (March 2007), Melbourne (May 2007), Vilnius (September 2007), Portland (March 2008), Gold Coast (May 2008), Edinburgh (September 2009), Auckland (October 2008), Wiesbaden (November 2008), London (November 2008), Sydney (November 2008), London (March 2009), Seattle (March 2009) and Melbourne (May 2009). There was therefore a continuum of development of the Artefact from July 2006 to December 2009.

In January 2008 I started discussions with Dr Suku Sinnappan, who had experience in new fields of technology, having been heavily involved in research into Second Life. Dr Sinnappan agreed to be my primary supervisor, with Dr Peter Ling as second supervisor, in February 2008, and the research proposal was prepared in March 2008. My application for candidature was accepted by the Research Higher Degrees Committee in August 2008, after which I began working full-time on the PhD project.

The form of the Artefact matured into *The DITA Style Guide*, which was eventually published by Scriptorium Press in March 2011.

In addition to drawing on my experience teaching and implementing DITA in the workplace, the Action Research approach to development of the Artefact and Exegesis leveraged my continuous involvement in OASIS DITA Technical Committees since June 2006.

Chapter 4

Project Detailed Analysis and Reflection

Topics:

The major analysis tasks and points for reflection during the research project were:

- Identify the scope of the Artefact, and develop a structure for the content.

  This task involved analysing the components of existing style manuals, comparing the scope and structure of other publications and learning materials, and developing a definitive structure for the finished Artefact.
- Reflect on the modular, structured, semantic writing approach to be taken to create the Artefact, and compare that approach with conventional linear, style-based approaches to writing.

  This reflection was developed into a writing strategy that was used during the project.
- Determine what software tools to use for the writing task.

  The original selection of the authoring and publishing tools turned out to be a poor choice, and tools were consequently changed during the life of the project. In hindsight, this provided a new opportunities to understand issues of migrating content from other forms to DITA.
- Take stock at different milestones throughout the project to reflect on whether the project goals were being met, and to use the experience gained in writing Artefact topics to better understand the needs of the intended audience for the Artefact.
- Document solutions to specific problems that were encountered during the development of both the Artefact and the Exegesis.

Scope and structure

Different methods of structuring information explaining DITA and related structured authoring techniques were analysed, as part of the process of developing the information architecture for the Artefact.

The structure and scope of the Artefact, and the structure of topics within, was developed after analysing the structures of four different related types of information sets:

- other style manuals
- a structured authoring training workbook
- a DITA and structured authoring tertiary education curriculum
- the official OASIS DITA specification

This analysis informed the eventual chapter structure of the Artefact, as well as the lower level topic structure and breakdown of expository information within topics. Decisions regarding the scope of the Artefact (what to include and what not to include) was also informed by the comparative analysis.

The resultant high level information architecture for the Artefact was:

- Information Types and Topics
- DITA Map Files
- Syntax and Mark-up
- Language and Punctuation
- Graphics and Figures
- Cross-referencing
- Content Re-use
- Metadata and Conditional Processing
- The DITA Documentation Process
- DITA Authoring Concepts

Component analysis of style manuals

The primary topic areas of existing style manuals were analysed to suggest a structure for The DITA Style Guide.

The major components of the style manuals analysed within the *Literature Review* (see page 20) can be accumulated and rationalised into a master list of components.

- (Article) titles, headings and sections
- Capital letters
- Terminology
- Typographical (italics, non-breaking spaces, mathematical symbols)
- Quotations
- Punctuation
- Chronological items
- Numbers
- Geographical items
- Units of measurement and currencies
- Tabulation
- Captions
- (Bulleted and Numbered) Lists
- Linking
- Illustrations, images, pictures, graphics, and multimedia
- Usability and accessibility
- Legal requirements
- Branding and visual identity, including page design and layout
- Information Architecture (document structure)
- File and folder naming and structures
- Metadata, indexing and search
- Glossaries
- Technical standards (HTML, CSS, XML, etc)
- Process
- Editorial style
- Forms, applications and interactivity
- Writing style
- Writing tasks, procedures and steps
- Writing for an international audience
- Editing marks

The content in style manuals can be broadly grouped into:

- Writing (editorial) style
- Presentational style
- Process

The major components of existing style manuals could be categorised into those three groups, and then identified as relevant to a generic DITA, in-house DITA style manual, or in-house process guide, as follows.

Table 2: Relevancy of style guide components

Component	**Category**	**Relevant to Generic DITA Style Manual**	**Relevant to In-House DITA Style Manual**	**Relevant to In-House Process Guide**
(Article) titles, headings and sections	Writing style, presentational style	Yes	No	Yes
Capital letters	Writing style, presentational style	Yes	Yes	Yes
Terminology	Writing style, presentational style	No	Yes	Yes
Typographical (italics, non-breaking spaces, mathematical symbols)	Presentational style	No	No	Yes

Component	Category	Relevant to Generic DITA Style Manual	Relevant to In-House DITA Style Manual	Relevant to In-House Process Guide
Quotations	Writing style, presentational style	Yes	No	Yes
Punctuation	Writing style	Yes	No	No
Chronological items	Writing style, presentational style	Yes	No	No
Numbers	Writing style	Yes	No	No
Geographical items	Writing style	No	Yes	No
Units of measurement and currencies	Writing style, presentational style	No	Yes	No
Tabulation	Presentational style	No	No	Yes
Captions	Writing style, presentational style	Yes	No	Yes
(Bulleted and Numbered) Lists	Writing style, presentational style	Yes	No	Yes
Linking	Writing style, process	Yes	No	Yes

Component	**Category**	**Relevant to Generic DITA Style Manual**	**Relevant to In-House DITA Style Manual**	**Relevant to In-House Process Guide**
Illustrations, images, pictures, graphics, and multimedia	Presentational style, process	Yes	No	Yes
Usability and accessibility	Writing style, presentational style, process	Yes	No	Yes
Legal requirements	Process	No	Yes	Yes
Branding and visual identity, including page design and layout	Presentational style	No	No	Yes
Information Architecture (document structure)	Writing style, presentational style	Yes	Yes	No
File and folder naming and structures	Process	Yes	Yes	Yes

Component	**Category**	**Relevant to Generic DITA Style Manual**	**Relevant to In-House DITA Style Manual**	**Relevant to In-House Process Guide**
Metadata, indexing and search	Writing style	Yes	Yes	No
Glossaries	Writing style, presentational style	Yes	Yes	Yes
Technical standards (HTML, CSS, XML, etc)	Process	No	No	Yes
Process	Process	No	No	Yes
Editorial style	Writing style	No	Yes	No
Forms, applications and interactivity	Presentational style, process	No	No	Yes
Writing style	Writing style	No	Yes	No
Writing tasks, procedures and steps	Writing style	Yes	No	Yes
Writing for an	Writing style, process	Yes	No	Yes

Component	Category	Relevant to Generic DITA Style Manual	Relevant to In-House DITA Style Manual	Relevant to In-House Process Guide
international audience				
Editing marks	Process	Yes	No	Yes

This categorisation produces the following component list for a generic DITA style manual:

- (Article) titles, headings and sections
- Capital letters
- Quotations
- Punctuation
- Chronological items
- Numbers
- Captions
- (Bulleted and Numbered) Lists
- Linking
- Illustrations, images, pictures, graphics, and multimedia
- Usability and accessibility
- Information Architecture (document structure)
- File and folder naming and structures
- Metadata, indexing and search
- Glossaries
- Writing tasks, procedures and steps
- Writing for an international audience
- Editing marks

Of these components, “file and folder naming” and “editing marks” are possibly out of place, because it deals exclusively with process, which is not typically the domain of a generic style manual.

Note: The final structure of the Artefact saw these difficult subject areas collected into a small section named “The DITA Documentation Process”.

Comparative structure of Structured Authoring Workshop Guide

> *Some of the source topics of The DITA Style Guide were used in a Structured Authoring Workshop participant guide. The structure of this participant guide makes a useful comparison.*

The TACTICS Consulting *Structured Authoring in XML Participant Guide (Self, 2008)* was developed from the same source topic repository used by *The DITA Style Guide*. The structure of that Guide is:

- XML
    - Introducing XML
    - XML Technologies and Concepts
- Structured Authoring
    - Introducing Structured Authoring
        - Structured vs Style-based Authoring
        - A Topic-based Approach
        - Single Sourcing and Content Re-use
        - Benefits of Structured Authoring
    - Getting Started with Structured Authoring
        - Separation of Content and Form
        - Presentation and Delivery Formats
        - Semantics
        - Metadata
        - Style Guides
        - Case Study
        - Job Roles and Project Strategy
        - A Standard Structured Authoring Process
        - Techniques to Learn
- DITA
    - Introducing DITA
        - About DITA
        - OASIS and the DITA Technical Committee
        - Content Models and Information Types
        - DITA Open Toolkit
        - DITA and Information Mapping

    - Identifying Information Types
- Ditamaps
    - About ditamaps
    - Working with ditamap Files
    - Generated Relationship Links
    - Collection Types
    - Relationship Tables
    - Tools
    - Case Study

The only topical area that is appropriate to *The DITA Style Guide* but not identifiable in the proposed structure was *Content Models*. This would logically be categorised with *Information Types*.

References

- Self, T. (2008). Structured Writing in XML Participant Guide (1st ed.). Sydney: TACTICS Consulting. (Commercial training workbook)
- *TACTICS Consulting learning materials* (see page 255)

Comparative structure of HATC424 syllabus

> *Some of the source topics of The DITA Style Guide were used for the HATC424 Structured Authoring syllabus. The structure of the curriculum (and the associated student notes) makes a useful comparison.*

The Swinburne University of Technology HATC424 Structured Authoring subject is designed around the same topic repository used by *The DITA Style Guide*. The subject comprises 12 modules, and the notes for each of the modules can be compiled into a comprehensive set of student notes, which faithfully follow the curriculum. An abbreviated structure of those Notes is:

- Elementary XML
- XML and DITA Concepts and Terminology
- Semantic Mark-up
    - Semantics Expressed in Elements and Attributes
    - Metadata
    - DITA and DocBook

- Information Types
- Structured Authoring
- Collections and the Open Toolkit
- Creating and Editing Content in DITA
- Content Re-use
    - Many ditamaps from same Content Repository
    - Embedded Topics and ditamaps
    - Content Reference (conref)
    - Transclusion
    - Benefits of Content Re-use
    - Management of conref Blobs
    - Writing for Re-use
- Publishing
    - Reading Formats
    - Publishing
    - Conditional Publishing
- Teams
    - Project Strategy
    - Roles and Responsibilities
    - Skill Sets
- Schemas and Specialisation
    - Planning a Content Model
    - Inheritance
    - Specialisation
    - Why Specialise?
    - Namespaces
    - “Official” Specialisations
    - Constraints on Specialisation

- Domain Specialisation
- Generalisation

The only topical area that is appropriate to *The DITA Style Guide* but not identifiable in the proposed structure was *Collections*. This would logically be categorised with *DITA Map Files*. *Publishing* and *Schemas and Specialisation* are not appropriate to *The DITA Style Guide* because specialisation is about customised (not generic) information types, and because *The DITA Style Guide* is a guide for authors, not publishers.

I developed the HATC424 Structured Authoring subject curriculum.

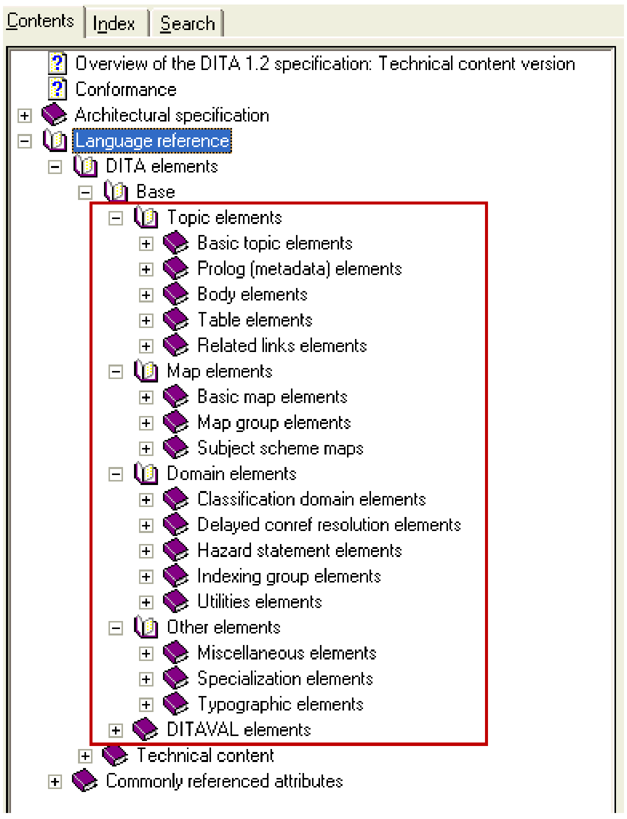
References

- Structured Authoring at Swinburne University of Technology. (n.d.). Swinburne University of Technology, Melbourne, Australia. Education, . Retrieved February 22, 2011, from http://courses.swinburne.edu.au/subjects/Structured-Authoring-HATC424/local.
- *Structured Authoring subject at Swinburne University of Technology* (see page 255)

Comparative structure of DITA Standards

> *A challenge similar to that of structuring a DITA style manual has already been tackled by the DITA Technical Committee in devising a structure for the language reference for the DITA standard.*

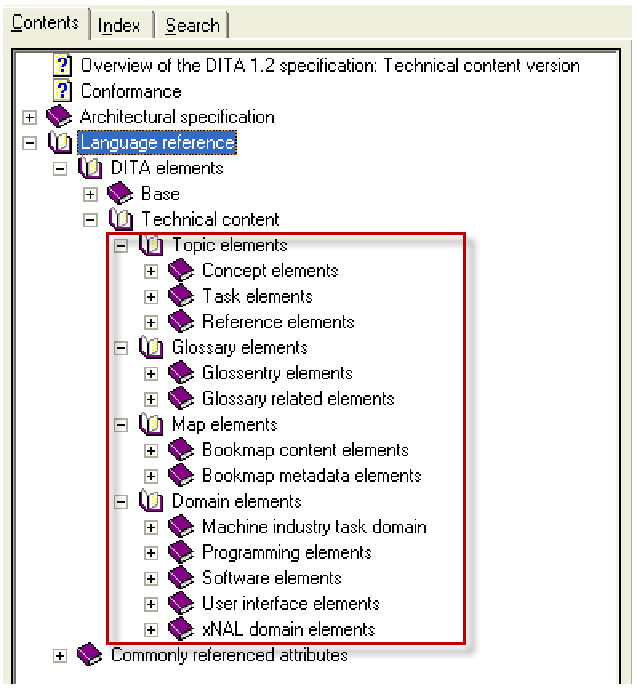
In devising a structure for the *DITA 1.2 specification* standards documentation, *Eberlein (2009)* proposed two options.



Contents | Index | Search

- Overview of the DITA 1.2 specification: Technical content version
- Conformance
- Architectural specification
- Language reference
  - DITA elements
    - Base
      - Topic elements
        - Basic topic elements
        - Prolog (metadata) elements
        - Body elements
        - Table elements
        - Related links elements
      - Map elements
        - Basic map elements
        - Map group elements
        - Subject scheme maps
      - Domain elements
        - Classification domain elements
        - Delayed conref resolution elements
        - Hazard statement elements
        - Indexing group elements
        - Utilities elements
      - Other elements
        - Miscellaneous elements
        - Specialization elements
        - Typographic elements
      - DITAVAL elements
    - Technical content
  - Commonly referenced attributes

Figure 11: Structure based on generic topic elements



Contents | Index | Search

- Overview of the DITA 1.2 specification: Technical content version
- Conformance
- Architectural specification
- Language reference
  - DITA elements
    - Base
    - Technical content
      - Topic elements
        - Concept elements
        - Task elements
        - Reference elements
      - Glossary elements
        - Glossentry elements
        - Glossary related elements
      - Map elements
        - Bookmap content elements
        - Bookmap metadata elements
      - Domain elements
        - Machine industry task domain
        - Programming elements
        - Software elements
        - User interface elements
        - xNAL domain elements
  - Commonly referenced attributes

Figure 12: Structure based on concept, task and reference topic elements

Another solution to the same structuring challenge was proposed by *Yeo (2009)*, who suggested an architecture based more on application of the standard than the schema of the standard.

- Topic elements
    - 2.0 Topic elements (rename this, e.g. to “topic structural elements”)
    - 7.0 Body elements
        - 11.03 fn
        - 11.07 tm
    - 8.0 Table elements
    - 10.0 Related links elements
    - 13.0 Typographic domain elements
    - 17.0 Utilities domain elements
    - 11.01 dita
- Elements for specific topic types
    - 3.0 Concept elements
    - 4.0 Reference elements
    - 5.0 Task elements
    - Glossary related elements
- Map and bookmap elements
    - 19.0 Map elements
    - 20.0 Map group elements
    - 21.0 Bookmap content elements
    - 22.0 Bookmap metadata elements
- Metadata elements
    - 9.0 Prolog elements
    - 18.0 Indexing group elements
    - Classification domain elements
    - 23.0 xNAL domain elements

    - Conref delayed resolution elements
    - 11.02 draft-comment
    - 11.04 indexterm
    - 11.05 indextermref
    - 11.06 index-base
- Industry-specific elements
    - 14.0 Programming elements
    - 15.0 Software elements
    - 16.0 User interface elements
    - Hazard statement elements
    - Machine industry task elements
- 12.0 Specialization elements
    - 11.08 data-about
    - 11.09 data
    - 11.10 foreign
    - 11.11 unknown
- Elements external to content
    - Subject scheme map elements
    - 24.0 DITAVAL elements

This list was further refined by Yeo *(2009)* and Eberlein *(2009)* to the final outline.

- Topic elements
    - Basic topic elements
    - Body elements
    - Table elements
    - Related links elements
- Map elements

    - Basic map elements
    - Map group elements

- Metadata elements
    - Prolog (metadata) elements
    - Indexing group elements
    - Delayed conref resolution elements

- Domain elements
    - Hazard-statement domain elements
    - Typographic domain elements
    - Utilities domain elements

- Classification elements
    - Subject scheme maps
    - Classification domain elements

- Specialization elements
- DITAVAL elements
- Legacy conversion elements

Yeo (2009) and *Eberlein (2009)* found it advantageous to avoid the use of a "Miscellaneous" section, and found that DITA elements without a clear place in the structure should be classified as follows:

Table 3: Revised Yeo and Eberlein structure

Element	Classification
dita	Basic topic elements
draft-comment	Body elements
fn	Body elements
indexterm	Indexing group elements

Element	Classification
indextermref	Indexing group elements
tm	Body elements
data	Specialization elements
data-about	Specialization elements
foreign	Specialization elements
unknown	Specialization elements

I considered these structuring options, but found while suitable for a language reference document, this structure would need to be adjusted for use for a style guide.

References

- Eberlein, K. J. (2009, August 21). Order for lang ref files (OASIS DITA TC correspondence). Retrieved from http://markmail.org/message/r67bf2jystkp34ir.
- Priestley, M., Anderson, R. E., & Hackos, J. (Eds.). (2007). DITA Language Specification v1.1. OASIS.
- Yeo, S. (2009, September 2). Order for lang ref files (OASIS DITA TC correspondence). Retrieved from http://lists.oasis-open.org/archives/dita/200909/msg00041.html.

Final structure of the Artefact

After analysing different high level structuring approaches, the final structure for the Artefact of ten major groupings was derived.

Modelling the structure of *The DITA Style Guide* on the proposed structure for the DITA 1.2 specification (see *Comparative structure of DITA Standards* (see page 77)) would result in six major groupings, as shown in *Figure 13: Major groupings following DITA Specification approach* (see page 82)

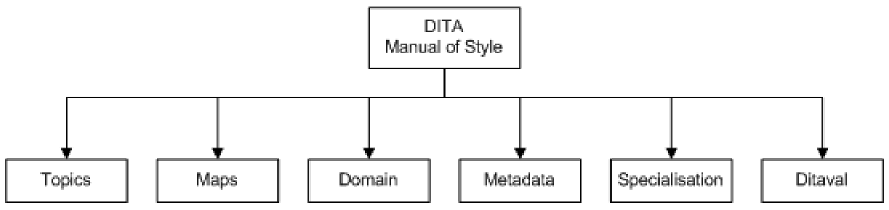


Figure 13: Major groupings following DITA Specification approach

This structure compares to that developed by associating topics during the life of the Artefact writing process, shown in *Figure 14: Major groupings resulting from association approach* (see page 83).

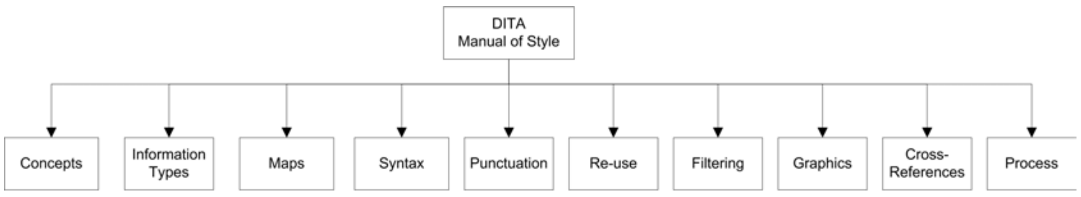


Figure 14: Major groupings resulting from association approach

When the two methods are compared (*Figure 15: Correlation between two approaches* (see page 83)), four groupings are found to be in common. The groupings that don't occur in both designs are in some cases due to the different scopes of the two documents, but in other cases highlight a flaw in the *association groupings*.

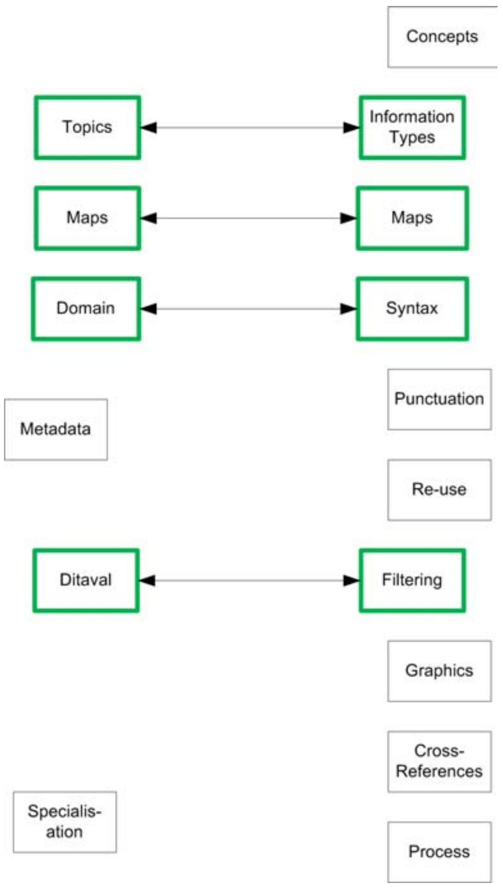


Figure 15: Correlation between two approaches

Specialisation was not to be covered in *The DITA Style Guide*, other than to provide specialisation naming conventions, because specialisation produces customised information structures that can't be documented in a generic style guide.

An analysis of the volume of content in each of the *association groupings* (shown in *Table 4: Content Volume in Association Groupings* (see page 84)) showed a slight imbalance of topics in the *Syntax* grouping. Moving the indexing topics from Syntax would start to redress the imbalance. Broadening *Filtering* to encompass metadata would resolve the issue of there being no grouping for that important area in the *association groupings*, and it would also enable some other topics to be moved out of *Syntax* (to *Metadata and Filtering*). Some other topics in Syntax may be more logically categorised into the *Punctuation* grouping if broadened to become *Language and Punctuation*.

Table 4: Content Volume in Association Groupings

Grouping	**Topic Count**
Concepts	12
Information Types	18
Syntax	39
Punctuation	11
Re-use	12
Filtering	11
Maps	12
Graphics	15
Cross-Referencing	12
Process	17

The changes would leave a revised structure of:

- DITA Authoring Concepts
- Topics and Information Types
- Working with Maps
- Syntax and Mark-up

- Language and Punctuation
- Working with Graphics and Figures
- Cross-referencing
- Content Re-use
- Metadata, Filtering, Flagging and Conditional Processing
- The DITA Documentation Process

Writing approach

Creating a document using DITA requires a different writing approach to that previously used for linear, style-based documents.

As part of the action research approach, for philosophical reasons (a book on DITA surely has to be written in DITA?), and for reasons of efficiency (DITA represents a more efficient way to plan and create documents), *The DITA Style Guide* was written in DITA. The act of writing helped identify some of the questions DITA adopters needs answered during a typical documentation project, and the observations made and lessons learned are documented in this Exegesis which accompanies the Artefact.

The primary difference in writing approach between a traditional, style-based document (authored using a tool such as Microsoft® Word) and a document authored in DITA is that the style-based authoring approach is linear. The document is made up of one long, multi-page file (or a small number of long files), written in a linear progression from first page to last page. The author visualises the document as a codex book, and the publication is usually produced as a codex book. DITA employs a topic-based architecture, which means that small, independent pieces of information are written individually, in any sequence. At the end of the process, the topics are collected into a sequence for publication in a variety of reading formats, including CD-ROM, book, e-book, and Web.

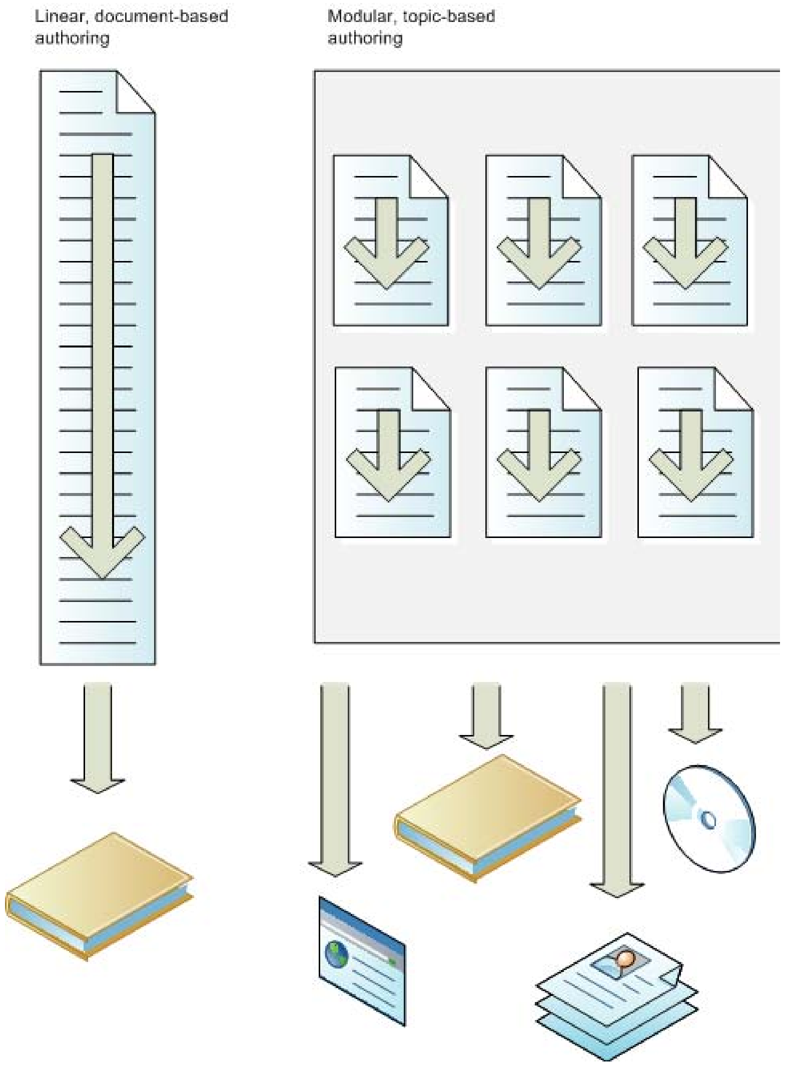

Linear, document-based authoring
Modular, topic-based authoring

Figure 16: Comparison of Linear and Topic-based Authoring

The topic-based approach can mean that the document is not started as a design skeleton, where parts, sections and chapters are planned in advance of the writing. An alternative approach is to write topics as the need for those topics arises, and then group those topics into the equivalent of parts, sections and chapters after the writing is complete. Topic-based authoring allows flexibility so that components of the writing task can be performed non-sequentially.

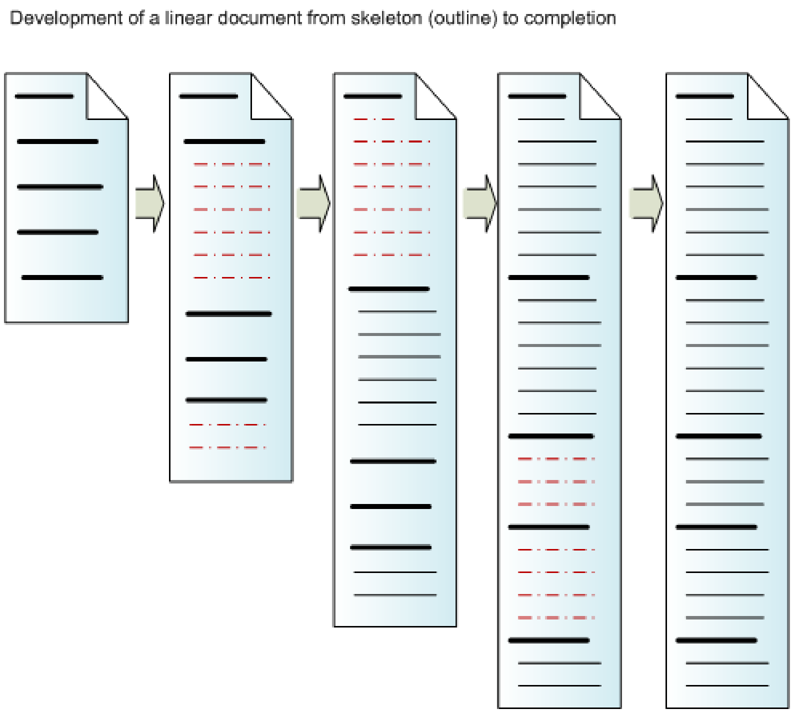

Development of a linear document from skeleton (outline) to completion

Figure 17: Development of Linear Document

Although topic-based authoring permits different document development workflows, *The DITA Style Guide* is being developed as individual topics before the document skeleton is created. A structure for the content will be designed by grouping topics together logically closer to the end of the project. This workflow suits the approach where the need for information to be included in the Style Manual is generated by analysis of DITA adopter mailing lists.

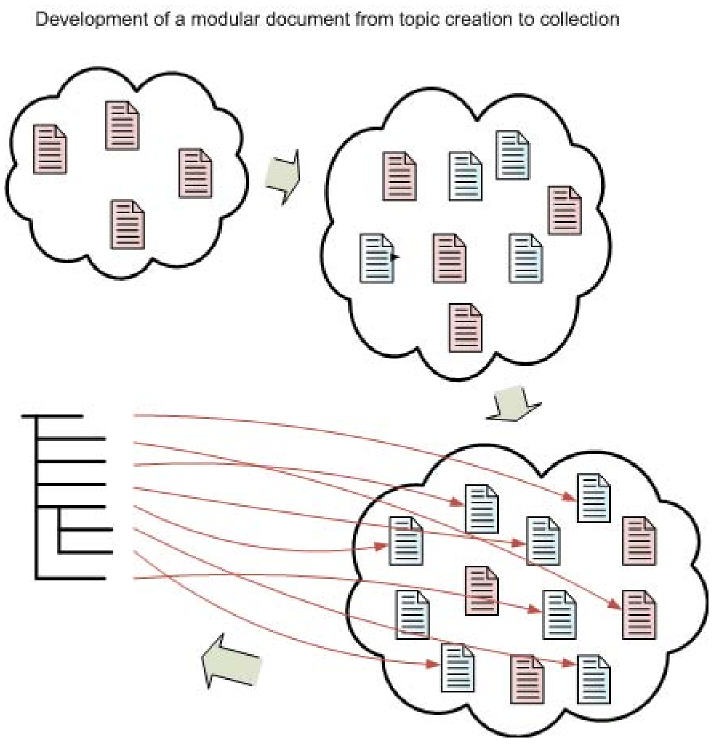

Development of a modular document from topic creation to collection

Figure 18: Development of Modular Document

In other words, topic-based authoring doesn't start with a definition of the deliverables. It starts with the requirement of a reader for information.

Putting the writing approach into practice

> *The theoretical topic-based writing approach was used in practice for the development of the Artefact. No major problems with this approach were encountered.*

The development of the Artefact progressed using a topic-based approach. Topics were written as independent information units, and indeed used in different publications.

Publications in which the topics were used included:

- Structured Authoring course materials (for the HATC424 subject at Swinburne University of Technology)
- *TACTICS Structured Workshop Participant Guide*
- academic papers
- the Artefact

Most topics were written to address or answer questions found during analysis of the Yahoo! DITA Users Group (see *Analysis of Yahoo! DITA Users Group* (see page 58)). While this approach was most suited to the Artefact being used as a reference (*look-up*) document, it may not be suitable for other information forms which have a more linear, narrative structure, such as a novel.

The upper level information architecture for the Artefact was developed well into the topic authoring process, and as a process mostly separate from the topic authoring process.

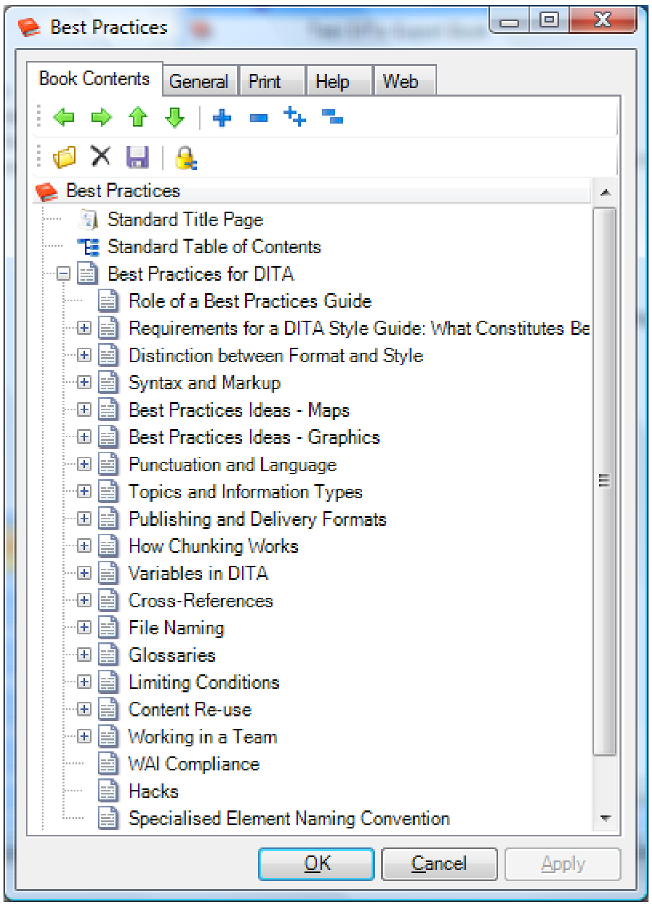
Progression of Artefact TOC

> *The bottom-up writing approach (writing standalone topics first and assembling them into a document structure later) is different to the linear skeleton approach (devising a topic structure, and later writing the topic content). The approach taken in The DITA Style Guide saw a working table of contents (TOC) evolve as more topics were added to the topic reservoir, before a final structure was developed as a separate endeavour.*

The high level structure for the Artefact was decided quite late into the topic authoring process. In fact, the major topic groupings were decided more than one year after the first topics were written.

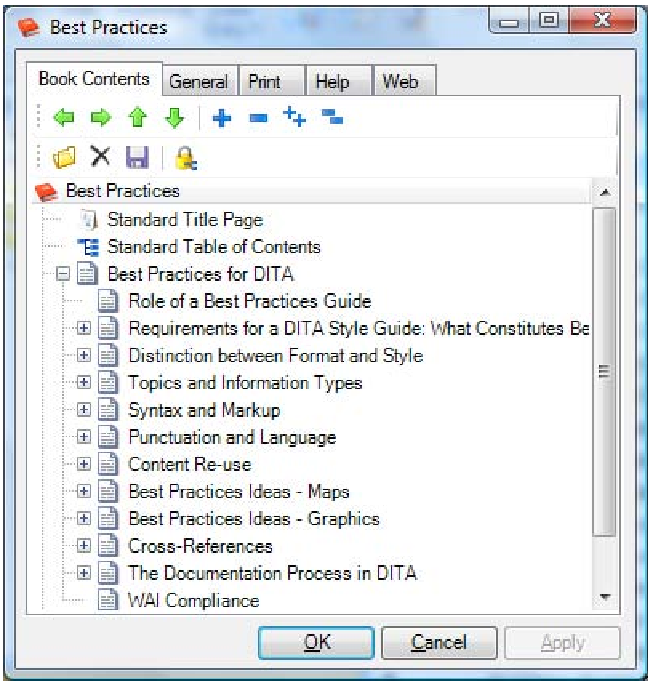
However, some sort of grouping was important during the authoring process to help manage the topics being developed, and so that draft publications could be produced for review along the way. The authoring tool used in the early stages, Author-it, provided a *book object* which is used to create a *table of contents (TOC)* for a publication. The book object was used to create a working TOC as the authoring of Artefact topics proceeded.

In September 2009, a major grouping of what had previously been a flat structure was undertaken, and the information architecture ideas being explored in the analysis of comparable information structures could be better visualised.



Best Practices
- Standard Title Page
- Standard Table of Contents
- Best Practices for DITA
  - Role of a Best Practices Guide
  - Requirements for a DITA Style Guide: What Constitutes Be
  - Distinction between Format and Style
  - Syntax and Markup
  - Best Practices Ideas - Maps
  - Best Practices Ideas - Graphics
  - Punctuation and Language
  - Topics and Information Types
  - Publishing and Delivery Formats
  - How Chunking Works
  - Variables in DITA
  - Cross-References
  - File Naming
  - Glossaries
  - Limiting Conditions
  - Content Re-use
  - Working in a Team
  - WAI Compliance
  - Hacks
  - Specialised Element Naming Convention

Figure 19: The DITA Style Guide TOC as at 11 August 2009



Best Practices
- Standard Title Page
- Standard Table of Contents
- Best Practices for DITA
  - Role of a Best Practices Guide
  - Requirements for a DITA Style Guide: What Constitutes Be
  - Distinction between Format and Style
  - Topics and Information Types
  - Syntax and Markup
  - Punctuation and Language
  - Content Re-use
  - Best Practices Ideas - Maps
  - Best Practices Ideas - Graphics
  - Cross-References
  - The Documentation Process in DITA
  - WAI Compliance

Figure 20: The DITA Style Guide TOC as at 1 September 2009

How it was done: Using citations

There are various standards for citing references in academic papers and journals. The standard used in the Exegesis is the Harvard style.

This procedure requires Mozilla Firefox®, and the *Zotero* extension to Firefox.

Use this procedure to create a *Harvard style* citation, and to mark up the DITA content to reference that citation.

1. If the reference does not already exist in Zotero, add a reference to Zotero normally.
2. In Zotero, right-click on the reference, and choose **Create Bibliography from Selected Item...**.

   The **Create Bibliography** dialog box will display.
3. In the **Citation Style** field, select **Harvard Reference format 1 (Author-Date)**.
4. In the **Output Format** field, select **Copy to Clipboard**.

   The citation will be copied to the Windows clipboard in Harvard format.
5. In XMetaL®, open the References topic (containing the bibliography), and paste the reference in as standard text.

6. In XMetaL®, open the topic where the citation is to be added, and type the citation in Harvard style (eg, *Jones, 2005*).
7. Mark-up the citation with the `cite` element, and then add a keyref attribute based on the citation itself (eg, `jones_2005`).

   The `keyref` attribute is not used by many (if any) DITA processors yet, but applying the keyref attribute will make it easier to take advantage of `keyref` functionality when it becomes available.

Example of Citation Code in DITA Content

```
<p>Butter was invented in 1765 (<cite>Jones 2005, p
34</cite>).</p>.
```

Authoring and publishing tools

The development of the Artefact was commenced with a non-DITA authoring and publishing tool, and changed to a DITA-specific toolset mid-way through the project.

The choice of authoring, publishing and content management tools plays a big part in the efficacy of the production process. However, when working to a standard, the authoring tool selection is not hard-to-reverse decision. A DITA topic, regardless of which tool it was created with, can be edited by any DITA authoring tool.

At the start of the Artefact authoring process, a non-DITA topic-based, modular authoring tool, Author-it, was used to write topics. Author-it is a very efficient authoring tool that offers a productive authoring environment. It also promised the option of exporting to DITA format. My initial intention was to use Author-it in the early stages of the project, and migrate to a rich DITA authoring tool, XMetaL®, after version 5 of that program became available. (Version 5.0 introduced support for DITA 1.1, which I had selected as the standard topics would be written to.)

When XMetaL® 5.0 became available, I created new topics in that software, and continued to revise existing topics in the Author-it software.

I completed the migration from Author-it to XMetaL® in November 2009.

Initial tools approach

Writing content topics requires an authoring tool. Author-it, which uses a proprietary topic format, and XMetaL®, which uses DITA, have both been used for this research project.

By April 2009, the research project was well-progressed, and hundreds of content topics had been written. The topics that would be drawn upon to form *The DITA Style Guide* had being written in two different technologies. Early topics were created in Author-it, an object-oriented content management system. Author-it had been selected because it offered a form of topic-based structured authoring before DITA tools had matured to the point of being effective and practical. Later topics were written in DITA, using a number of different authoring tools but primarily JustSystems XMetaL®.

The planned approach was that Author-it topics would be converted to DITA towards the end of the drafting stage of the project, by which time Author-it would be due to support a DITA-like structure overlay. This structure feature would make the conversion easier.

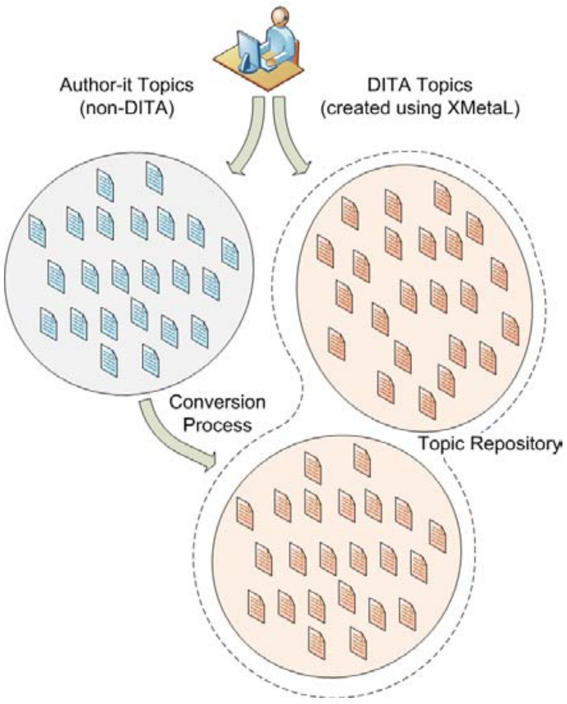

Author-it Topics (non-DITA)
DITA Topics (created using XMetaL)
Conversion Process
Topic Repository

Figure 21: Schematic Diagram of DITA Authoring Process

The DITA topics were written to the DITA 1.1 standard. It was originally planned that the page layout output of the Artefact would be generated using Elkera XML Print, although it only supported DITA 1.0 at the commencement of the project. It was anticipated that DITA 1.1 support in Elkera XML Print would be available by the time of completion of the Artefact.

Revised tools approach

The authoring and publishing software tools that were planned to be used for the research project were changed as the project developed, due to the tools having not evolved as expected.

DITA is an open source technology, so many software tools can be used in a DITA workflow.

In general, DITA tools fit into three categories:

- authoring tools
- publishing tools
- management tools

Although it is not vital to use management tools, authoring tools and publishing tools must be used in order to create a deliverable document suitable for reading.

The majority of the topics written in the early life of the research project were created in Author-it, a non-DITA proprietary authoring, publishing and management tool. From the start, the aim was to eventually extract the Author-it topics as DITA topics, using an *export* feature promised by the Author-it developers. Although this feature was still unavailable the end of 2009, an alternative method had been discovered to migrate content from Author-it to DITA. The majority of topics were thus moved to DITA over a three week period in November 2009.

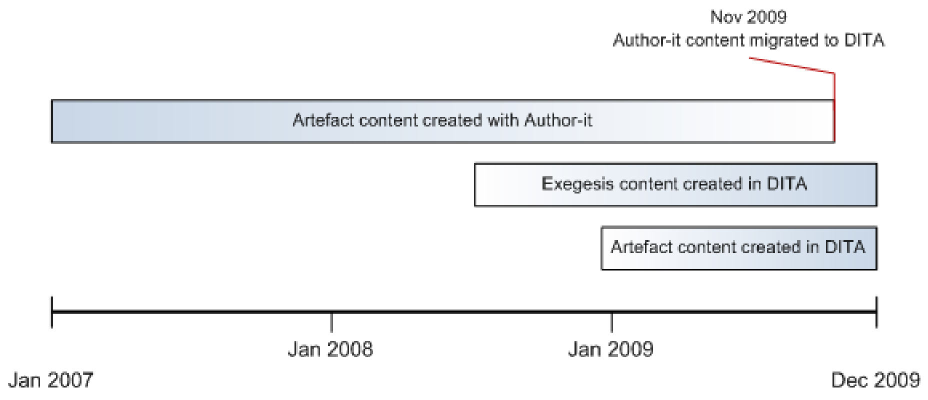


Figure 22: Timeline showing change from Author-it to DITA

From the publishing perspective, it was anticipated that the Elkera XML Print tool would be able to be used for publishing the project content into a Microsoft® Word or PDF document format. HTML-based content would be produced using the DITA Open Toolkit, and my own WinANT Echidna publishing utility. However, the promised support for DITA 1.1 was not available by November 2009. However, the DITA Open Toolkit *Idiom FO (PDF2)* plug-in had developed sufficiently to make it a viable alternative to XML Print. I developed templates, or *layouts*, to ensure the output was in an appropriate page design.

By the end of November 2009, the plan for publishing had changed so that PDF2 would be used for page layout output (via WinANT Echidna), and that the Open Toolkit would be used for HTML-based output (via WinANT Echidna). If PDF2 proved to be inadequate, the newer open source *ditac* tool would be investigated as an alternative.

WinANT Echidna therefore took on a more important role as the project progressed.

My development of WinANT Echidna was heavily influenced by the needs of the research project. A *layouts* feature was added so that a look-and-feel for a PDF publication could be pre-defined and then selected, during publishing, from a menu. Reports were introduced so that I could manage review and quality assurance aspects of the Artefact.

The final toolset used for authoring and publishing of the Artefact and Exegesis was:

- JustSystems XMetaL®
- WinANT Echidna
- DITA Open Toolkit

Reflection

> *At different stages during the project, I reflected on the challenges of authoring in a DITA environment, and recorded observations made. The most important checkpoint was in November 2009, when the bulk of the topics had been drafted, and the shape of the Artefact was becoming clear.*

What a DITA author needs

> *When writing within a DITA framework, an author needs to access style information as a guide to the use of semantic elements within DITA. The information needed will answer an author's specific question.*

The art of technical communication can be summarised as providing the right information in an easily accessible manner. The technical communicator needs to learn the user's goals in order to “design effective information to meet them” *(Hackos & Stevens, 1997, p. 20)*.

The goals of a technical communicator using *The DITA Style Guide* include choosing the best semantic elements for particular mark-up scenarios. The information in *The DITA Style Guide* therefore needs to be arranged in such a way as to help DITA authors meet their goals if it is going to be a useful resource.

Authors may discover semantic mark-up questions during the writing, file creation, indexing, structuring, metadata application, and specialisation stages of a documentation project.

Questions that I encountered when authoring the Artefact are likely to be typical of questions that any author working with DITA may encounter.

Writing

A typical question that a DITA author might ask during the writing process might be:

> *If I want to refer to a common name of an object, I might write, "often referred to as the common name". What semantic mark-up should I use for the common name?*

A style guide should answer these sort of questions. (The answer to this particular question is the `q` element.)

Structuring and File Naming

- Should I store glossary definitions in individual files, or collect glossary entries into a single, multiple-entry file?
- What's the difference between a topichead and a topicgroup?
- Where should conref files be stored?
- Should I name conref files so they appear at the top of a file list?

Indexing

- What's the difference between keywords, keyword and indexterm?
- Why doesn't a keyword appear in the index?

Mark-up Structure

- Should a table reside inside a paragraph?
- What element do I use to refer to the attribute name of an XML element? Or the attribute value, or a name-value pair?
- Should a codeblock (or similar) reside inside a paragraph?

Metadata

Should I conref author information in the prolog?

Specialisation

Should I use existing information types, or specialise?

Note: For *The DITA Style Guide*, I considered the question of whether base information types (principally concept and reference) should be used for the content, or whether a specialised `style-manual` information type should be used. The decision was to use the base information types, so that *The DITA Style Guide* be used as an exemplar of DITA mark-up, using base information types.

Content model

> *The content of a style manual should follow a consistent design pattern.*

The DITA Style Guide will contain some general conceptual topics, but the majority of the informational content will be best practice guidelines or rules, structured in a consistent pattern. That consistent pattern can be expressed in the form of the following content model.

```
Rule Title
Rule
Example
Related elements or attributes
Rationale
```

This model would best be formalised as a specialised form of the concept information type; however, *The DITA Style Guide* is also designed to serve as an exemplar of the use of the DITA base content model. As a consequence, an alternative method of using the audience attribute to define the rationale (this is semantically correct, because the rationale is designed for a different audience and publication than the rest of the content), and a restriction `note` element to define the related elements or attributes, has been adopted.

The DITA structure in pseudo-code for a rule topic is therefore as follows:

```
<concept>
 <title>...</title>
 <shortdesc>...</shortdesc>
 <prolog>...</prolog>
 <conbody>
  <p>...</p>
  <p>...</p>
  <example>
   <p>...</p>
   <codeblock>...</codeblock>
  </example>
  <section audience="academic">
   <p>...</p>
   <p>...</p>
  </section>
 </conbody>
</concept>
```

If a topic (or part of a topic) concerns features introduced in DITA 1.2, the revision (`rev`) attribute will be used to record the value `1.2`. If the entire topic concerns a 1.2 feature, the `rev` attribute will be set on the top level `topic` element, and the `topicref` element.

Testing ideas in Europe, November 2009

> *By November 2009, the bulk of the Artefact topics had been drafted or outlined, and the shape of the Artefact was becoming clear. I used two technical communication conferences*

> *in Europe to seek out opinions and views on the approach he had taken, and to otherwise test some of the ideas being proposed for The DITA Style Guide.*

I spent most of November 2009 in Germany; I attended to technical communication conferences, and spent the period between conferences *in retreat* in Nuremberg, working exclusively on the research project. As the data collection stage of the project was essentially complete at this point, the European visit gave me the opportunity to test some ideas developed in the project with OASIS colleagues from around the world, to review my progress to date, and to plan for the remaining stage of the project.

The two conferences I participated in were the tekom tcworld conference in Wiesbaden, Germany, and the DITA Europe Conference in Munich, Germany. The tcworld conference is a large conference organised by the Germany technical communication society (tekom), while the DITA Europe Conference is organised by the American ComTech company. Approximately 2100 people attended tcworld, and about 80 people attended DITA Europe.

During my visit to Germany, I had in-depth conversations regarding DITA and technical communication best practice with the following people:

- Scott Prentice (USA, DITA TC Member, and developer of the DITA-FMx software tool)
- Alan Houser (USA, DITA TC Member, and DITA consultant)
- Su-Laine Yeo (Canada, DITA TC Member, and JustSystems DITA evangelist)
- Dr Dimitra Anastasiou (Ireland, University of Limerick Localisation Research Centre)
- Kurt Ament (Germany, author of *Single Sourcing: Building Modular Documentation* reference)
- Jang Graat (Netherlands, DITA consultant)
- Bernard Aschwanden (Canada, author of XMetaL® training manual)
- Bryan Schnabel (USA, DITA TC Member and XLIFF TC co-chair)
- Dr JoAnn Hackos (USA, DITA Adoption TC chair, author and CIDM director)
- Kristen James Eberlein (USA, DITA TC Member)
- Robin Sloan (USA, DITA TC Member, and PTC DITA evangelist)
- Gunthilde Sohn (Germany, instinctools founder)
- Prof Sissi Closs (Germany, DITA TC Member, Comet Communications founder, Karlsruhe University of Applied Sciences)
- Nolwenn Kerzreho (France, Université Rennes 2 Haute Bretagne)
- Rodolfo Raya (Uruguay, DITA TC Member, XLIFF TC Member)

- Marie-Louise Flacke (France, Université Rennes 2 Haute Bretagne)

Summary of Discussions

Individual conversations clarified my thoughts on many of the rules, and provided suggestions for areas of contention that also needed coverage. There was a universal enthusiasm for the product of the research project, namely *The DITA Style Guide*. There was also a consensus that the ultimate aim should be to move *The DITA Style Guide* to a Wiki, so that its ongoing development can be an open source, collaborative effort. Participation bonds the writers to the style guide *(Hyttinen, 2009, 59)*.

DITA Europe Conference

I made the following observations about the information presented at the DITA Europe conference.

- Many case studies identified a need for best practices information.
- Many case studies showed the importance of a project style guide.
- Some case studies owned up to poor semantic mark-up for reasons of:
    - the need to maintain legacy practices (eg, including stem sentence in last paragraph of context - XMetaL templates session)
    - obviating a requirement to change the publishing process (eg, not including notes within paragraphs because the presentation of such structures was inconsistent)
- Some delegates revealed poor titling practices, such as starting all concept topics with “About the...” or task topics with “How to...”.
- Few people (perhaps 5%) were using specialisations. (My approach of not including specialisations appeared to be the correct decision, as covering specialisations would complicate matters for the remaining 95%.)
- Controlled language may play a role in best practices.
- Many talks promoted the need to simplify DITA for the novice author.
- Even the DITA Technical Committee members at the conference were unsure of the meaning of some semantic elements.

Julia Malkin from Endeca, in a presentation entitled *Ensuring Consistency with Map and Topic Templates*, explained the use of templates in her organisation to provide “self-documenting usage guides”. For a map architecture, Endeca use *unit maps* to allow re-use of “chapter” content chunks. A *unit map* always starts with a concept topic at the root. The *unit maps* are grouped into *collection maps*, made up only of *maprefs* to the unit maps (ie, with no *topicrefs*). A document is defined as a bookmap (for page layout) or an online document ditamap, containing a title, document metadata, and *maprefs* to the constituent *collection maps*.

Other speakers referred to the need for writing guidelines to circumvent the approach of “DITA allows me to, so why can't I?”; an approach that often results in technically correct but semantically incorrect topics. For example, a warning that needs to appear above a step can be placed at the end of the preceding step. The result satisfies the XML rules for DITA file structure, but is semantically incorrect, as the warning doesn't belong with the step in which it is nested. Another example is using a footnote (`fn`) element within a footnote element. While DITA permits this technically, in that it allows an `fn` element to be nested within an `fn` element, it is not at all good practice (and will result in unpredictable rendering in the output).

For those involved in localisation, there seemed to be an agreement that translation and localisation best practice for DITA environments is *XLIFF*.

Case studies commonly showed poor semantic mark-up used to overcome processing inadequacies. It was sometimes a mystery as to why the processing wasn't adjusted instead of providing ugly hacks to the DITA mark-up. For example, Sheila D'Annunzio, in a session called *Ugly DITA*, reported that her organisation was placing images side-by-side in tables in order to achieve a suitable PDF output presentation.

How the Artefact was created

> *When a significant problem or challenge was encountered, I documented the problem and solution in “How it was done” topics.*

Writing *The DITA Style Guide* was a DITA authoring project in itself, and could therefore be capitalised upon as a case study. The primary purpose of the Exegesis overall is to document and reflect upon the process of creating the Artefact, and a number of “How it was done” topics were accordingly created during the project.

How it was done: Migration from Author-it to DITA

> *Migrating The DITA Style Guide content from Author-it to DITA format was a carefully planned and executed task.*

The overview of the process was:

1. Work on a subset of about 10 topics at a time.
2. Change the template for the subset of topics to Swinburne DITA Template.
3. Modify the styling in each topic to make the structure valid.
4. Adding the topics to a specially-created DITA Export book.

5. Publish the DITA Export book as DITA.
6. Copy the resultant DITA topics and supporting graphics to the Artefact DITA folder structure.
7. Rename the topic files to more descriptive file names, following the naming convention.
8. Add the topics
9. Add metadata to each of the migrated DITA topics.
10. Add draft-comments to the migrated DITA topics to note any editing suggestions.
11. Add image metadata (width, height, etc).
12. Rename the image files to use more descriptive file names, following the naming convention. Alternatively, you can change the Web **Filename** in the **Topic Properties** in Author-it
13. In Author-it, change the status of the exported topics to *Migrated to DITA*, locking the topics to prevent further changes. The colour of the topic name will change to red.
14. In Author-it, remove the exported topics from the DITA Export book.

Note: Had to add q, synph, term, and other character styles to improve quality of resultant DITA concept topics. Also had to modify the XSL-T D2H transformer to permit lq and cite elements to be correctly handled.

How it was done: Links to Language Reference

> *Topics in The DITA Style Guide are linked to relevant topics in the DITA Language Reference.*

Topics in *The DITA Style Guide* typically relate to one or more DITA elements or attributes. These topics are linked, in the HTML-based version of *The DITA Style Guide*, to the associated topics extracted from the *DITA Language Reference*, published by OASIS.

The linking of external topics is made possible by one of the benefits of a standards-based structured authoring environment: *interchange*. DITA topics created by different authors can easily be integrated into the same deliverable document, as the mark-up, based on semantics, not presentation, is essentially identical.

DITA is also built on modular documentation principles, and it is generally more manageable for large projects to be broken down into smaller modular components at both the map and the topic levels. The Artefact topics are collected into a top-level *bookmap* primarily made up of *maprefs* to subordinate *ditamaps* at section (chapter or part) level. Maps contain not only the hierarchical sequence of topics in the collection, but also relationships between topics stored in a relationship table. The relationship table information is manifested as cross-reference hyperlinks in the output. The relationships between Artefact topics and DITA Language Reference topics are stored in a relationship table in a dedicated ditamap (named `artefact_crossreferencing.ditamap`), stored alongside the other Artefact section ditamaps.

The DITA Language Reference is very complex, with conref elements and embedded cross-references in most topics. Including the cross-references in the Artefact to the Language Reference topics adds significantly (approximately 20 minutes) to the build time. While the document is at a draft stage, there is no need to include the cross-references, except perhaps for milestone builds. Likewise, there is no need to include the cross-references in the page layout outputs (typically PDF).

Conditional publishing techniques can be used to exclude the cross-references from draft and print output. The method adopted was to apply a `product` attribute of `academic_only` to the topicref to the ditamap containing the relationship table.

The topicref code in the top level map is:

```
<topicref format="ditamap"
  href="Artefact/artefact_langref_reltable.ditamap"
  scope="local" navtitle="DITA Authoring Process"
  print="no" product="academic_only">
</topicref>
```

This approach meant that a publishing condition to exclude the *academic_only* would temporarily remove the cross-referencing from the build. For final (and milestone) builds, the condition could be easily removed.

The `print` attribute was also set to `no`, which, although apparently not functional in the DITA Open Toolkit (the build time is not significantly reduced), is semantically correct, as the relationship tables are not intended to be applied to any print output.

The links to the Language Reference were not activated in the published *The DITA Style Guide*, by using conditional processing to exclude the relationship table containing the links from the output. The links may be later activated in either electronic versions of the publication or when the source is donated to the DITA community as an open source Wiki.

How it was done: Links to Microsoft Manual of Style

> *Topics in The DITA Style Guide are linked to relevant topics in the HTML version of the Microsoft Manual of Style for Technical Publications.*

Many topics in *The DITA Style Guide* have a common purpose with topics from the *Microsoft Manual of Style for Technical Publications*, published by Microsoft Press. In some circumstances, it is useful for readers of the Artefact to compare usage recommendations with that of the *Microsoft Manual of Style*. This cross-referencing capability was highly desirable in the authoring stage, and may therefore be equally useful when the Artefact is maintained in future.

As an HTML version of the *Microsoft Manual of Style* was available via a download from the Microsoft site, relationship tables were used to establish a relationship between an Artefact topic and any relevant *Microsoft Manual of Style* topics. The approach was similar to that taken for the links to the DITA Language Reference, although for the Language Reference the topics were DITA, while the *Microsoft Manual of Style* were HTML.

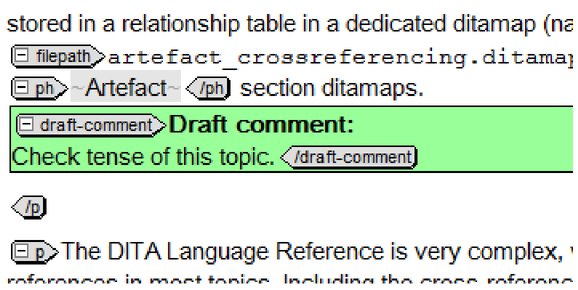
The links to the *Microsoft Manual of Style* were not activated in the published *The DITA Style Guide*, by using conditional processing to exclude the relationship table containing the links from the output. The links may be later activated in either electronic versions of the publication or when the source is donated to the DITA community as an open source Wiki.

How is was done: Private author comments in topics

> *The draft-comment element was used to store author comments during the document development process.*

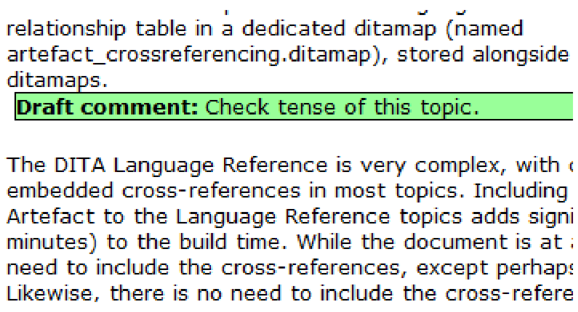
As for most writing projects, there were often times during the development of *The DITA Style Guide* when a topic could not be completed, or facts needed to be checked. The `draft-comment` element provided a useful means of leaving private author comments within the flow of the topic.

One of the benefits of using `draft-comment`, as opposed to standard XML comments, is that the `draft-comment` content can easily be excluded **or** included in the output through a DITA Open Toolkit build file parameter. This allows the comments to be visible to writers and editors during the drafting stage, but otherwise excluded.



stored in a relationship table in a dedicated ditamap (na
filepath artefact_crossreferencing.ditama
ph Artefact /ph section ditamaps.
draft-comment **Draft comment:**
Check tense of this topic. /draft-comment
/p
p The DITA Language Reference is very complex,
references in most topics. Including the cross-referenc

Figure 23: Example of draft-comment Mark-up in DITA Source


relationship table in a dedicated ditamap (named
artefact_crossreferencing.ditamap), stored alongside
ditamaps.
Draft comment: Check tense of this topic.

The DITA Language Reference is very complex, with
embedded cross-references in most topics. Including
Artefact to the Language Reference topics adds signi
minutes) to the build time. While the document is at
need to include the cross-references, except perhaps
Likewise, there is no need to include the cross-refere

Figure 24: Example of draft-comment Displayed in Output (HTML)

How it was done: Audit trail

> *The Artefact source includes a record of the rationale behind a DITA usage recommendation; that record or audit trail is not output in The DITA Style Guide.*

Many of the usage recommendations in *The DITA Style Guide* were devised in response to questions raised on the Yahoo! DITA Users Group or other DITA adopter forums. To the average DITA adopter, the audit trail and rationale of a recommendation is not important; for these readers, the rationale should not be provided. However, for other users of the Artefact, such as people extending the recommendations, or future contributors of a Wiki version of *The DITA Style Guide*, the rationale may be important.

To provide for this dual purpose, the *audit trail* has been included after the usage recommendation at the bottom of the respective topic. The audit trail is contained within a `section` element, which is marked with an audience attribute of `contributor`.

In the primary DITA authoring tool used for the project (XMetaL®), applying an attribute changes the background colour in the *WYSIOO* editor. This made it easy to distinguish between *The DITA Style Guide* content and the *audit trail* rationale.

Where a topic cited references, and where the references are not appropriate for *The DITA Style Guide* but necessary or useful for other forms of the Artefact, a similar approach was adopted. The section at the bottom of the topic is titled “References”, and marked with an audience attribute of `contributor`. The inline citation is also marked with the `contributor` audience attribute.

How it was done: Moving conref Source without a CMS

> *During the evolution of the Artefact and Exegesis, the original location of the conref source file was found to be inferior to an alternative, and the file had to be moved.*

When the Artefact and Exegesis were being written, a common file naming and structuring approach was adopted. During the process, a more effective structure was discovered.

The original *conref source file* location was in the root folder of the topic repository, with the conref source file named with a `conref_source` prefix. This prefix distinguished the conref source file from topic files, which were prefixed according to information type (such as `t_`, `c_` or `r_`).

Originally, both Artefact and Exegesis topics were stored in the same folder. As the project developed, the topics were organised into separate /Exegesis and /Artefact folders. The overwhelming majority of topics belonged to the Exegesis, as the Artefact was initially written in the Author-it authoring tool. All the *conref target* topics (places into which the conref source content was transcluded) were in the Exegesis topics. To avoid broken references, the *conref source files* were moved with the Exegesis topics into the `/Exegesis` folder.

Over time, the information to be conreffed was organised into five files:

- `conref_source.dita`
- `conref_source_email.dita`
- `conref_source_footnotes.dita`
- `conref_source_metadata.dita`
- `conref_source_references.dita`

As the number of topics in the Exegesis grew, it became more difficult to locate the *conref source* file in the folder; prefixing concept topics with a `c_` meant that alphabetical sorting of file names would leave conref topics scores of files down the list, after the concepts. This added to the time required to use a conref, and would therefore discourage an author's use of transclusion.

As the Exegesis was migrated from Author-it to DITA, the need to conref from Author-it topics to *conref source* files arose. Although the directory structure would allow this (the main technical requirement is that the ditamap is at a higher level than any topics referenced), logic dictated that the *conref source files* be outside the `/Exegesis` folder. The new location was to be in a dedicated `/conref` folder immediately under the root folder.

Moving the *conref source file* meant that than 140 references needed to be changed to reflect the new relative URL. As a Content Management System was not used in the authoring process, editing of the references had to be done manually.

Locating conref usage would have been difficult, even after adding a conref usage report to the WinANT tool. An alternative approach using a search and replace software tool called *Helpware FAR* was devised. FAR allows a folder of text-based files, including XML files, to be searched for a snippet of code, and found snippets to be replaced with another snippet. FAR was used to search for `"conref_source.dita` in the Exegesis topic files, and replace them with `"../Exegesis/conref_source.dita`.

This approach resulted in the entire conref relocation task taking less than 20 minutes. The new file location resulted in easier conref usage in the primary authoring tool, XMetaL®.

How it was done: Nomenclature

> *Structured authoring in DITA is sufficiently different from style-based narrative authoring that different terminology is used in describing the documentation process.*

One of the difficulties in working within a single-sourcing, topic-oriented, XML-based, structured, semantic mark-up, DITA methodology is that familiar technical communication and documentation terms are no longer adequate.

A simple term such as *document* could mean many things. It could be a DITA document, that is, a topic. It could be an output document, such as a PDF. It could be a Help document, which is a collection of topics perhaps compiled into one or more Help files.

Likewise, common terms such as publication, file, link, cross-reference, TOC and map, have different meanings. The core reason for the changed or confused meaning is that DITA separates content from form. Most of the terms we use have a single meaning in a world where content and form are intertwined. A Word document that an author creates is the same document that the reader will read. By contrast, a reader will never read a DITA document; the DITA document has to be assembled and transformed into a reading document format before the reader sees it.

During the development of the Artefact, the use of terms had to be careful considered. A document (irony intended) called *A new vocabulary for semantic, structured authoring* was created to serve as a working glossary during development. This helped ensure that terminology was used consistently in the Artefact.

Most terms defined in the *Publication and collection defined**A publication (also known as the output or the deliverable) is the document, Web site, eBook, Help system or other product produced when a collection of DITA topics is processed through a publishing program.* topic are marked up with `term` elements when being described in the in the Artefact topics. When a DITA element is being used literally, it is marked up with the `synph` element. When a common DITA element is being used as a common term, such as “conref”, “conreffed”, “topicrefs”, “DTD”, “bookmaps”, or “ditamaps”, they are not marked up. However, avoid capitalisation of such terms (eg, “Bookmap”) when practical. The phrase “embedded ditamap” is preferred to “nested ditamap”. Don't use “reltable”; use “relationship tables” or “reltable sections” instead.

How it was done: Exemplars and sample topics

> *In many cases, an exemplar or sample topic is an effective way of explaining mark-up practices.*

There are two main reasons why exemplar or sample topics were created during the development of the Artefact:

- to illustrate best practice (to readers of *The DITA Style Guide*)
- to test different mark-up approaches for different output formats (for my own purposes)

Exemplar and sample topics are of little use to the reader when the content is in a reading format. (Without seeing the mark-up, there is little benefit.) However, if the reader has access to the DITA source for the Artefact, the exemplar and sample topics could be very useful.

For this reason, exemplar and sample topics needed to be hidden when the DITA source was output to a reading format, while remaining in the source file set, in the correct position in the ditamap hierarchy.

Linking of the exemplar and sample topics also required to be controlled, so that such topics were not viewed in isolation from the subject matter they were designed to complement.

To meet these requirements, the exemplar topic must be referenced in the ditamap as a child of topic that references it. The `topicref` attributes for the exemplar topic must be set as follows:

linking	target only
toc	no

If the nature of the example does not warrant a whole topic, such as if the example is a small section of code, the sample should be added as a `example` element, with the title set to `Sample of [title of sample]`. If the sample is used for testing purpose only, set the `audience` attribute of the `example` element to `contributor`. If the example illustrates something described in the body of the topic, the example should have no `audience` attribute (thus ensuring it is output for all publications).

To make it easier to find exemplar topics and code examples in HTML-based versions of the Artefact, an index entry of `exemplar` was used in the `prolog` of such topics. (In some cases, secondary keywords were created beneath `exemplar`.) The index entry was given an `audience` attribute of `academic`, so they keywords could be removed from the final, paper-based, published version of the Artefact.

How it was done: Rules and `importance` attributes

> *When a rule recommendation is made in the Artefact. the rule paragraph is marked with an `importance` attribute of high. This mark-up permits identification of rules in the publishing process.*

The role of *The DITA Style Guide* is to provide a set of guidelines and rules for DITA adopters. Specific and deliberate rules are scattered across the document. In the future, it might be required to extract the rules alone from the document to create a checklist or shortened form of *The DITA Style Guide*.

Although not designed for this reason, the `importance` attribute (normally used to identify the relative importance of individual steps in a procedure) has been used to identify rule recommendations in the Artefact. The rule paragraph (or other element) is marked with an `importance` attribute of *high*.

Rules should be written in the second person.

Although not currently implemented, some future publishing process may use the importance metadata during processing. A WinANT Echidna report was added to prove the principle, and to provide a quality assurance function. (Refer to *WinANT Echidna project* (see page 254).)

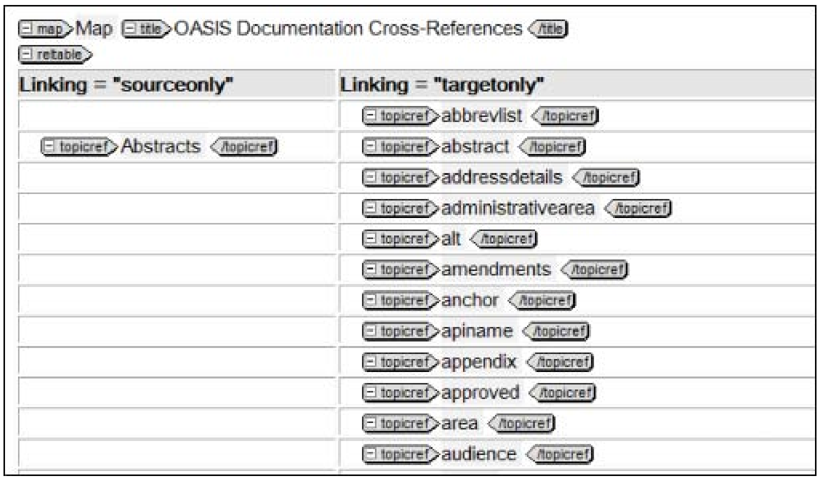
How it was done: Ensuring all DITA elements were covered

> *The relationship table used to generate links between topics in the Artefact and the corresponding topics in the DITA Language Reference was used to ensure that the Artefact covered all relevant DITA elements.*

DITA contains hundreds of elements and attributes, and for *The DITA Style Guide* to be effective, it must provide guidelines for the use of most of those components. To help manage this requirement in a topic-based authoring project (using a bottom-up rather than a top-down planned approach), the DITA Language Reference was used as a checklist.

Plans were previously made to link elements in the Artefact to corresponding topics from the OASIS DITA Language Reference through a relationship table. By adding all topic references for the Language Reference to the table in advance, and then populating the table with the Artefact topic references as the topics were completed, the `reltable` itself could be used for managing completeness.

A WinANT Echidna extension was written to generate a ditamap from a directory of topics, referencing every found `.dita` and `.xml` topic with a `topicref`. This generated file was then copied and pasted into the reltable in the ditamap file used to generate the links between Artefact and Language Reference. It was found that having *target* (Language Reference) topic references without source topic references (Artefact) did not affect the build process. (Refer to *WinANT Echidna project* (see page 254).)



map Map title OASIS Documentation Cross-References /title
reltable

Linking = "sourceonly"	Linking = "targetonly"
	topicref abbrevlist /topicref
topicref Abstracts /topicref	topicref abstract /topicref
	topicref addressdetails /topicref
	topicref administrativearea /topicref
	topicref alt /topicref
	topicref amendments /topicref
	topicref anchor /topicref
	topicref apiname /topicref
	topicref appendix /topicref
	topicref approved /topicref
	topicref area /topicref
	topicref audience /topicref

Figure 25: Relationship Table (in XMetaL editor) with WinANT-generated topicref

How it was done: American English

> *The Artefact was written in Australian English, but published in US English. The "translation" was managed with the `xml:lang` attribute.*

One of the challenges of large-scale single-sourcing is that of language. There are many forms of English, for example, and a topic written in one variation (such as Canadian English) will appear to have inconsistent spelling when included in a collection primarily made up of UK English topics.

This challenge was encountered with the Artefact. Some of the topics in the Artefact (in particular some in the "Authoring Concepts" appendix), were used in academic papers (in Australian English) and magazine articles (in Australian English). However, the publisher of the Artefact required the content to be supplied in US English.

Rather than replace the Australian spelling with US spelling (which would have introduced other single-sourcing problems), contentious spelling was marked up with the `xml:lang` attribute. This attribute is a generic XML attribute used to identify the human language of the content.

The `xml:lang` was used at phrase level for Australian spelling (eg, "capitalisation"); with the attribute value set to en-AU. The top level topic element of completed topics were given `xml:lang` attribute values of en.

To help manage the wording, I created new WinANT Echidna reports so that all instances of en-AU usage could be easily retrieved. the use of report on understand what Australian English word have been used. Exegesis in Australian English, Artefact in Australian English, *The DITA Style Guide* in US English. Conrefs used (on phrase elements) for common words such as “specialisation”.

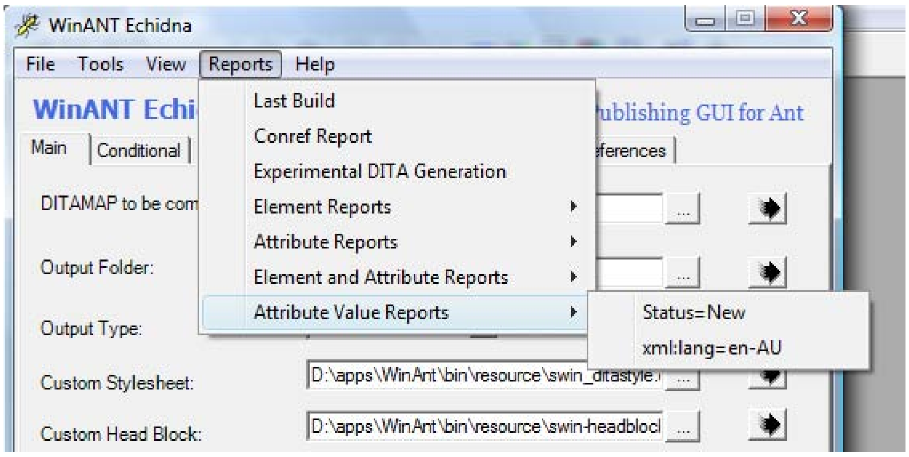


Figure 26: Reports menu options in WinANT Echidna

At the end of the project, Australian English phrases in the text flow were replaced with conreffed phrase elements (in the `conref_source.dita` file). A US English version of this conref source file was then created, and supplied to the publisher as an alternative to the Australian English version.

The WinANT Echidna reporting feature proved to be invaluable for this task.

How it was done: Using citations

> *There are various standards for citing references in academic papers and journals. The standard used in the Exegesis is the Harvard style.*

This procedure requires Mozilla Firefox®, and the *Zotero* extension to Firefox.

Use this procedure to create a *Harvard style* citation, and to mark up the DITA content to reference that citation.

1. If the reference does not already exist in Zotero, add a reference to Zotero normally.
2. In Zotero, right-click on the reference, and choose **Create Bibliography from Selected Item...**.

   The **Create Bibliography** dialog box will display.
3. In the **Citation Style** field, select **Harvard Reference format 1 (Author-Date)**.
4. In the **Output Format** field, select **Copy to Clipboard**.

   The citation will be copied to the Windows clipboard in Harvard format.

5. In XMetaL®, open the References topic (containing the bibliography), and paste the reference in as standard text.
6. In XMetaL®, open the topic where the citation is to be added, and type the citation in Harvard style (eg, *Jones, 2005*).
7. Mark-up the citation with the `cite` element, and then add a keyref attribute based on the citation itself (eg, `jones_2005`).

   The `keyref` attribute is not used by many (if any) DITA processors yet, but applying the keyref attribute will make it easier to take advantage of `keyref` functionality when it becomes available.

Example of Citation Code in DITA Content

```
<p>Butter was invented in 1765 (<cite>Jones 2005, p
34</cite>).</p>.
```

Chapter 5

Essays

Topics:

Essays in this Exegesis fulfil the following purposes:

- reflective documentation of observations made during the creation of the Artefact
- canvassing of topics for further research or later development into academic papers or journal articles
- documentation of the decisions made and approaches taken in the Artefact project

Each type of essay may have a different intended audience. In some cases, the essay has been written in academic style, while in others, a journalistic style has been adopted.

- *Writing to STOP*
- *STOP and DITA*
- *Writing paragraphs*
- *Using journalistic techniques in structured technical communication*
- *Changes in technical communication practice*
- *Levels of re-use*
- *DITA schema declarations*
- *Constraints in DITA 1.2*
- *Filtering and flagging content in DITA*
- *Variables: conrefs or attributes?*
- *DITA and semantics*
- *Escape from Author-it*
- *DITA and Content Management Systems*
- *The Artefact and Miller's Law*
- *Approaches to procedural topics*
- *Making decisions to remove context*
- *Improved glossary and terminology handling*
- *Examples of letter case in titles*

- *Separation of content and form in People and Place Style Sheet*
- *Semantic elements for user documentation*

Relationship between Essays and Artefact

Essays written during the research project helped crystallise my approach to different aspects of the Artefact.

Essay	**Chapter in Artefact**	**Comments**
The nub of the best practice problem: stem sentences (see page 117)	4 - Language and punctuation	Stem sentences, glue text, and transitional information.
The importance of DITA as a standard (see page 119)	Appendix A - DITA authoring concepts	Role of a style guide, and document interchange.
The content model for Artefact and Exegesis content (see page 121)	-	Considered in developing content model for topics.
Reflections on writing short descriptions (see page 123)	3 - Syntax and mark-up	Short descriptions.
Thesis statements in short descriptions (see page 125)	3 - Syntax and mark-up	Guidelines for crafting short descriptions.
Citations in base content model DITA (see page 126)	4 - Language and Punctuation	Quotation marks.
Benefits of content re-use (see page 130)	7 - Content re-use	
Using indexes with conditional publishing (see page 131)	8 - Metadata, conditional processing and indexing	Indexing, conditional processing concepts.
Writing to STOP (see page 132)	4 - Language and punctuation	
STOP and DITA (see page 135)	4 - Language and punctuation	

Essay	**Chapter in Artefact**	**Comments**
Writing paragraphs (see page 136)	4 - Language and punctuation	Crafting paragraphs.
Using journalistic techniques in structured technical communication (see page 137)	-	
Changes in technical communication practice (see page 139)	9 - The DITA documentation process	
Levels of re-use (see page 140)	7 - Content re-use	
DITA schema declarations (see page 142)	-	Considered during production of the Artefact.
Constraints in DITA 1.2 (see page 143)	9 - The DITA documentation process, and Appendix A - DITA authoring concepts	Restricting authors and limiting element choices, Constraints.
Filtering and flagging content in DITA (see page 144)	8 - Metadata, conditional processing, and indexing	Filtering and flagging.
Variables: conrefs or attributes? (see page 145)	7 - Content re-use	Variables.
DITA and semantics (see page 147)	-	Considered in overall context of writing the Artefact.
Escape from Author-it (see page 148)	-	Documented experience when producing the Artefact.
DITA and Content Management Systems (see page 152)	9 - The DITA documentation process	Content Management Systems.

Essay	Chapter in Artefact	Comments
The Artefact and Miller's Law (see page 153)	-	
Approaches to procedural topics (see page 153)	3 - Syntax and mark-up	Procedures and steps.
Making decisions to remove context (see page 154)	-	Documented experience when producing the Artefact.
Improved glossary and terminology handling (see page 204)	6 - Cross-referencing	Links to glossary terms.
Examples of letter case in titles (see page 162)	4 - Language and punctuation	Titles and headings.
Separation of content and form in People and Place Style Sheet (see page 163)	-	
Semantic elements for user documentation (see page 165)	3 - Syntax and mark-up	

The nub of the best practice problem: stem sentences

Technical communicators ingrained with style-based authoring techniques face a great challenge in migrating to a structured authoring environment.

A question that might be asked by such a technical communicator might be:

What element do I use for the stem sentence to introduce a procedure, such as the "To start the engine:" preceding a numbered list of steps. DITA doesn't give me a lot of choices.

Using DITA's base content model, there is no element suitable for a stem sentence. In other words, a stem sentence can't be used with the base DITA content model.

Leaving aside the option of *specialising* a new topic type (based on the base task topic type) to include a stem sentence element, the answer to this question should be “DITA does not permit a stem sentence, so don't use one”.

It is possible to argue that a stem sentence can't be semantically identified, and therefore is redundant in a task topic. (What is the purpose of stem sentence? Isn't it obvious to the reader? Doesn't the title of a task provide the same information as a stem sentence?) Stem sentences evolved within narrative writing, and are not required in a topic-based environment where a rule of one task per topic is generally applied.

In the following example of a style-based procedure, the stem sentence does not add to the meaning.

> ***Starting a Car*** *You must start the car before driving. To start a car: 1. Insert the key. 2. Turn the key. 3. When the engine starts, release the key.*

In DITA, the same procedure would be represented as:

```
<task>
      <title>Starting a Car</title>
      <taskbody>
      <shortdesc>You must start the car before driving.</shortdesc>

      <steps>
      <step><cmd>Insert the key</cmd>.</step>
      <step><cmd>Turn the key</cmd>.</step>
     <step><cmd>When the engine starts, release the key.</cmd></step>

      </steps>
      </taskbody>
      </task>
```

Many technical communicators may not be happy about having to drop the stem sentence, and will try to find a way to shoehorn it into the topic.

For example, someone could try to use the `prereq` element for the stem sentence, even though this would make the mark-up semantically incorrect. Likewise, the `context` element could be used, provided the `prereq` was not.

One company has instituted a practice whereby the last paragraph of the `context` element is used for the stem sentence. To make the stem sentence stand out in the print or online output, the company had their writers put the stem sentence within a `b` element (to make it “look noticeable”), breaking the *separation of content from form* principle. This approach, like others that break the semantic mark-up paradigm, will neuter interoperability. That company's use of DITA has become specific to the company; unless other people use the last paragraph of `context` in the same way, their documents cannot be interchanged. Standard tools can no longer be used when there is a company-specific special purpose of a particular paragraph of a particular element. And if you break the semantic rules that underpin the DITA methodology, there's really little point in using DITA.

The DITA Technical Committee did not forget to include a stem sentence element. It was not included deliberately, after careful consideration and consensus amongst the committee members. Procedural stem sentences are a form of *transitional information*. Transitional information is the *glue text* intended to inform readers of what has come before or what comes after a particular procedure, description, or explanation.

Dr JoAnn Hackos, in a posting to *DITA XML.org*, wrote:

In topic-oriented authoring, which forms the basis for the DITA Model, transitional text has become problematic. If the goal is to write standalone topics that may be used in more than one context, where does one put the transitional text? Does transitional text belong in a topic by itself? Should it be built into a task, concept, or reference topic? Does transitional text belong in the map so that it does not encumber the standalone topics?

...I contend that transitional text plays no role in most technical manuals, being an unneeded relic of a book-oriented design. People using manuals for perform work do not approach a text in the way we assume they should, as a book to be studied with concepts to be learned. They are not readers but are users of information, reading only enough to reach a particular goal.

Hackos, 2006

Technical authors moving to structured authoring may need to be prepared to abandon some beliefs erroneously held to be fundamental truths, and accept that DITA generally reflects the consensus on best practice, topic-based documentation.

The importance of DITA as a standard

Standards are used in many industries to promote quality and interoperability. DITA provides a standard that can be likewise used to promote quality and interoperability in the documentation industry.

DITA is an open source *standard*. The purpose of a standard is to allow people to work to an agreed set of rules, guidelines or designs. One of the benefits of DITA is *interchange*: the ability of content to be exchanged between authors, authoring tools, and publishing processes seamlessly.

If two authors are working to the same technical standard (in this case, DITA), then they can edit each other's work, collaboratively develop content, and share common text components, without any special requirements or expertise, such as knowledge or a corporate style.

If an author is using an industry document standard (in this case, DITA), he or she can work on the same document using one authoring tool at work, and an entirely different authoring tool at home, without any importing, exporting or translation.

If content written to a technical standard (in this case, DITA) is sent for publishing in different channels, then the publishing process can reliably produce the correct deliverable documents. For example, standard DITA content sent to a standard DITA PDF generation process requires no special or additional processing or handling. The same content sent to a standard DITA Web publishing process will also require no special handling.

Standardisation therefore introduces efficiency, predictability and reliability.

Contrast this with non-standard document formats. A Microsoft Word document sent to an author with Adobe FrameMaker or even Microsoft Publisher may not open without a special tool, and may lose formatting or metadata during the process. A RoboHelp project from work can't be edited in Author-it from home. A Web page authored in Adobe DreamWeaver to one design can't be dropped into a Web site using a different design.

Standards are one of the building blocks of civilisation. When standards work well, we don't notice them. Everyone in the world has standardised on a time system, made up days with 24 hours, and hours with 60 minutes. We don't think of time measurement as an international standard, though. Most people in the world have standard units of length and weight. These standards were introduced in the agrarian age. Within a country (and sometimes within groups of countries), there is a standard currency.

While we tend not to notice standards that work, we do notice competing standards that inhibit interoperability. When railways were first laid down in Australia, before Federation, each state chose its own rail gauge. Australians are still paying for that lack of standard a century later in increased interstate freight costs. The increase in international travel has exposed the weakness of the lack of a standard currency. While Europe has made steps to address this weakness, other nations are burdened by the extra effort involved in foreign currency exchange, arbitrage, and currency fluctuations. The competing standards of metric and Imperial measurements creates overheads for many organisations, be it a car manufacturer having to make two different types of speedometers, map makers having to print two sets of measurements, or software developers having to provide input for a health Web site in pounds and kilograms. We also end up with strange, sometimes politically-driven compromises, such as in aviation, where horizontal distance is measured in kilometres, but vertical distance is measured in feet!

In those fields where there are no standards at all, waste and confusion can be found everywhere. A personal video recorder saves TV programmes in a format that can't be read by an iPod. A car wheel with different stud spacings that can't be attached to a different car. A pool chlorinator that won't fit to the connections of another manufacturer's pump. A Web site with a Help icon that few people can recognise.

The worst problems arise when standards exist, but are ignored. Non-standard sized books won't fit on a bookshelf. A Web page that will open in Internet Explorer, but not in Firefox. A storage device that only fits one brand of camera. A door that needs a special size jamb. A light fitting that requires special bulbs. A computer application with a non-standard interface.

In the paper *Technical Communications Standards: New Directions in Innovation (Krechmer, 1999)*, different evolutionary strata of standards were identified. In fields of enterprise, *units and reference standards* are first devised (such as inches), then *similarity and methodology standards* follow (such as barrels must be 36 inches in height, and made of oak), before *compatibility standards* are agreed (barrels must have a one inch circumference hole to fit a tap).

For documentation content, the units and reference standards have long been established as words and pages. (A technical manual is measured by page count or word count.) Some *similarity and methodology standards* have arisen, such as readability requirements and style guide conformance. It is much harder to find established *compatibility standards*, however. This is where DITA has most to offer, although *etiquette standards*, the final stratum of standard *(Krechmer, 1999)*, is also served by DITA's XML foundations. Krechmer identified the view that public compatibility standards should be open, because they are “too important to allow any private organization overwhelming proprietary advantage”. *(Krechmer, 1999)*

DITA offers technical communication the opportunity to standardise on a document storage format. This opens up efficiencies by way of improved interchange, reduced learning curves, cheaper production processes, lower translation costs, shared content, and reduced need for format conversion. Coupled with standardised approaches to documentation design, such as *ISO 26514, Requirements for designers and developers of user documentation*, DITA clarifies and simplifies the planning, design, writing, editing and publishing phrases of technical document production.

The content model for Artefact and Exegesis content

> *There are many ways in which the content model for a document or suite of documents can be designed. This essay documents the approach taken for a PhD by artefact and exegesis.*

The authority of a public style manual (such as *The Chicago Manual of Style*) normally stems from the prestige or standing of the style manual's publisher. Over time, a style manual will create its own authority through its popularity and longevity. Style manuals generally don't provide a rationale for their rules and recommendations. The reader assumes that the authors of the style manual are authoritative.

For a new style manual, particularly one for a new writing domain such as DITA, there is little authority of the publisher's reputation, or of the longevity or popularity. It is important, therefore, for a new style manual to provide some rationale for rules and recommendations to provide a scaffold while the reputation of the manual is built over time.

In the case of *The DITA Style Guide* being created as the Artefact within a PhD by Artefact and Exegesis project, the Exegesis provides a mechanism for documenting the rationale for decisions made in the Style Manual, particularly those subjective decisions likely to be contentious.

The Style Manual may be delivered in a number of forms, including one where the rationale is included or cross-referenced, and another where the rationale is excluded. Conditional processing techniques in DITA can be used to generate these different publications from the same source content.

The two main content model approaches to support this conditional processing are as follows:

- Create a rule topic (Artefact) and a rationale topic (Exegesis) and link the two through a relationship table. Conditionally process using different ditamaps.
- Create a single topic containing rule and rationale, and use `product` metadata to identify each. Conditionally process using the `product` metadata in the ditaval.

Modular documentation architectures are more flexible when the content is more granular. *Ament (2003, p6)* suggests that the criteria for choosing what information goes into a topic is whether it can “stand-alone”. Using metadata to identify elements within a single topic imposes a maintenance workload that is higher than the alternative method of controlling publications through ditamaps and relationship tables. It is also likely that two separate version of *The DITA Style Guide* would need two separate ditamaps in any case (due to different colophon information, for example), so it would be simpler to follow the separate topics design.

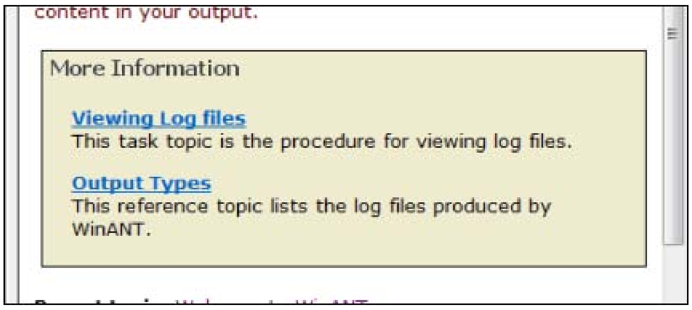
The separate topic approach relies on relationships being established the rule topic and its corresponding rationale topic. The DITA relationship table (*reltable*) feature accommodates this type of metadata. The reltable will define hundreds of relationships between rules and rationales, and its creation and maintenance is functionally and logically separate from the hierarchical topic structure. For this reason, the reltable should be maintained in its own ditamap in this scenario, and that ditamap should be nested inside the Artefact and Exegesis ditamaps.

Reflections on writing short descriptions

> *During the course of the Artefact project, I changed my approach to writing short descriptions from a simple, single sentence expansion of the title, to a thesis statement approach. This has resulted in more effective, expository short descriptions, even though the crafting task is more difficult.*

When I started writing DITA topics, my approach to short descriptions was to write a one sentence expansion of the topic heading, start with "This topic...". I believed at the time that this approach delivered a useful introductory paragraph for the topic.

When I started to view the short description as metadata, rather than content, my views changed. When the `shortdesc` is used as progressive disclosure devices, particularly in stub topics, it became obvious that the "this topic" form was clumsy, and did not result in expository preview text. Such "this topic" short descriptions often presented redundant information, repeating the information in the topic heading albeit with different wording. I needed to spend more time on the short description, in direct proportion to their importance. If readers are reading less and less information *(Self, 2008)*, short descriptions may be the only part of a topic they read!



content in your output.

More Information

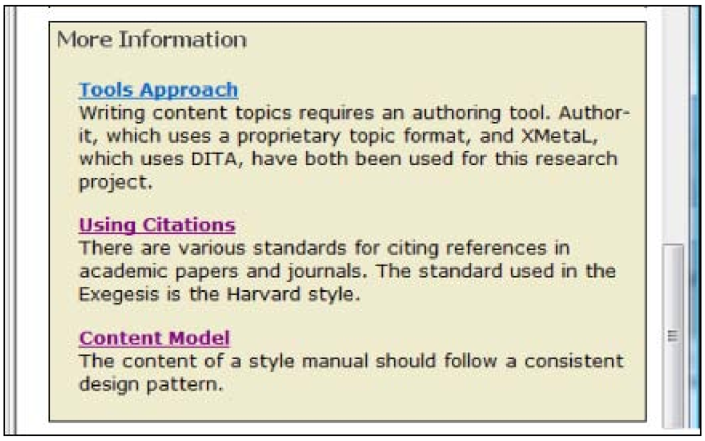
Viewing Log files
This task topic is the procedure for viewing log files.

Output Types
This reference topic lists the log files produced by WinANT.

Figure 27: Example of Short Descriptions drafted with the "This Topic "approach

After attending a presentation by Kris Eberlein on crafting short descriptions at the 2008 DITA Europe Conference *(Eberlein, 2007)*, reviewing the slides from the *Short Description Guidelines* presentation by Michelle Carey and Shannon Rouiller to the Silicon Valley DITA Interest Group *(Carey and Rouiller, 2008)*, and researching the STOP methodology *(Tracey et al., 1965)*, my view of short descriptions had changed. I now see short descriptions as the most important component of a DITA topic, and the most difficult component to craft.

I adopted a thesis statement approach to writing short descriptions, and this has resulted in better quality summaries that work effectively as overview paragraphs in a topic, and as progressive disclosure devices. The focus is now more on the information than on the topic unit.


More Information

Tools Approach
Writing content topics requires an authoring tool. Author-it, which uses a proprietary topic format, and XMetaL, which uses DITA, have both been used for this research project.

Using Citations
There are various standards for citing references in academic papers and journals. The standard used in the Exegesis is the Harvard style.

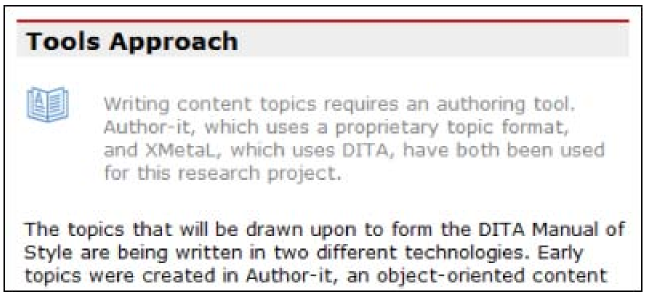
Content Model
The content of a style manual should follow a consistent design pattern.

Figure 28: Example of Short Descriptions drafted with the "thesis statement"approach

Thesis statements, applied to each topic in a modular, topic-based document, serve to:

- advise the reader of the significance and relevance of the topic
- help the reader decide whether to continue reading the topic
- help the *skimming reader* understand the central theme of the topic without reading the remainder of the content
- answers
- the reader's questions of "what" and "why".

I also modified the processing of my own topics so that the short descriptions in the topic were more easily recognisable. Rather than appearing in the same style as normal paragraphs in the output, I used a distinctive styling which distinguished the `shortdesc` metadata from the content.


Tools Approach

Writing content topics requires an authoring tool. Author-it, which uses a proprietary topic format, and XMetaL, which uses DITA, have both been used for this research project.

The topics that will be drawn upon to form the DITA Manual of Style are being written in two different technologies. Early topics were created in Author-it, an object-oriented content

Figure 29: Example of distinctive styling applied to short descriptions in topics

Thesis statements in short descriptions

Techniques used for writing thesis statements in different forms of writing can be applied to the challenge of writing effective short descriptions. Thesis statements reveal the precise purpose of the topic to help both the author and the reader.

Thesis statements can be used in different forms of writing to focus attention on the main point or theme of a text chunk. For *expository* texts, where a principle, issue or idea is being explained, the thesis statement should summarise the facts concerned. For *analytical* texts, where an issue or idea is broken down and analysed, the thesis statement should summarise the analysis and state the result. For *argumentative* or *persuasive* texts, the whole proposition and supporting evidence should be summarised into the thesis statement. Thesis statements are less helpful for narrative texts.

Thesis statements were a central pillar of the modular *STOP* writing methodology developed at Hughes-Fullerton in the 1960s *(Tracey et al., 1965)*. According to the STOP approach, “the precise object of the passage can always be kept clear for both the author and the reader through the device of the thesis sentence” *(Tracey et al., 1965)*.

Thesis statements, applied to each topic in a modular, topic-based document, serve to:

- advise the reader of the significance and relevance of the topic
- help the reader decide whether to continue reading the topic
- help the *skimming reader* understand the central theme of the topic without reading the remainder of the content
- answers
- the reader's questions of “what” and “why”.

In DITA, short descriptions should be written as thesis statements, not as opinion statements. The *Jackson School of International Studies Writing Center* at the University of Washington explains the key difference between an opinion statement and thesis statement as *(Tips on Writing Your Thesis Statement, n.d.)* “a thesis conveys to the reader that the claim being offered has been thoroughly explored and is defendable by evidence”.

The STOP approach *(Tracey et al., 1965)* suggested that a thesis statement should:

- state your proposition concisely
- repeat the key words of the topic body
- use adverbial conjunctions which show a train of reasoning (because, since, so, therefore, however, but moreover, etc.)
- use comparative adverbs and adjectives which show attitude and conclusions (more, least, highly, almost, too, very, good, better, only, etc.)

- must boil down the theme body to 25-30 words
- must show the whole proposition and proof (or substance otherwise) at a glance
- should be an argument, or an arguable hypothesis.

This is a good approach to take into writing DITA short descriptions.

Citations in base content model DITA

> *DITA authors can use referencing systems for citations and bibliographies without needing to use a specialised information type. The `cite` element in base DITA, and references imported from reference management systems into unordered lists, are the features used. References can be and re-used using the DITA conref mechanism. Future DITA developments, and interchange with open source XML-based referencing standards, will streamline the process further.*

Base content model DITA (before DITA 1.2) comprises the topic proto information type, and concept, task and reference information types. Many DITA adopters choose to use the concept, task and reference information types without specialising into more focussed information types.

There are a number of semantic elements in base DITA that can be used to meet the requirements of academic publishing. One of the principle concerns when preparing an academic paper is that of citations. Academic publication style guides typically dedicate a lot of space to detailing reference presentation; in fact, some referencing systems are known by the name of the publisher.

Some popular referencing systems include:

- American Psychological Association (APA)
- Harvard
- Chicago Manual of Style
- Nature Journal

References are made up of citations and bibliographies. A citation is an abbreviated reference key to a published or unpublished source. The full detail of the referenced work is contained in a bibliography, or in bibliographical footnotes. A bibliography is a compilation of referenced works, typically located at the end of a document. Bibliographies can stand in their own right, but citations must be used in conjunction with a bibliography, because the citation is only a reference key.

Note: The Chicago Manual of Style refers to a citation as an “in-text citation” or a “note” and a bibliographic reference as a “reference-list entry” or a “bibliographic entry” *(The Chicago Manual of Style Citation Quick Guide, n.d.)*.

In the APA style, a citation may take the form of `(Weber, 2004)`, where the full reference in the bibliography is `Weber, S. (2004). The Success of Open Source (p. 320). Harvard: Harvard University Press.`

Referencing systems allow interested readers to find a referenced work for reading in its own context, for delving deeper, or for fact checking.

The fact that citation systems are numerous is a hindrance to single-source academic publishing. If a portion of an academic paper is to be re-published in a second or subsequent publication, the citations and bibliographical references may need to be re-worked to suit the different publication requirements. Reference management systems such as EndNote® provide ways for an author to manage references and referencing systems, but they requires manual intervention at a number of stages.

Open source referencing standards, and open source reference managing software, may provide new opportunities for efficiency. In particular, XML-based standards and tools may allow greater interchange of citations and references. XML may permit a single, world-wide repository of standard published work references to be accessed and re-used by academic authors.

As an XML-based standard, DITA provides an insight into how referencing may work.

DITA has a semantic `cite` element designed for citations. According the DITA Language Reference, the `cite` element “is used when you need a bibliographic citation that refers to a book or article” *(Priestley et al., 2007)*. Although the definition goes on to indicate that the citation “specifically identifies the title of the resource” *(ibid.)*, it also reveals the future intention of “[allowing] the citation to be associated to other possible bibliographic processing” through the as yet unused `keyref` attribute.

Glossaries of terms have a similar information model to bibliographies, particularly for a bibliographic style which includes the citation abbreviation and the full reference. A publicly-available glossary specialisation, used in conjunction with a future implementation of `keyref`, will provide a better solution than a simple base DITA implementation, but it is still instructive to examine how a current base DITA implementation can work.

Bibliographic references are authored as list items within an unordered list in a special topic intended for re-use. The file name of the topic is `conref_source_references.dita`. Each reference item is given a human readable `id`, such as `cite_OASIS_DITAv1_1_LangRef`, so that it can be easily *transcluded* into a bibliography for a particular publication. All references for the entire repository of topics (which may be used in multiple documents) are stored and maintained in the one re-use topic, so that no reference ever gets re-typed.

If the bibliographical reference style needs to be changed, the re-use topic can be replaced with a similar document with an alternative reference style, or filtering attributes used in conjunction with conditional publishing.

For example, one reference could be recorded in the re-use topic as:

```
<li id="cite_Peters_CambridgeEnglishUsage">
<ph platform="APA">Peters, P. (2007). The Cambridge Guide to
Australian
English Usage (2nd ed., p. 924). Sydney: Cambridge University Press.
</ph>
<ph platform="Harvard">Peters, P., 2007. The Cambridge Guide to
Australian
English Usage 2nd ed., Sydney: Cambridge University Press.
</ph>
<ph platform="Nature">1. Peters, P. The Cambridge Guide to Australian
 English Usage.
924(Cambridge University Press: Sydney, 2007).
</li>
```

If the APA version was required, conditional publishing could exclude the versions with a `platform` attribute of Harvard and Nature.

The master references are stored in a Zotero database, through a Mozilla Firefox® plug-in. Zotero stores the reference information in an SQLite database. However, the data can be exported to a number of formats, including the open source MODS and BibTeX standards, and Zotero's own RDF XML format. The XML formats of MODS and Zotero RDF are the most suitable for interchange with DITA, as transformation from an XML with a rich semantic mark-up to one with the same or lower fidelity semantics will be technically trivial.

The MODS format stores a bibliographic reference as in the following example:

```
<mods>
 <titleInfo>
  <title>The Cambridge Guide to Australian English Usage</title>
 </titleInfo>
 <typeOfResource>text</typeOfResource>
 <genre authority="local">book</genre>
 <genre authority="marcgt">book</genre>
 <name type="personal">
  <namePart type="family">Peters</namePart>
  <namePart type="given">Pam</namePart>
  <role>
   <roleTerm type="code" authority="marcrelator">aut</roleTerm>
  </role>
 </name>
 <part>
  <extent unit="pages">
   <start>924</start>
   <end>924</end>
  </extent>
 </part>
 <originInfo>
```

```
  <edition>2</edition>
  <place>
   <placeTerm type="text">Sydney</placeTerm>
  </place>
  <publisher>Cambridge University Press</publisher>
  <copyrightDate>2007</copyrightDate>
 </originInfo>
 <identifier type="isbn">0521702429</identifier>
</mods>
```

In theory, references could be stored in MODS format within a DITA topic, in the same way that other *foreign* XML content, such as SVG and MathML, can be incorporated into DITA XML mark-up.

The scenario would then result in the following workflow.

1. References are recorded normally with the reference management software (such as Zotero).
2. All references in the management system library are exported to MODS format.
3. The MODS content is copied into the DITA re-use topic.
4. The individual reference items (`<mods>` nodes) are transcluded into a publication's bibliography DITA topic.
5. When the publication is published, a bibliographical format is selected. This results in the references in the output taking on the required reference format, such as *Harvard.*

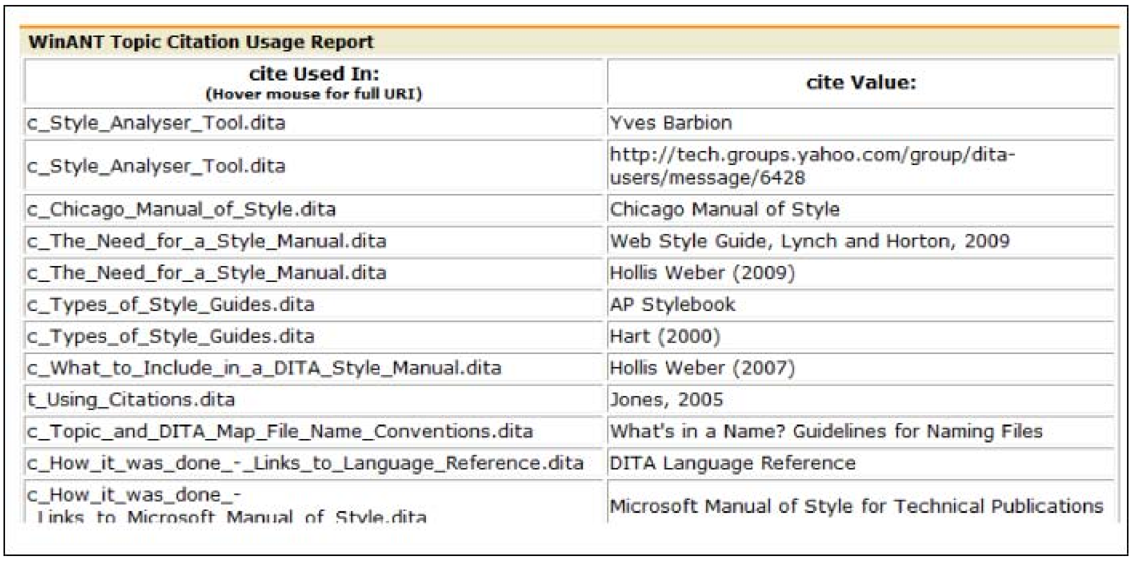
The only technical element that would prevent this scenario from working is that the MODS documents exported from Zotero do not have a unique identifier that could be used as a basis for conreffing. However, the Zotero RDF format does include such a number, so it may be possible to implement the suggested workflow using RDF instead of MODS.

This leaves the problem of citations. The conref mechanism could be again used, so that any citation is drawn from a re-use topic. The filtering idea could be applied in the same way as for bibliographical references, with a sample code block being:

```
<cite>
 <ph platform="APA">(Peters, 2007)</ph>
 <ph platform="Harvard">(Peters 2007)</ph>
 <ph platform="Nature">16</ph>
</cite>
```

An alternative approach would be to develop a specialised DITA information type for citation and bibliographic references.

Using `cite` elements and transcluded reference lists in base DITA requires manual tracking and management of citations. A DITA author would find useful an automated reporting facility that would provide a list of all citations used within a collection. To test the feasibility of such a report, I built such a report into the WinANT Echidna application. This report returns the contents of all `cite` elements used within a DITA publication, and the topic file name in which the citation exists.



WinANT Topic Citation Usage Report

cite Used In: (Hover mouse for full URI)	cite Value:
c_Style_Analyser_Tool.dita	Yves Barbion
c_Style_Analyser_Tool.dita	http://tech.groups.yahoo.com/group/dita-users/message/6428
c_Chicago_Manual_of_Style.dita	Chicago Manual of Style
c_The_Need_for_a_Style_Manual.dita	Web Style Guide, Lynch and Horton, 2009
c_The_Need_for_a_Style_Manual.dita	Hollis Weber (2009)
c_Types_of_Style_Guides.dita	AP Stylebook
c_Types_of_Style_Guides.dita	Hart (2000)
c_What_to_Include_in_a_DITA_Style_Manual.dita	Hollis Weber (2007)
t_Using_Citations.dita	Jones, 2005
c_Topic_and_DITA_Map_File_Name_Conventions.dita	What's in a Name? Guidelines for Naming Files
c_How_it_was_done_-_Links_to_Language_Reference.dita	DITA Language Reference
c_How_it_was_done_-Links_to_Microsoft_Manual_of_Style.dita	Microsoft Manual of Style for Technical Publications

Figure 30: Sample Citation Report developed for WinANT Echidna

Benefits of content re-use

There are many business benefits in re-using content, including cost and time savings in both creation and maintenance stages of the documentation process.

Only having to type information once, and once only, would be more efficient than copying or re-typing. In most style-based document workplaces, the same information is re-typed, copied and pasted, and re-typed again. Microsoft introduced some content re-use features into Word, but these were never adopted wholesale, and were not particularly powerful. Adobe were more successful with re-use with Framemaker's *variables* feature, but it was only practical within one project.

More recently, style-based (but topic-based) authoring tools such as Adobe RoboHelp and MadCap Flare have introduced the idea of *snippets*, where a block of HTML code can be re-used in different places. But again, that is practically limited to re-use within the same project.

Author-it, with its object-oriented documentation approach, has very effectively implemented content re-use, and although not a true, XML structured authoring environment, it does gives us an insight into the benefits of re-use.

The very big XML picture is for the same information to be re-used across projects, across companies, across industries, and across the world. The re-use of Wikipedia articles, where Wikipedia content can be dynamically transcluded into any Web page, is a good example of this ambitious objective.

Interestingly, the efficiency of content re-use is not mainly in the reduced cost of writing. It is in the reduced maintenance cost. If information is re-used in 12 places, then it only has to be updated once.

DITA provides a superb platform for content re-use. The table below shows a few ways in which content reuse can be approached.

Introduction to DITA for UA presentation, Su-Laine Yeo (JustSystems), Andrew Van Conas (Quadralay)

Reuse Opportunity	**Solution**
Multiple similar deliverables	Flag some content as conditional
Piece of content used in different contexts	Include it in different topics using content references
Topic used in many different deliverables	Include it in different deliverables using DITA maps

Using indexes with conditional publishing

Topics in the Artefact sometimes include a rationale section, which explains why and how a DITA usage guideline has been devised.

The `section` element is marked up with an `audience` attribute of contributor. When *The DITA Style Guide* is produced, the conditional publishing removes the rationale section through a *ditaval* setting of `audience="contributor":exclude`. When produced in other forms, however, the section is included normally (though it may be flagged in some way).

Normally, index entries are placed, for convenience, in the `prolog` section of the topic. Index entries relating to content in the conditional rationale section would incorrectly appear in the index for an output where the section was excluded.

For this reason, index entries relating to conditional content should be placed inline in the text (as the first child element of the element with the condition attribute applied).

Writing to STOP

> *A writing methodology known as STOP Sequential Thematic Organisation of Publications was developed at in the 1960s. The purpose was to improve the speed of document production, and to allow multiple authors to work simultaneously on the same document. The STOP approach still resonates today, as we still have the same needs to reduce document creation times and to work collaboratively. In this workshop, we will look at how the STOP approach worked and how it might be re-applied even more effectively in the 21st century.*

The Cold War

The escalation of the Cold War in the 1960s saw massive increases in military spending in the United States, and the rise of *military-industrial complex*. One of the beneficiaries of this increased military spending was Hughes Aircraft Company. The Hughes company had been formed in 1932 by the enigmatic aviator, film director and inventor, Howard Hughes. It had grown to become a massive conglomerate. (At the start of World War II, Hughes had 4 full-time employees; by the end, it had 80,000!) In 1959, the Hughes company opened a radar division in Fullerton, California. It was in the Hughes-Fullerton Ground Systems Group that a new technical writing methodology was devised: the Sequential Thematic Organisation of Publications (STOP).

The Problem

The STOP methodology was devised to overcome a major problem within Hughes-Fullerton. The company had an endless stream of requests for proposals, all of which were extremely complex. One single writer or engineer could not work alone on the proposals, as the *Red Alert* schedule made this impossible. Teams of writers needed to prepare proposals, but the existing writing approaches didn't lend themselves to teamwork (or collaborative authoring, as we know it now).

The authors of proposals needed to answer the questions:

- How do we get the proposal finished on time?
- How do we work collaboratively?
- How do we maintain coherence of the proposal?
- How do we control quality?
- How do we avoid bottlenecks?

The Solution

The solution to the problem was a topic-based approach to writing, where a proposal document was built as a set of modules using a storyboard device as a planning and project management tool. Rather than designing a document into categories, proposal documents prepared under STOP were structured using argumentative or persuasive outlining as a basis.

STOP was developed in the Technical Publications department by three of its “author corps”, James Tracey, David Rugh, and Walter Starkey, who published a paper *STOP - How to Achieve Coherence in Proposals and Reports* in 1965.

***Thematic Quantization* Theory**

The STOP authors understood that coherence is best achieved by the reader being able to recognise “topical units of discourse”, or *topics*. They believed the best way to achieve topic recognition was through the device of uniform modules. Much emphasis was placed on the idea that a group of topics arising from a theme was a better organisation structure than arbitrary rules of “logical categorisation”.

A document written to STOP is made up of uniformly-sized topics, arranged into a sequence of themes.

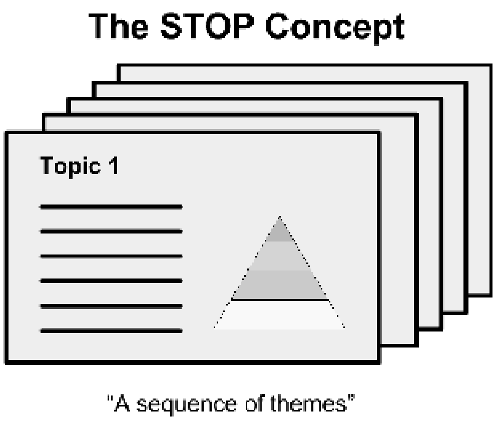


Figure 31: Overview of the STOP Concept

The thematic, topic-based structure of STOP

The uniform size of the topics (“thematic unity”) was important. Topics had to be two pages in size, each must include an illustration, and the required number of words was 500. The fixed physical size of a topic, two pages, was an adaption of the idea of index cards. A document became more modular and flexible if the topic "cards" could be easily shuffled and re-arranged.

Not Just Presentational

The STOP approach wasn't just a method of presentation. STOP also changed the way in which documents were produced. The previous method involved distinct stages in the process, where you would research, then outline, then draft, and then revise. And you could not change this linear order. Headings used a parallel grammatical form because they were part of a hierarchical taxonomy, and reflected subdivisions of a larger grouping of information.

By contrast, the STOP approach defined no particular order to the production. Some topics might be complete, while others were yet to be drafted, and others were under review.

Storyboarding

The document was planned, and the project managed, through a *Storyboard Wall*. This was a public, visual plan of the document, and could be understood as a "collective" outline. This approach to planning allowed revisions to the publication to be made before the writing proper had even begun!

The storyboard wall permitted individual authors could see where their topics fitted in the overall document. The topic-based structure allowed each to write independently of other authors, yet everyone was still able to see the "big picture" of the publication.

Key Features

The key features of STOP were:

- Meaningful headings
- Distinct modules, paragraphs, or chunks
- Modules "signalled" by formatting treatment
- Focus on a topical "thesis statement"
- Illustrations supporting the thesis and directly visible in the two page topic

Was it a success?

According to *Weiss (1999)*:

> *STOP, ... in nearly every instance, reduces dramatically the stress of designing, drafting, editing, testing, maintaining and modifying publications - while simultaneously producing the most readable technical documents ever published. It maximises the talents of the best writers, fully exploits (and disciplines) the work of the worst, allows for the early discovery of flaws, supports project management methods and technology... - while simultaneously yielding books that can be read with a minimum of searching, branching, looping.... Moreover, it is a process that can be learned in a week and mastered in about a month.*

What Happened to STOP?

The success of the Hughes-Fullerton approach (they achieved a greater win rate on proposals) led to other companies adopting the storyboarding approach. But over time, and especially with the introduction of computerised writing tools such as word processors, the STOP method faded away. Some ideas can be seen in Word's outlining feature, which made non-linear writing workflows easier. (The outline is a tool used during planning, writing and revision.) Some of the ideas of visual “signalling” of topic chunks were adopted by Information Mapping.

The STOP approach still resonates, particularly as we re-discover topic-based authoring and the need for collaborative authoring. We have similar needs now to the author corps at Hughes.

We can re-interpret the STOP approach to suit a 21st Century documentation workflow by:

- Emphasising visual aids, such as photos, illustrations, videos, cartoons, and diagrams
- Signalling thematic units through presentation devices
- Using a constrained, topic-based architecture such as DITA
- Adopting structured authoring techniques
- Using storyboards throughout the project life cycle
- Using thesis sentences.

STOP and DITA

Short descriptions and abstracts can be used as the basis of document summary topics in DITA, mirroring the STOP approach of assembling topic thesis sentences into a proposal summary.

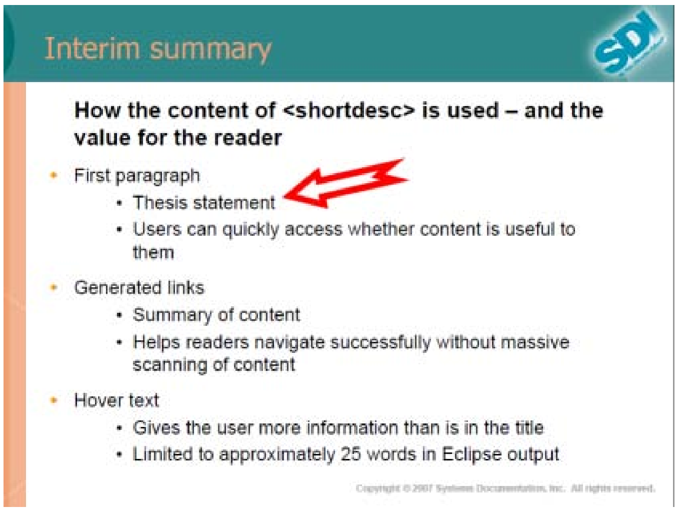
The business factors that led to the development of STOP included the need to work collaboratively, and the need to produce documents more quickly. These same factors drove the development of DITA.

The STOP approach was an early implementation of modular document architecture. As with STOP, modern modular documentation systems like DITA often involve multiple authors, rather than the single author model typical of linear documentation.

One of the key features of STOP is the topic thesis sentence, or thesis statement. When the synopsis of a STOP proposal was required, the thesis sentences of the constituent topics were extracted and assembled to form that overall synopsis.

There is a close parallel between STOP thesis sentences and DITA short descriptions. Kris Eberlein, in a conference paper on best practice for DITA short descriptions (*Eberlein, 2007*), even used the term “thesis statement” in her summary of the content of short description.

Eberlein, 2007



Interim summary

How the content of <shortdesc> is used – and the value for the reader

- First paragraph
  - Thesis statement
  - Users can quickly access whether content is useful to them
- Generated links
  - Summary of content
  - Helps readers navigate successfully without massive scanning of content
- Hover text
  - Gives the user more information than is in the title
  - Limited to approximately 25 words in Eclipse output

Copyright © 2007 Systems Documentation, Inc. All rights reserved.

Figure 32: Eberlein Conference Presentation Slide

To achieve a similar overall summary of a set of DITA topics, the `abstract` or `shortdesc` elements could be drawn into an overview topic, using conrefs or an automated extraction process.

If `shortdesc` elements from multiple topics are to be conreffed into an overview topic, the best practice is to include a single `shortdesc` within the overview topic's abstract, and then conref each of the `shortdesc` elements into a summary topic's abstract element. (The summary's `abstract` would therefore contain multiple `shortdesc` elements.) Using `shortdesc` rather than `abstract` for the conref referenced element is required because a topic can contain only one `abstract` element,

This approach would not work effectively if `abstract` or `shortdesc` elements were excluded from the output.

Another approach would be to create a topic with references to each topic, and rely on the processor's ability to incorporate the `shortdesc` as a topic preview in the output to create some sort of overview summary.

Writing paragraphs

> *DITA authors should avoid long paragraphs by using no more than nine sentences or about three lines, and using alternative devices such as lists and tables where appropriate.*

The DITA methodology embraces minimalism as a philosophy, but the DITA standard itself cannot force authors to adopt a minimalism writing approach. Authors are free to waffle, to add superfluous information, to include redundant text....

Another methodology, Information Mapping, faces a similar challenge to DITA, that of inducing authors to write concisely and to the point. Information Mapping introduced the concept of *information blocks* to replace the paragraph as a unit of discourse. An *information block* can contain one to nine sentences, and are categorised into a taxonomy of about 40 different block types *(Horn, 1999)*.

In a paper entitled *Improving International Communication Through the Containment of Prose Paragraphs*, Edmond Weiss writes that “the paragraph form may be the worst way to present technical information” *(Weiss, n.d.)*. He suggests ways of replacing prose paragraphs with “structured and graphical alternatives”, including:

- lists
- synoptic tables
- decision trees
- flowcharts.

Authors working within the DITA methodology should adopt practices similar to those advocated by Weiss *(ibid)*, and avoid long paragraphs.

References

- Horn, R. E. (1999). Two Approaches to Modularity: Comparing the STOP Approach with Structured Writing. Journal of Computer Documentation.
- Weiss, E. H. (n.d.). Improving International Technical Communication by Containing Prose Paragraphs.pdf (application/pdf Object). Retrieved November 11, 2009, from http://www.edmondweiss.com/PDF/Improving%20International%20%20Technical%20Communication%20by%20Contaning%20Prose%20Paragraphs.pdf

Using journalistic techniques in structured technical communication

Technical communicators may be able to learn from the techniques of journalists, when moving from narrative, style-based authoring to semantic, structured authoring.

Journalism has traditionally taken a different approach to the structure and wording of text than has technical writing. Interestingly, journalism has also largely bypassed the phase in the history of technical communication where content and form merged into the domain of the writer. (Newspaper journalists have always written unformatted text; formatting was the job of compositors, designers and typesetters.) It can be argued that journalistic writing involves more rhetoric than technical writing, as persuasion, or calls-to-action, play a greater role in journalism.

A daily newspaper has a topic-based structure. The definition of *topic* (a stand-alone chunk of information able to be understood in its own right) fits well with the definition of a news story. News stories are stand-alone, and can be read in any sequence. New stories are certainly re-used and re-purposed content; *syndicated content* is a synonym for *re-purposed*.

News stories have an implicit structure, and that structure works well in writing for the Web. According to *Lynch and Horton (2009)*, the *inverted pyramid* method is effective in Web writing.

> *The inverted pyramid style used in journalism works well on web pages, with the conclusion appearing at the beginning of a text. Place the important facts near the top of the first paragraph where users can find them quickly.*

Bleiel (2008) also recommends the inverted pyramid for writing Help systems.

> *...the most important facts first, with the others following in descending order. This gives readers the most pertinent information before they lose interest, and also makes it possible for editors to easily cut the text (if necessary for lack of space) from the bottom up. This is an excellent way to plan and write documentation that can serve the majority of users immediately, and everyone successively over time.*

The *inverted pyramid* method is a practice of presenting the crucial information at the beginning (at the base), followed by important information that is not essential to understanding, continuing to the least important information at the bottom (the tip). The structure is made up of the title, followed by the *lead* (the summary), then the body. This approach *front-loads* the text, allowing the reader to choose when to abandon reading.

Ensuring that the lead is at the start of the topic (following another journalist principle of “don't bury the lead”) also helps guide the writing, in that it provides a thesis statement around which the text can be developed.

Changes in technical communication practice

DITA is forcing changes to technical communication practice. DITA advocate Scott Abel suggests that technical communicators are not taking advantage of these changes.

In the foreword to DITA 101 *(Rockley, Manning, et al, 2009)*, Scott Abel describes the changing role of technical communicators.

> *Technical writers and editors have been forced - like it or not - to move to a more formal method of creating content, often for a global audience. Gone are the days of the free-for-all approach to creating technical documentation products one-at-a-time using desktop publishing tools. While this technique was the best method possible in the 80s and 90s, today, those who create user manuals, online help systems, and other types of documentation are increasingly expected to take a more formal approach to content creation, utilizing content standards like the Darwin Information Typing Architecture (DITA)...*

If you walk into the library at Bond University in Queensland, and at many other modern universities, you will probably be surprised by the lack of books. Libraries have changed from book repositories to knowledge and learning centres. Vast worldwide online book and journal resources, provided by EBSCO, GALE, Google Scholar, Informit, Scopus and Web of Knowledge are a more cost-effective way for libraries to provide and lend written works than physical books on shelves. Free World Wide Web access is therefore now a core service for modern libraries, and the ability to locate specific works in enormous online knowledge databases has become a key skill for librarians. As ambitious open source learning resource projects gain critical mass, libraries will again need to respond to the changing landscape of the Information Age.

> *Over the years, Detroit bosses kept repeating: "We have to make the cars people want." That's why they're in trouble. Their job is to make the cars people don't know they want but will buy like crazy when they see them. I would have been happy with my Sony Walkman had Apple not invented the iPod. Now I can't live without my iPod. I didn't know I wanted it, but Apple did. Same with my Toyota hybrid.*
>
> ***Cars, Kabul and banks, by Thomas Friedman***

The chains of habit are too weak to be felt until they are too strong to be broken.

Dr Samuel Johnson

Scott Abel, in responding to a pointed blog posting by Tom Johnson *(Johnson 2009)*, suggested that technical writers need to get over their “addiction to the idea that the manual is as important as [they] think it is”.

Abel places the blame for the lack of improvement in technical communication practice at the feet of the technical communicators.

Why aren't we talking about delivering content as a service through widgets, social networks, mobile devices and Apps? Why aren't we developing systems that put HELP at the forefront of the system, not the HELP CONTENT we write? Why not allow users to help one another and to participate in the documentation process? Why not make the content shareable, tweetable, embeddable? Why are we not allowing users to tell us what they want and need? Why are we talking about manuals when most technical communication departments are still creating unstructured content using tools designed for word processing and desktop publishing? Why aren't we using component content management systems to create personalized content on demand? Why aren't we using controlled vocabularies to help avoid ambiguity and the resulting challenges terminology-related issues add to the mix[...] And, why aren't we educating our shareholders about the importance of video - the most successful way to SHOW someone HOW to DO something?

Scott Abel

According to Abel, the paper manual should be “killed”, but predicts that a view such as his will result in “manual-loving, desktop publishing-worshipping, style guide-driven tech commers foaming at the mouth”.

Levels of re-use

Content re-use can be implemented at four different levels: document, topic, paragraph and phrase.

Content re-use is implemented in DITA through the *conref* (transclusion) and *ditamap* mechanisms. The depth to which re-use can be implemented is determined by the smallest taggable element. For example, a word cannot be re-used unless it is wrapped in its own tag.

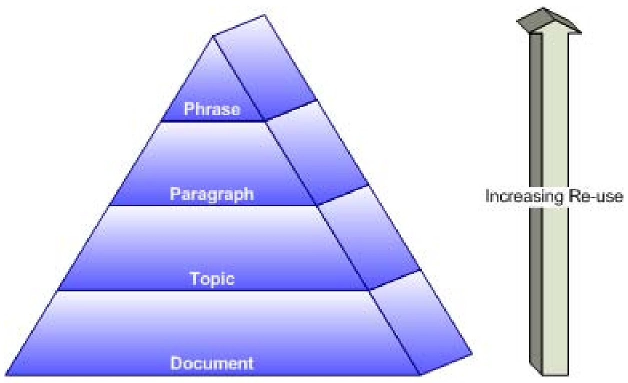


Figure 33: Four Levels of Content Re-use

At the lowest level, content re-use through ditamaps may involves referencing one ditamap in multiple ditamaps. This allows a section of a document to be repeated across multiple documents. This is *document level* content re-use, where the unit of re-use is a ditamap.

Perhaps the most common form of re-use is when one topic is referenced by more than one ditamap, and therefore the same topic is re-used in different content collections. This is *topic level* re-use, where the unit of re-use is a topic.

A block element within a topic, such as a paragraph or note, can be re-used in another topic through a conref transclusion. This type of re-use is *paragraph level* re-use, where the unit of re-use is a block element.

The finest re-use granularity is *phrase level* content re-use. This involves using conref transclusions to re-use words or phrases (or inline elements) from one topic in other topics, where the unit of re-use is an inline element. Examples of this sort of re-use includes product names, trademarks, and user interface labels. Phrase level re-use should be limited to grammatically complete sentences and proper nouns *(Zydron, 2004)*.

Re-used phrases can be used within re-used paragraphs. Likewise, re-used paragraphs can be used within re-used topics, and re-used topics can be used within re-used ditamaps.

DITA schema declarations

> *XML documents usually contain a reference to the schema or standard to which they are written. That reference is called the schema declaration.*

Schema declarations look like:

```
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN"
"dtd/concept.dtd">
```

The purpose of the declaration is to tell the program reading or processing (*parsing*) the file what sort of document it is (and sometimes what version of the applicable rules it is using).

The parts of the DITA declaration above are:

`concept`	The type of document, which corresponds to the root element of the document.
`PUBLIC`	Can be PUBLIC or SYSTEM. PUBLIC is used in conjunction with the *Formal Public Identifier* (see below). SYSTEM indicates the DTDs are located
`"-//OASIS//DTD DITA Concept//EN"`	The *Formal Public Identifier (FPI)* of the schema, ending with the language. If the computer processing the XML file has a record to indicate where the DTDs are for this exact *FPI*, it will use those DTDs to parse the file.
`"dtd/concept.dtd"`	The relative or absolute URI of the schema file. For a PUBLIC declaration, the computer processing the XML file will only reference this URI if it has no record of where the DTDs are for the nominated *FPI*. For a SYSTEM declaration, the URI is the only place it can look to find the DTDs.

So in short, if the *DOCTYPE* has an *FPI*, and the *FPI* is recognised, then the path to the DTD is ignored.

Mapping the *FPI* to a local copy of the DTD is done by *resolver* software. DITA OT, for example, uses the Apache Resolver. Apache Resolver looks up the mappings in a file called `catalog-dita.xml`. (The file that maps the *FPIs* to DTDs is called a catalog.)

Constraints in DITA 1.2

Constraints are an architectural enhancement to DITA, introduced in DITA 1.2.

A constraint defines a set of restrictions that conform to the rules of specialisation. (The constraints mechanism was originally proposed as *replacement domains*.) Constraints aim to provide a more simplified authoring environment by restricting the mark-up options available to DITA authors to conform to business rules.

A constraint can remove an optional element from a content model, make an otherwise optional element required, and modify the attributes of an element by adding attributes or making them required. However, a constraint cannot make a mandatory element optional, and nor can it make a mandatory element optional.

Constraints are similar in purpose to specialisations, but attempt to avoid the perceived technical complexity of specialisation. They also allow non-semantic structural changes to be made without the specialisation mechanism, which was designed more for semantic changes.

A typical constraint might apply the following business rules to an authoring environment:

- Remove the `ph` element, and prefer its `filepath`, `userinput`, and `systemoutput` specialisations instead.
- Remove the `keyword` element, and prefer its `cmdname` and `varname` specialisations.
- Remove the ability to nest block elements such as `ol` and `table` within `lq` and `p` block elements.
- Make `shortdesc` a required element.

A constrained topic will always be valid as an unconstrained topic, as constraints only restrict what it permitted, not extend what is permitted. For example, if a constrained concept content model prevented `section` elements in a `concept` topic, a constrained concept topic would still be a valid concept topic, because the base content model allows a concept topic without a `section`.

Elements of a constrained topic can be transcluded (*conreffed*) into a base (unconstrained) topic, but elements of a base (unconstrained) topic may not be able to be transcluded (*conreffed*) into a constrained topic if the source topic would not conform to the constraint rules. In other words, elements from a less restrictive document cannot be incorporated into a more restrictive document, regardless of whether the conreffed element itself would be valid in the target (constrained) topic.

Constraints are specified, or declared, in attributes in the topic element, and reference the constraint rules specified in DTD or XSD modules. There are two constraint attributes: `constraints` and `constraint-scope`. A constraint stored in `shortdescReqConstraints.mod` is declared in the topic as `shortdescReq-c`.

Example of a constraint declaration

A constraint to make the audience attribute required, to specify a list of valid values for the audience attribute, and to make the title element within a `shortdesc` element required, would look like:

```
<reference ...
        audience="administrator"
        constraints=" db-audience-enum-c
topic-audience-req-c shortdescReq-c"
        constraint-scope="db-audience-enum-c(
topic/topic ... )
            topic-audience-req-c( topic/topic )
            shortdescReq-c( reference/shortdesc )">
```

Filtering and flagging content in DITA

Filtering of DITA content during conditional publishing process can be achieved through metadata attributes in the topic, or in the topicref elements in the ditamap. Topic references can also be omitted from the ditamap as an alternative approach. These and other filtering and flagging issues require guidelines as to which approach is best.

One of the strengths of DITA is its in-built support for content filtering. This means that text (phrases, paragraphs, topics or ditamaps) can be given metadata attributes, which can later be used as a basis for inclusion or exclusion in a deliverable.

For example, a collection of training material might be generated as a student's guide **and** an instructor's guide. In this case, the instructor's guide is essentially the student's guide with the addition of notes for the instructor at various places in the content. Content only intended for the instructor's guide could be marked with a metadata `audience` attribute of `instructor`. When the student's guide is generated, a processing instruction of “exclude anything with an audience of instructor” is passed. When the instructor's guide is generated, a processing instruction of “flag anything with an audience of instructor” is passed, specifying that any `instructor` content gets output with a mortar board icon next to it.

In some cases, you may want to filter entire topics. This can be done in two ways:

- applying the metadata attribute in the topic
- applying the metadata attribute in the ditamap

So which is the correct method? It depends on circumstance.

If you were constructing different ditamaps for the two guides, in this example, you wouldn't even need to use the audience attribute... you'd just omit the instructor topics from the student ditamap.

If you were using a common ditamap with the filtering specified in a *ditaval* (filtering) file, and were intending to use the instructor topic in other collections (such as a training centre manual), you may prefer to apply the metadata attribute in the topic. In other cases, it would be appropriate to apply the attribute to the topicref in the ditamap. Because some DITA processors, including the DITA Open Toolkit, produce an error when filtering a topic based on the topic metadata, whenever you apply an attribute to a topic you should also apply the attribute in the topicref in the map.

If you choose to apply the attribute to the topicref element in the hierarchy section of the ditamap, the topic may still be included in the output if the reltable also includes a topicref to the topic. The topicref itself will not be output in the form of a node in the output TOC, but the topic may be accessed from the search, an index, or from links generated from the reltable relationships.

It is possible in DITA to have nested ditamaps. Although it does depend on circumstance, it is generally better practice to filter in the `mapref` element. The reason for this, and a point that should always be considered when deciding on a filtering strategy, is the reduced processing load during generation of the output. If the processor has to read in a topic, or a nested map, only to find it's not needed in the current build because of the filtering metadata, it is an inefficient model. It is much better for the exclusion to be realised by the processor without needing to load the file.

Variables: conrefs or attributes?

A *variable* is an often used phrase (such as a product name) that is re-used by replacing the value of the variable at the time the information is published.

For example, a publication may be generated from the same source in two forms: once as a manual for Product A, and then as a manual for Product B. At publish time, the product name to use is updated at a single point, and all instances of that product name variable are updated to reflect the updated product name.

When conditional publishing is used for variables, whenever the product name is needed, all the different product name options are coded, each with a separate product attribute. At publish time, filtering through the *ditaval* file is used to exclude all product names except the one valid for the particular publication. When conrefs are used for publishing, a separate variables topic is set up, and whenever a product name is needed in a content topic, it is conreffed in from the variables topic. At publish time, the variables file is updated with the current product name (or a replacement variables topic "swapped in"), and all references to the product name are updated for publishing. *Figure 34: Variables through Variables File and conref* (see page 146) and *Figure 35: Variables through Attributes and ditaval* (see page 146) illustrate the two approaches.

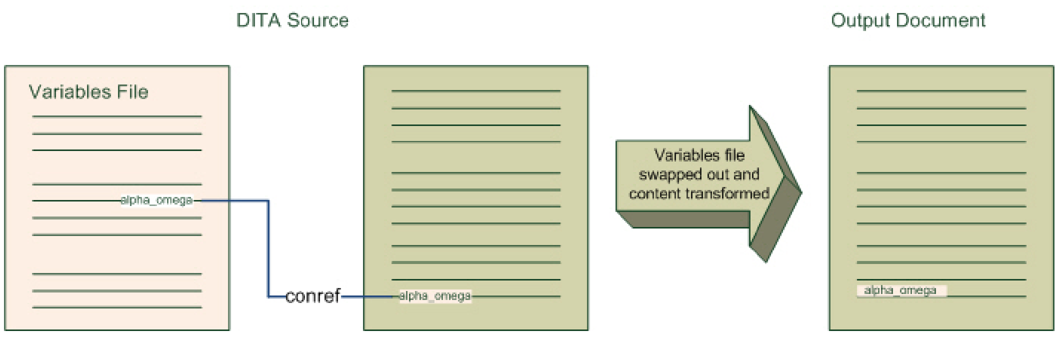


Figure 34: Variables through Variables File and conref

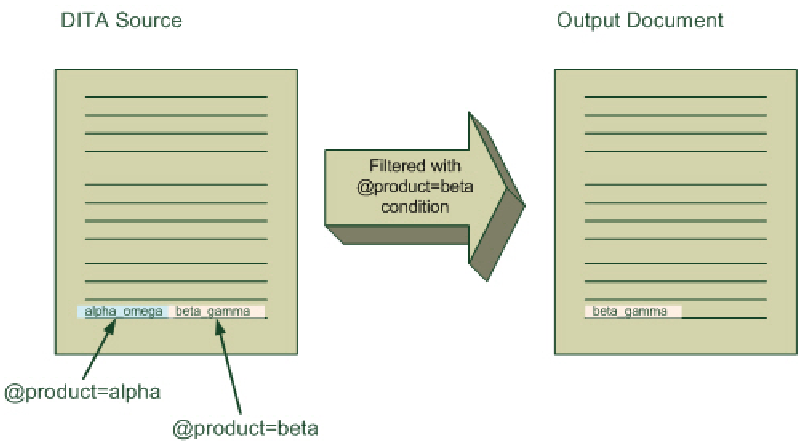


Figure 35: Variables through Attributes and ditaval

When writing for localisation, an added complication when using conref variables is that some languages use different forms of a word when it is used in different parts of speech. In other words, a word may be spelled the same in English whether it is used as a verb or a noun, but may be spelled differently as a noun and a verb in a more inflected language.

A simple example is the word for shoe in Irish language (Gaeilge). The word is spelled as "brog" when used independently, but when the previous word is a preposition, it changes its form to "bhrog".

- an brog = the shoe brog = a shoe mo bhrog = my shoe.
- Páras = Paris i bPáras = in Paris Aontas Phháras = Union of Paris

One recommendation (from the OASIS DITA Translation Subcommittee) is that if you use variables for product names and other technical terms, use them as subjects of the sentences in which they occur.

Tip: DITA 1.2 introduces the keyref mechanism, which provides a better option than the above two.

DITA and semantics

DITA is a language, with syntax and semantics. An understanding of the meaning of semantics helps understand the theoretical underpinnings of communication, and specifically structured authoring, with DITA.

All language is structured, whether human (or *natural*) languages, animal languages or computer languages. DITA is at one level an XML-based computer language, but more specifically, a text mark-up language. It allows the semantics of the textual components to be store alongside the content.

Language is made up of syntax and semantics. Syntax is the mechanics of the language, or the language rules. Semantics is the meaning that the language communicates. All language has structure; the structure of a natural languages is defined by its syntax and semantics.

Concepts such as nouns, verbs, grammar, objects and subjects are part of the syntax of a natural language. Concepts such as tags, delimiters and tokens are part of the syntax of mark-up languages.

Concepts such as directions, height, names, admonishments, and imperatives are examples of semantics in a natural language. Concepts such as steps, warnings, reference, date and owner are examples of semantics in a mark-up language. Such semantics are captured as metadata. When a language has a very detailed set of semantic metadata, it is said to be *strongly typed*; languages that cannot store a lot of semantic detail are *weakly typed*.

DITA allows the semantics of text written in a natural language to be captured so that those semantics can be used as a basis for computer processing and manipulation of the text. DITA semantics are at a much higher level than natural language semantics. For example, each word in a natural language has meaning, and sequences of words also contain meaning. However, not each word in a tract of DITA mark-up has specific semantic metadata; some phrases and most text blocks (such as paragraphs) are semantically identified.

In other words, natural language, however expressed, is full of semantics. Semantic mark-up languages capture some of those semantics.

DITA semantics define the structure (the hierarchy and sequence of information) as well as the explicit meaning (domain mark-up). The DITA vocabulary is helpfully segmented into *structural* elements and *domain* elements.

Escape from Author-it

Although Author-it is a modular, topic-based, single-source authoring environment, with some support for structured authoring, it is a proprietary system. Extracting content from Author-it's clutches into open source DITA format proved not to be easy.

The Problem

In 2003, I started using Author-it as a content management system for content related to my work within the Swinburne University technical communication programme. The tool allowed me to single-source Web, Help and PDF output for class notes, and allowed me to re-use elements of one class into that of another class. Over time, the topic repository grew larger as I taught or designed more subject material. In 2007, I started compiling information to use as the basis for the new HATC424 Structured Authoring with DITA subject, as well as for commercial training DITA workshops and for a proposed PhD project relating to DITA.

The new content relating to DITA was not created in DITA because at the time, tools were not sufficiently evolved to provide an efficient authoring environment. Furthermore, I was keen to maintain a single repository of learning content, with the aim that eventually I could migrate all Author-it content to DITA. Indeed, Author-it did promise a pathway to DITA.

In 2009, I started to make specific plans for migration from DITA to Author-it, and discovered that the promised pathway was not as obvious or straight as I had hoped.

The Promise and Limitations of Author-it's Structured Authoring Approach

Author-it has for some years promoted its DITA support, which was initially a simple ability to output to *ditabase* information types. This feature had little practical application, as DITA is a storage format, not an output format, and *ditabase* is designed as a tool for specialisation, rather than a content model in its own right. (It is too weakly typed to provide much advantage over XHTML.) In version 5.2, Author-it introduced a set of structured authoring features, along with the promise of Author-it becoming a viable DITA authoring platform.

Upon release, the limitations of the Author-it content model became apparent. The structured authoring feature only seemed to be suitable for a simple, flat, two-level content model. For example, I can create a rule and nominate paragraph styles that are allowed in that rule, and permitted character styles within a paragraph style rule. That design only works a content model such as:

- concept topic
    - containing shortdesc styles
    - then paragraphs
    - or list items
    - or a table.

Anything more complex, such as rules with nested block elements, were not possible. (For example, a rule that a note can only occur within a step, or a step paragraph that is structured into command and response.)

Author-it 5. 2 did not ship with any rule sets, and it would seem a monumental task to write a ruleset that approximated a DITA concept topic. To be practical, Author-it would have to come shipped with pre-built rulesets, or have some ruleset object that would allow importing and exporting. Fortunately, this was addressed in version 5.3, when four extremely simple DITA rulesets were provided, and the ability to import and export rulesets (as part of the topic object) was added.

A new grouping rule type in version 5.3 also opened up the possibility of some nesting of rules, allowing a structure such as steps/step/cmd/info/stepresult to be defined. However, inline styles can not be nested (required to permit a phrase element to contain a trademark, for example), and nor can elements containing data be nested (required to permit a paragraph to contain a list or a code block, for example).

Table functionality is also problematic. Because Author-it does not architecturally separate content from form, the editing interface does not restrict table formatting functionality, such as cell background colour, in the structured authoring mode. (Even outside tables, this problem occurs; for example, the editing interface allows line breaks - which have no place in DITA - to be used.) Formatting code will either be lost during export to DITA, or incorrectly interpreted during export.

Further limitations included insufficient metadata fields to match those in DITA; no provision was made for nested metadata structured such as prolog elements, which contain copyright, author, critical dates, etc. In any case, this topic level metadata is not transformed into the requisite DITA mark-up structures. In other words, while Author-it does allow metadata to be stored in its own variables, that variable information is not output to DITA. It could be therefore argued that Author-it offered a rival and equivalent approach to DITA, but not that it was a DITA environment.

Escape to DITA through the Structured Authoring Features

There are some bugs in Author-it 5.2 and 5.3 which complicate the setup of *Related Topics* groups, which I had used within my Author-it topic templates. To set up a new structured template therefore required some work-arounds. (*Related Topics* groups no longer inherit from template to template, and new groups cannot be copied from existing groups within a structured template. Further, there was a problem with using the same names for inherited and template-specific groups.) The problem was resolved making the structured topic the top-level template, and re-creating (not copying) the `More Information` and `Breadcrumbs` groups.

There are structural incompatibilities between Author-it 5.2 and DITA. Author-it is very two-dimensional, in that it doesn't support nested content blocks. For example, it doesn't support a topic containing sections containing paragraphs. It only supports inline mark-up within block mark-up within topics. This becomes a critical problem when attempting to apply a DITA figure structure. The simplest DITA figure structure of:

```
<fig>
  <title></title>
  <image></image>
</fig>
```

could only be represented in Author-it as:

```
[para object with figure heading style]
[image object]
```

A more complicated DITA figure structure, with constructs such as figure groups, and paragraphs or tables within the figure, are impossible to even attempt to replicate in Author-it.

In Author-it 5.3, a small number of these shortcomings were addressed by the addition of some add-on variables and templates, which affect the processing of the DITA output. For example, a topic template of `DITA Concept` can be applied to a topic, and this template introduces structure rules and a <*dita-topic-type*> variable to the topic. On publishing to DITA, Author-it uses the variable value to determine which DITA information type to use for that topic. A mapping of a small number of Author-it styles to DITA elements allows simple topic structures to be generated as valid DITA XML. The simple figure structure can be generated using a clumsy arrangement where an image in a paragraph style of image, followed immediately by a paragraph with a style of caption will be translated into a <fig><title/><image/></fig> nested structure.

There are two primary sources of the Author-it limitation: the underlying architecture of the Author-it database, and Author-it's two-stage DITA publishing approach which converts first to XHTML and then to DITA.

Author-it stores its objects in an Access or SQL Server database. The bulk of textual content is stored in XML format in *text* or *memo* fields within the *topic object* data tables. The XML format is a proprietary Author-it design, the schema of which is based on the database structure. For example, the `topic` database table has fields of `CreatedBy`, `CreatedDate`, `Description`, `GUID`, `Text`, etc, and the Author-it XML format has corresponding elements of `CreatedBy`, `CreatedDate`, `Description`, etc. The database structure is a lot flatter than DITA's heavily nested element structure, and this incompatibility results in some limitations in Author-it's ability to interchange with DITA.

When DITA is generated from Author-it, the publishing routine makes two passes over the output. On the first pass, it processes the topics as XHTML, but names the files with a .dita extension. Topic object 9999 will become `9999.dita` in the output. A paragraph in a style of "Note" in an Author-it topic becomes `<p class="note">` in this first pass. On the second pass, the publishing process converts the XHTML to DITA. To do this, it uses a modified, open-source XSL-T file originally developed by IBM. The original file was h2d.xsl. It is shipped with Author-it as AuthorIT-DITA.xslt. The H2D *transformers* are a standard *XHTML to DITA* converter. H2D look for the `<p class="note">` in the example above, and converts it to `<note>`. The resultant DITA file also takes its name from the topic object number, but is prefixed with a "D". (The XHTML file `9999.dita` is thus converted to DITA file `D9999.dita`.) Author-it has modified the H2D code to suit the inherently flat structure of its HTML output.

Any inconsistencies in Author-it's XHTML generation leads to further inconsistencies in the DITA output. For example, Author-it does not map a style of `Cite` to the standard HTML `<cite>` element, but instead generates it as `<p class="cite">`. This mark-up becomes `<p outputclass="cite">` in the DITA transformed from XHTML; this mark-up is valid, but semantically wrong. To adjust the transformation process, an user must manually adjust the custom H2D XSL-T transformation file.

The Door

Ultimately, the pathway to export from Author-it to DITA became apparent. The process involved the following steps.

1. Create a new media (section) object (named `Swinburne DITA Section`), based on the Author-it `DITA Section` media object, but referencing the Swinburne HTML Web template. (This would allow content to continue to be generated to the same HTML Help output types during the transition to DITA.)
2. Create a new topic template, based on the Author-it `DITA Concept` topic template, with the same **RelatedTopics** settings as the previous non-structured template, and associated with the new media object.
3. Change the template used by the Structured Authoring topics to the new topic template (named `Swinburne DITA Concept`).
4. Build the rules for the Swinburne DITA Concept template to closely resemble those of a DITA concept topic.
5. Add new style objects to use in association with the DITA concept ruleset.

6. Over time, use the validation feature within Author-it to ensure that Structured Authoring topics complied with the DITA concept ruleset.

   Note: During this time, I could continue to generate HTML Help and XHTML output with no apparent difference in the output.

7. When ready to migrate content to DITA, remove the reference to the Swinburne HTML Web template in the `Swinburne DITA Section` media object. (This prevents HTML-specific code being included in the DITA output.)
8. Publish the required book objects as DITA.
9. Move the published book and topic objects to an `Obsolete` folder, and carry out all future content maintenance in a DITA authoring environment (ie, outside Author-it).

DITA and Content Management Systems

Some DITA experts advise that a CMS is necessary for authoring in DITA.

Scott Abel (of Content Wrangler) has observed that:

> *DITA without a content management system is dangerous - you need to assure the validity of content and prevent invalid content being checked into the repository.*
>
> ***Abel, 2006***

So what does Abel mean by this? Why does authoring in DITA seem to need a management system where working in Microsoft Word does not?

The answer to this is "complexity". Linear, style-based writing using tools such as Microsoft Word have limited project sizes. A project is normally one document. Even topic-based authoring tools such as RoboHelp have a relatively limited project size. With DITA, however, the "big picture" is to collect all the documentation for an entire company into a repository, and select *collections* to publish as required. Further, this repository will have many interconnected re-usable content blocks. The greater the content re-use, the harder it is to keep track of all the content.

Object-oriented documentation tools such as Author-it have approached this issue by building content management into the authoring tool. The latest version of Author-it includes a phrase-matching intelligence, so as the user starts typing text that is similar to text already written, the tool locates that other text and suggests re-using it.

DITA is a standard, an architectural approach and a methodology, but it is not a tool. We should expect that DITA authoring tools will eventually offer great content management features that will improve the productivity of working within DITA. Currently, many DITA tools

do offer integration with version control systems, but full, comprehensive content management is not yet on the horizon.

The Artefact and Miller's Law

> *Miller's Law suggests that recall of information is best achieved when the data is chunked into groups of approximately seven. The information architecture of the Artefact follows Miller's Law where possible.*

In technical communication, the so-called *Magic Number* design is used commonly as an information architecture tool. Based on Miller's Law *(Miller, 1956)*, the design approach involves structuring the major elements of a document into groups of seven plus or minus two. A manual may therefore be designed with seven parts, each with seven chapters, and so on. Likewise, procedures are designed with the objective that every task be broken down into five to nine steps.

As Miller's Law forms a theoretical underpinning of other documentation methodologies such as Information Mapping, it is a logical extension to apply the approach to the design of documents in a DITA workflow. This information architecture of the Artefact therefore applies, where possible, Miller's Law.

In considering the applicability of the *Magic Number* to hypertext documents such as Help systems, *Ellison (2007)* suggested that Miller's Law be approached with scepticism. He considered that the optimum number of items in information design depends on factors including:

- audience type
- audience language, vocabulary and domain knowledge
- the type of action being described.

These considerations have also been made in the design of the Artefact.

Approaches to procedural topics

> *The task information type in base DITA provides a structure for procedural, and to a lesser extent, process information. However, its content model doesn't suit all types of procedural discourse.*

The `task` DITA information type provides a strongly-typed structure for procedural and process information. In overview, it has a block data structure of title, pre-requisites, context, steps, result, example, and post-requisites. Simplified, the steps block is made up of steps comprising command, step information, step example, and step result.

This tight structure suits *streamlined-step procedures (Farkas, 1999)*, which are perhaps the most common procedure design patterns. The *streamlined-step* model encourages authoring efficiency, minimalism, and consistency. Farkas *(1999)* describes the defining characteristics of *streamlined-step* procedures as:

- brief steps
- simple formatting
- action statements built around an imperative verb
- very little information leading into the steps
- hypertext links used to layer the procedural information (in electronic presentation)

Farkas *(1999)* argues that the brevity and simple format of *streamlined-step* procedures make them “easy to write, format, and localise”, and make it “relatively easy for a large team of writers to follow a style guide and achieve consistency in their work”.

Other forms of procedural discourse design include:

- *rich step* procedure
- *playscript* procedure
- flowchart procedure
- *wizard*
- *paragraph format* procedure

Making decisions to remove context

> *Separating context from content when writing in DITA is a challenge that is best met by a change of mindset.*

In writing the Artefact, I found that removing context was certainly a mindset change, and required a willingness to remove words or phrases that would otherwise provide clarification.

I reflected on my decision-making process when writing the *Specialised element naming convention* topic in the Artefact.

My original text was:

The Artefact is designed so that rules are clearly separated from any rationale, discussion or background, with rules marked with an `importance` *attribute set to high. This attribute selection will permit the rule to be highlighted in the output in some way. However, a secondary benefit is that the rules can be extracted into a form of checklist or summary of The DITA Style Guide.*

In my original text, the discussion (about what is technically valid) and the rule (best practice) were in the same paragraph. To permit highlighting of the rule, the paragraph needed to be split to become:

It is technically valid for the specialiser to use all lower case, lower case with hyphens, camelCase, lower case initial, or some other permutation.

However, best practice is to use all lower case characters, with hyphens separating compound name.

If paragraphs with a high `importance` attribute are able to be re-used in a (different) checklist context, the paragraphs must read independently of others. The rule sentence, starting with the conjunction “however”, does not read correctly out of the context of the preceding discussion paragraph. So that the paragraph could serve as a stand-alone rule, I had to delete the “however”. The paragraph then became:

Best practice is to use all lower case characters, with hyphens separating compound name.

I felt that deleting “however” caused the two paragraphs to lose a little sense, as the transition information that joined the second paragraph to the first was lost. I made the decision that the loss was not significant enough to warrant the loss of re-use opportunity. Leaving “however” in place would render the stand-alone use of that rule paragraph impossible.

The resultant text will read satisfactorily in the contexts of both paragraphs appearing together, and the second (rule) paragraph appearing independently.

Improved glossary and terminology handling

> *DITA 1.2 introduces significant improvements to the way in which terminology is handled. A new glossary entry element provides a flexible semantic structure to store terminology information, and the new indirect linking functionality allows terms occurring in the content to be linked through a key reference or transcluded using a new* `abbreviated-form`

> *element. Keys used for key referencing are defined in a new `glossref` specialisation of `topicref` in the map.*

Purpose of a glossary

The purpose of a glossary is to allow readers who encounter terms that they do not understand to look up a definition for the term. Terms may be single words or longer phrases, they may be acronyms or abbreviations, or they may be domain-specific jargon. In a printed publication, a glossary (usually referred to as a *Glossary of Terms*), appears at the front or back of the book content. In an online document, the glossary is often accessed from a link on the document toolbar and typically displays in a separate window or frame from the document content proper. Many online documents make the glossary easier to access by hyperlinking terms in the content to the associated definitions in the glossary.

Purpose of the change to glossary-related elements in DITA 1.2

The new elements introduced in DITA 1.2 will make it easier for publishing tools to automatically hyperlink terms appearing in the text with their respective definitions in the glossary (in online document outputs). The new elements will also make it easier for users to semantically identify terms and their definitions through a new `glossentry` information type.

The `glossentry` is a specialisation of the concept information type.

A new `glossref` element has also been introduced to provide for the possibility of linking terms in the document content with the glossary definitions.

There are three distinct processes in setting up a glossary in DITA.

- Creating individual glossary topics (using the `glossentry` information type)
- Incorporating the glossary topics into a ditamap
- Linking terms in content topics to their glossary definitions.

Benefit to users

The new glossary-related elements will open up opportunities, through improvements in publishing tools, for the following features in output documents:

- Automated pop-up or expansion linking of terms in the content to definitions in the glossary.
- Generation of a rich and comprehensive Glossary of Terms for a printed document.
- Generation of a dynamic Glossary of Terms (perhaps sortable and searchable) for an online document.

Another benefit will be the ability to use a term in an abbreviated form (for example, an acronym) or in a longer form, yet still identify the relationship between the term and its definition.

The `glossentry` information type

The `glossentry` information type contains an individual glossary entry which is made up of the following elements.

Element	Element Name	Provides	Example
Glossary term	`glossterm`	The preferred form of the term.	`<glossterm> USB flash drive </glossterm>`
Definition	`glossdef`	The definition or explanation of one sense of the term.	`<glossdef> A small portable drive. </glossdef>`
Body of the glossary	`glossBody`	Container for more details about a glossary term (such as part of speech or additional forms of the term)	`<glossBody><... /></glossBody>`

In turn, the optional `glossBody` element contains the following elements, used to store more detailed information about the glossary entry.

Element	Element Name	Provides	Example
Surface form of term	`glossSurfaceForm`	The unambiguous full form of the term (which may include the acronym or short form in parentheses). The surface form is suitable to introduce the term in new contexts	`<glossSurfaceForm> USB flash drive (UFD, or flashie) </glossSurfaceForm>`
Part of speech	`glossPartOfSpeech`	The part of speech of the preferred form of the term.	`<glossPartOfSpeech value="noun"/>`
Symbol or icon	`glossSymbol`	Reference to an image file used as a synonym for the term.	`<glossSymbol href="ufd_logo.jpg" scope="local"> <alt> Identification logo for USB flash drives </alt> </glossSymbol>`
Notes on scope of usage	`glossScopeNote`	Information on what the term does or does not apply to.	`<glossScopeNote>Not to be used for other flash memory cards that do not use a USB interface</glossScopeNote>`
Status of term	`glossStatus`	Business rule recording the status of the use of the term. The status is usually explained with `glossUsage`.	`<glossStatus value="obsolete"/>`
Correct usage of term	`glossUsage`	Supplementary information explaining the correct use of the term.	`<glossUsage>Do not use the term in title case (as in "USB Flash Drive") because that suggests a trademark.</glossUsage>`

Element	Element Name	Provides	Example
Variants of the term	`glossAlt`	Container for alternative representations of the glossary term (such as an acronym used for the term)	`<glossAlt><... /></glossAlt>`

In turn, the optional `glossAlt` element contains the following elements, used to store more detailed information about the glossary entry.

Element	Element Name	Provides	Example
Abbreviated term	`glossAbbreviation`	An abbreviated form of the term	`<glossAbbreviation> Flash </glossAbbreviation>`
Acronym	`glossAcronym`	An alternative form of the term as an acronym	`<glossAcronym> UFD </glossAcronym>`
Association with another alternative form	`glossAlternateFor`	Cross-reference to a similar variant of the same term.	`<glossAlternateFor href="#usbfd/memoryStick"/>`
Short form of term	`glossShortForm`	A short form of the preferred form of the term, but not an acronym	`<glossShortForm> flashie </glossShortForm>`
Status of variant	`glossStatus`	Business rule recording the status of the use of a variant of a term. The status is usually explained with `glossUsage`.	`<glossStatus value="prohibited"/>`
Notes on usage of the term	`glossUsage`	Supplementary information explaining the usage of the variant of the term.	`<glossUsage>This is too colloquial.</glossUsage>`
Synonym	`glossSynonym`	A synonym of the glossary term.	`<glossSynonym>memory stick</glossSynonym>`

The `glossgroup` information type may be used to collect multiple glossary entries into a single topic. It serves a similar purpose to the *ditabase* information type, being a nesting container for `glossentry` topics.

Example of a complete `glossentry`

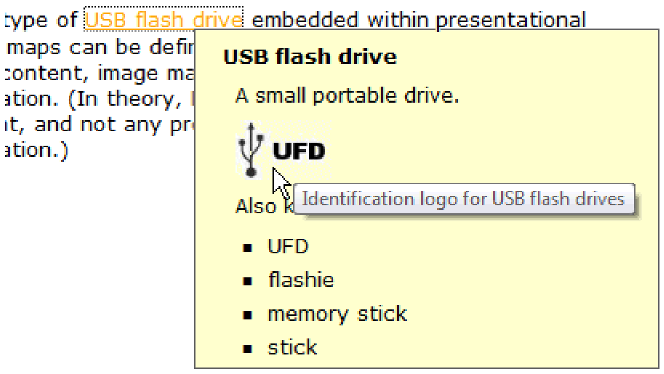
The following is a glossary entry for *USB flash drive*.

```
<glossentry id="usbfd">
  <glossterm>USB flash drive</glossterm>
  <glossdef>A small portable drive.</glossdef>
  <glossBody>
    <glossSurfaceForm>USB flash drive (UFD, or
flashie)</glossSurfaceForm>
    <glossPartOfSpeech value="noun"/>
    <glossUsage>Do not provide in upper case (as in
"USB Flash Drive") because that
    suggests a trademark.</glossUsage>
    <glossSymbol href="ufd_logo.jpg" scope="local">
```

```
       <alt>Identification logo for USB flash
drives</alt>
    </glossSymbol>
    <glossScopeNote>Not to be used for other flash
memory cards that do not use a USB
      interface.
    </glossScopeNote>
    <glossAlt>
      <glossAcronym>UFD</glossAcronym>
      <glossUsage>Explain the acronym on first
occurrence.</glossUsage>
    </glossAlt>
    <glossAlt>
      <glossShortForm>flashie</glossShortForm>
      <glossUsage>Usually describes low cost
devices.</glossUsage>
    </glossAlt>
    <glossAlt id="memoryStick">
      <glossSynonym>memory stick</glossSynonym>
      <glossUsage>This is a colloquial
term.</glossUsage>
    </glossAlt>
    <glossAlt>
      <glossAbbreviation>stick</glossAbbreviation>
      <glossAlternateFor href="#usbfd/memoryStick"/>
    </glossAlt>
    <glossAlt>
      <glossAbbreviation>flash</glossAbbreviation>
      <glossStatus value="prohibited"/>
      <glossUsage>This short form is
ambiguous.</glossUsage>
    </glossAlt>
    <glossAlt>
  </glossBody>
</glossentry>
```

How a glossary entry might be rendered

Publishing tools may provide different methods of rendering glossary elements. The following shows an example of how a publishing tool may render a glossary entry in HTML form.



type of USB flash drive embedded within presentational
maps can be defir
content, image ma
ation. (In theory, I
it, and not any pr
ation.)

USB flash drive

A small portable drive.

UFD

Identification logo for USB flash drives

Also k

- UFD
- flashie
- memory stick
- stick

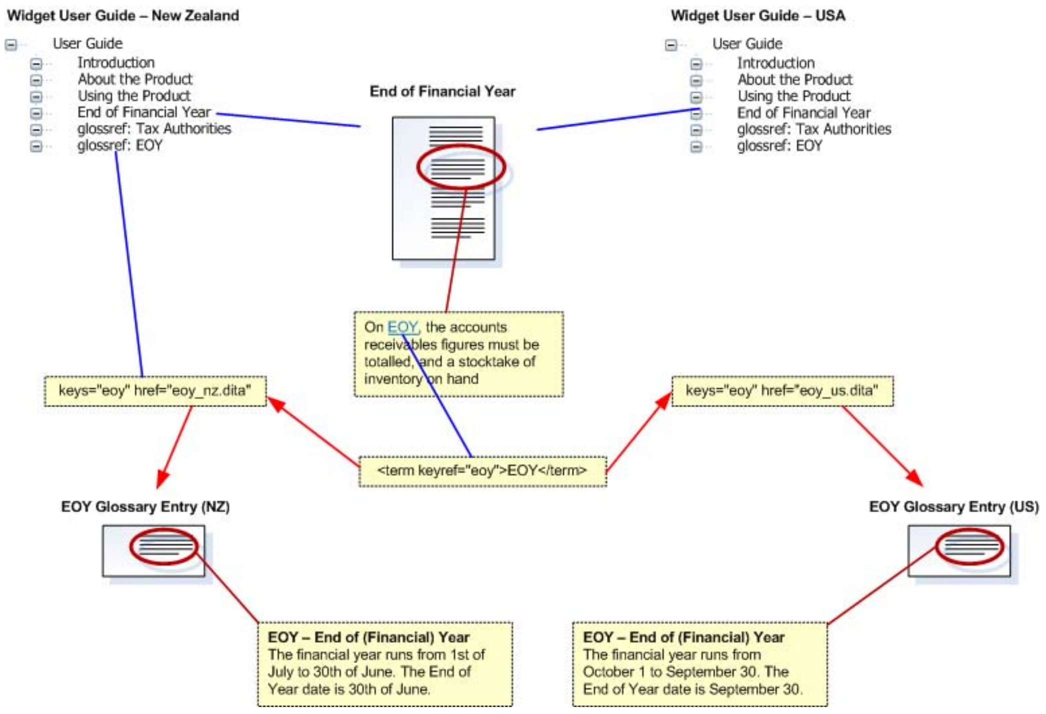
Figure 36: Hypothetical example of how a DITA glossary entry may be rendered in HTML.

Incorporating a glossary in a ditamap

Glossary topics (both `glossentry` and `glossgroup`) can be incorporated into a ditamap using a standard `topicref` element or through a new `glossref` element.

The `glossref` element has a required `keys` attribute used to help link terms in content topic (*inline terms*) to their glossary definitions. Rather than linking an inline term to a particular glossary topic file name, you can link the term to the keys attribute of the relevant `glossref` entry. This *indirect addressing* mechanism allows greater flexibility for single-sourcing, where the glossary definition might be different for different collections of topics.

The diagram below shows the glossref being used for indirect linking of terms.



Widget User Guide – New Zealand
User Guide
Introduction
About the Product
Using the Product
End of Financial Year
glossref: Tax Authorities
glossref: EOY
End of Financial Year
Widget User Guide – USA
User Guide
Introduction
About the Product
Using the Product
End of Financial Year
glossref: Tax Authorities
glossref: EOY
On EOY, the accounts receivables figures must be totalled, and a stocktake of inventory on hand
keys="eoy" href="eoy_nz.dita"
keys="eoy" href="eoy_us.dita"
<term keyref="eoy">EOY</term>
EOY Glossary Entry (NZ)
EOY Glossary Entry (US)
EOY – End of (Financial) Year
The financial year runs from 1st of July to 30th of June. The End of Year date is 30th of June.
EOY – End of (Financial) Year
The financial year runs from October 1 to September 30. The End of Year date is September 30.

Figure 37: Indirect Referencing of Glossary Topics using keys

An important difference between `glossref` and `topicref` is that `glossref` won't result in the glossary topic being inserted in the output at the position the reference appears in the ditamap.

Creating a glossary section

A glossary section in an output document is defined by referencing, through normal `topicref` elements, all the glossary (`glossentry` and `glossgroup`) topics in a normal ditamap. The topics must be manually sorted into alphabetical order unless the authoring tool offers an automated topic sorting feature.

```
<map>
.
 <topicref href="abs.dita" />
 <topicref href="awd.dita" />
.
 <topicref href="wrx.dita" />
.
</map>
```

Linking terms in content topics

The linking of a term occurring in-line in a normal content topic to its relevant `glossentry` definition is a matter of associating the term (semantically identified in a `term` element) with the hypertext reference (`href`) or the key reference (`keyref`) of the glossary topic.

For example, the mark-up `<term keyref="eoy">EOY</term>` establishes the association between the term "EOY" and the glossary topic with a `keys` attribute of `eoy`. The related map entry may be `<glossref keys="eoy" href="eoy_nz.dita"/>`.

In addition to using the `term` element as the basis for linking, a new `abbreviated-form` element provides an alternative method of using glossary entries by substituting the element for the glossSurfaceForm or glossAcronym variants from a glossary entry on rendering. It is expected that a processor will substitute the first occurrence of an `abbreviated-form` element with the `glossSurfaceForm` variant in the glossary entry and subsequent occurrences with the `glossAcronym` variant of the glossary entry. The `abbreviated-form` element acts like a content reference but uses the `key` rather than the `href` as the reference.

For example, mark-up of:

```
<p>An <abbreviated-form keyref="abs"/> helps a driver to stop. For
this reason many find an <abbreviated-form keyref="abs"/> useful.</p>
```

might be rendered in HTML as:

```
<p>An Anti-lock Braking System (ABS) helps a driver to stop. For
this reason many find an ABS useful.</p>
```

where the glossref in the map was `<glossref keys="abs" href="abs.dita"/>`, and the `abs.dita` glossary entry topic was:

```
<glossentry>
  <glossterm>Anti-lock Braking System</glossterm>
  <glossBody>
    <glossSurfaceForm>Anti-lock Braking System
(ABS)</glossSurfaceForm>
    <glossAlt>
      <glossAcronym>ABS</glossAcronym>
    </glossAlt>
  </glossBody>
</glossentry>
```

About `keys` and `keyref` attributes, and indirect linking

Whenever a topic has a reference to other content, it makes the topic less reusable because of the dependency on the target being still available and still relevant. The `keys` and `keyref` attributes provide a simple redirection scheme that leverages existing attributes and map architectures to provide support for redirectable conrefs, topicrefs, xrefs, links, terms, and other reference elements and attributes. They also provide a simplified architecture for managing variable or volatile content (such as product names) which need to be easily swapped out when a topic is reused in new contexts.

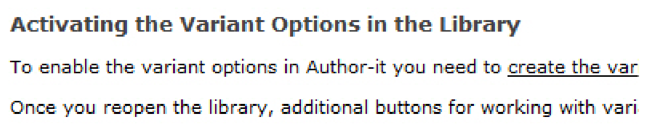
A topic's keys are always defined in the map. A `keys` attribute can contain one or more keys. Keys resolve to the resources given as the `href` value on the topicref element with the matching key.

Summary

The new glossary features in DITA 1.2 offer many opportunities for processing glossaries and terms into rich features in output documents. The enhancements in DITA include new `glossentry` and `glossgroup` information types, a new `glossref` map element, a new `abbreviated-form` alternative to the `term` element, and the addition of a `keys` attribute to the `term` element to allow for indirect linking.

Examples of letter case in titles

Different forms of letter case are used in different publications, according to the particular convention or house style in use. In deciding on a recommended letter case most appropriate for DITA content, the range of alternatives needs to be considered.

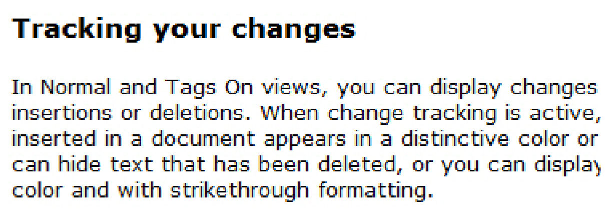


Activating the Variant Options in the Library

To enable the variant options in Author-it you need to create the var

Once you reopen the library, additional buttons for working with vari

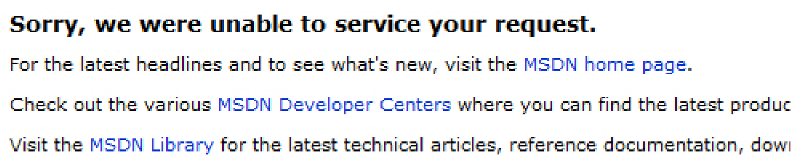
Figure 38: Title Case (no full stop) - Author-it 5.3 Help System



Tracking your changes

In Normal and Tags On views, you can display changes
insertions or deletions. When change tracking is active,
inserted in a document appears in a distinctive color or
can hide text that has been deleted, or you can display
color and with strikethrough formatting.

Figure 39: Sentence case (no full stop) - XMetaL 5.5 Help System

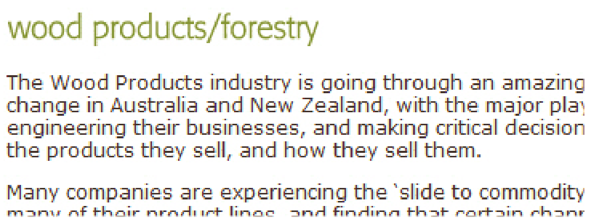

Sorry, we were unable to service your request.

For the latest headlines and to see what's new, visit the MSDN home page.

Check out the various MSDN Developer Centers where you can find the latest produc

Visit the MSDN Library for the latest technical articles, reference documentation, dow

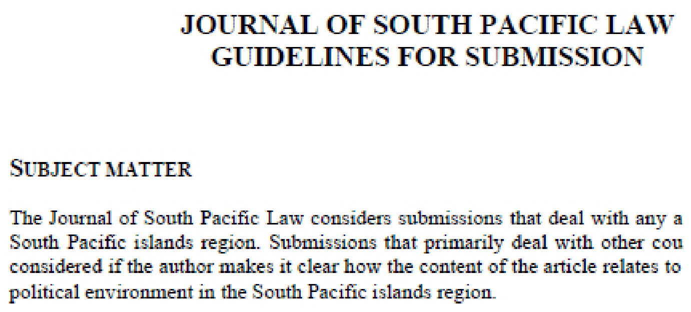
Figure 40: Sentence case (punctuated) - Microsoft MSDN Web Site


wood products/forestry

The Wood Products industry is going through an amazing change in Australia and New Zealand, with the major pla engineering their businesses, and making critical decision the products they sell, and how they sell them.

Many companies are experiencing the 'slide to commodity

Figure 41: All lower case (no full stop) - Touchstone Consulting Web Site


JOURNAL OF SOUTH PACIFIC LAW
GUIDELINES FOR SUBMISSION

SUBJECT MATTER

The Journal of South Pacific Law considers submissions that deal with any a South Pacific islands region. Submissions that primarily deal with other cou considered if the author makes it clear how the content of the article relates to political environment in the South Pacific islands region.

Figure 42: All capitals - Journal of South Pacific Law Document

Separation of content and form in People and Place Style Sheet

This topic categorises the style rules for the People and Place journal into those relating to content (semantics) and those relating to presentational form.

Rule	Category	Comments
Capitalisation of Headings	Form	Provided the title is semantically identified as such, the capitalisation, bolding or italicising of a heading can be controlled in the transformation process. The content,

Rule	Category	Comments
		however, would have to be written in sentence case to provide the greatest range of transformation options.
Treatment of numerals	Content	These rules are not extra-ordinary.
Quote Marks	Form	If semantically identified as quotations, quote marks and treatment of nested quotes can be applied automatically in the transformation process.
No Abstract	Form	Any abstract, if semantically identified, can be omitted from the output by the transformation process.
Financial Years	Content	These rules are not extra-ordinary.
Acronyms and Dashes	Form	If semantically identified, acronyms and abbreviations, and there explanations or definitions, can be treated in many different ways automatically through the transformation process. For example, an abbreviation might be automatically hyper-linked to a matching glossary definition in HTML output from DITA source. Em-dashes, being typographical glyphs, are specific to page-layout outputs. In other words, they are for specific presentational purposes, and don't have a semantic meaning.
Referencing	Form	If semantically identified, and provided there is sufficient metadata, references can be represented in any one of many ways through the automated transformation process. Ultimately, an XML standard for storing references could be integrated with DITA, in the same way that SVG XML for graphics or MathML XML for mathematics can currently be embedded inside DITA source and processed through the same transformation process to an output rendition.

Rule	Category	Comments
Illustrations	Content	There are many opportunities for efficiently handling graphics using XML standards (which can all be integrated with DITA source for processing), including SVG, CML, ODF, and MathML.

Semantic elements for user documentation

The DITA standard groups elements used within the body of topics into five domains: typographical, programming, software, user interface, and utilities. These domains were designed for documentation sets for particular technologies; for other purposes, it is difficult to identify which element is appropriate for a specific scenario. This topic recommends base elements to use, or to base specialisations upon.

Item	Base DITA Element to Use	Rationale
Button in Web application UI	<uicontrol> (User Interface domain)	Buttons, dropdowns, option buttons, checkboxes and menu items are all clearly user interface controls.
Explanatory (UA) text next to a field, as in “Enter your full name” adjacent to the “Name” field	<uicontrol> (User Interface domain)	The distinction between a field label and user assistance text is moot.
Database table	<synph> (Programming domain)	(Example: The *Tax File Number* is stored in the TFN field within the Customer table.)
Database field	<synph> (Programming domain)	(Example: The *Tax File Number* is stored in the TFN field within the Customer table.)

Item	Base DITA Element to Use	Rationale
Column in table	<synph> (Programming domain)	In this case, the table may be displaying the rows of a database table, with the column representing one of the database fields.
Template, skin or theme	<option> (Programming domain)	(Example: Use the External skin for mailout brochures to customers.)
Link in a Web application	<uicontrol> (User Interface domain)	A link in a Web application is analogous to a button in a standalone executable application, and should be treated semantically in the same way. (Example: Login link for Client Access.)
A Web address (URL) or other URI	<filepath> (Software domain)	The address of an Internet resource is logically analogous to the address of a file, which is a file path.
Name of an XML element or attribute	<synph> (Programming domain)	The `synph` element is the closest fit to the semantics of an XML element name or attribute name, such names being part of the syntax of the XML language.
An XML element, described literally	<codeph> (Programming domain)	If the element is being described using tag delimiters (<>), then it should be treated as a snippet of code.
The value of an XML element or attribute	<userinput> (Software domain) or <option> (Programming domain)	What the user enters in an element is usually either freeform text, in which case `userinput` is the more appropriate mark-up, or a choice from a set of appropriate values. If the value is normally chosen from a list, `option` is the more appropriate mark-up.

Item	**Base DITA Element to Use**	**Rationale**
The value of an XML element, where the valid options are a set of pre-defined values (*controlled values*)	<option> (Programming domain)	When the user is selecting a value for an element or a field from a list of some sort, `option` is the more appropriate mark-up. Although intended for the programming domain, `option` is a closest fit when documenting XML and SGML mark-up.
Setting on a dropdown list	<option> (Programming domain)	Although part of the programming domain, the UI domain has no semantically superior alternative. The best alternative in the software domain is `userinput`, but this is better suited to denoting keyboard entry. If an option can be selected from a dropdown or typed in to the interface, use `userinput`.
Placeholder for a name	<varname> (Software domain)	There is no appropriate semantic element, so a convention of surrounding the text with square brackets should be used. The `ph` element can't be used, because it is not permitted within some elements (eg, `filepath`) where a placeholder might be used.
File or directory names, including file prefixes and file extensions	<filepath> (Software domain)	The filepath element may be used in conjunction with the placeholder convention; for example, `[document_name].pdf`.
File or directory name as part of user input	no special mark-up	The fact that the user input happens to be a file path is not semantically relevant. It is just user input. (The `filepath` element is not permitted within `userinput`.)

Item	**Base DITA Element to Use**	**Rationale**
A selection previously made or set by user input	<option> (Programming domain)	The text may describe the location of a code element with a value of important. Although from the programming domain, this element is the closest semantic match to the requirement.
The name of a key on a keyboard	<uicontrol> (User Interface domain)	The keyboard is technically an input peripheral of a computer. Keys therefore form part of the computer application's user interface, and are best described by `uicontrol`.
One or more keystrokes that the user is required to enter	<userinput> (Software domain)	When the user needs to type something into a field or control, the semantic wrapper is userinput. It should make no difference whether the key is an alpha-numeric key or a special purpose key such as **Tab**.
Quoting a piece of text to illustrate the use of punctuation	<q> (proto)	This information is semantically a literal quotation.
Photo credits	<desc><cite> (proto)	Photo credits and similar citations should be included as part of the `figure` element's `desc` element, normally contained in a `cite` element.
Citation, author name or other credit for a long quotation	@reftitle	The `lq` element has a `reftitle` attribute, which should be used to record the source of a long quotation.
Output or responses from a computer system, a device, or	<systemoutput> (Software domain)	Should also be used to describe a characteristic in generated output from a

Item	Base DITA Element to Use	Rationale
other equipment, other than system messages.		publishing process, such as a numbering sequence.
The mode that a computer application might be in, such as Edit or View modes of a Wiki.	<option> (Programming domain) or <state> (Specialisation domain) or	Although `state` may appear to be the most semantically appropriate, this element cannot contain content. It is mainly intended for specialisation, and is appropriate for things that may be described by name/value pairs, including "mode=edit", or "alarm=active". The corresponding coding should normally be `<state name="mode" value="edit"/>`; `<state>edit</state>` mode is invalid. It is typically rendered as state: mode=edit. If `state` is not appropriate, `option` can be used, even though it is not an exact semantic match.
The name of a script, program or command to be run or executed.	<cmdname> (Software domain)	This can be used for command names, SQL jobs, macros, program names, script names, etc, in the context of an end-user's application of that command. When referring to a batch file names (eg, `winant.bat`), use `cmdname` if referring to the command entered in the command line. If referring to the batch file itself, the `filepath` element should be used.
The name of a function, method, sub-routine, API call, or procedure.	<apiname> (programming domain)	The `apiname` element is similar to the `cmdname` element, but should only be used for function, method, sub-routine and procedure names in the context of documentation for programmers and

Item	Base DITA Element to Use	Rationale
		developers. For example, when referring to a JavaScript function that a Web developer may use, `apiname` is appropriate.
A placeholder in syntax or user input to indicate substitution with a parameter or variable	<parmname> (Programming domain) or <varname> (Software domain)	Parameter name (`parmname`) is appropriate when documenting programming languages or APIs, but `varname` is more appropriate for software applications and other scenarios. There is a similar element to `varname` named `var`, but this should only be used in syntax diagrams.
A list of parameters, arguments, variables, or command switches	<parml> (Programming domain)	Parameter list (`parml`) has a list structure comprising `plentry` (rows), `pt` (term), and `pd` (definition).
A status message automatically displayed by a system in the user interface	<msgph> or <msgblock> (Software domain)	An example of a status message might be the `Send/Receive complete` message displayed in the **status bar** in Outlook.
The response from a system to user input of some kind	<systemoutput> (Software domain)	When the system displays a direct response to input from the user, such as a total being displayed, a light changing colour, a receipt being printed, or a door unlocking, the `systemoutput` element should be used to describe the response.
Product names (and company names), if not a trademark	<keyword> (proto)	The `keyword` element is used to identify any sort of keyword or token, product name, or a lookup key for a message. The use of this element is complicated by the fact that it will

Item	Base DITA Element to Use	Rationale
		automatically be included in the topic's metadata.
Program name or application name	<keyword> (proto)	A program name, such as Internet Explorer, is just another type of product name. If you are referring to the name in the title bar of an application, in order to identify the application's window, use <wintitle>.

Conclusion

What constitutes best practice in DITA?

The research in my PhD by Artefact and Exegesis *action research* project addressed the question of **what constitutes best practice in DITA authoring** by creating a book that documents that best practice. The book, *The DITA Style Guide: Best Practices for Authors* (the Artefact) is an original synthesis that finds new knowledge through the analysis and development of existing disparate material together with my own practical experience.

DITA is an innovation with the potential to change habits of composition, expression and presentation for a broad range of writers and in so doing enhance productivity across a number of industries and occupations. Establishing best practice in this emerging field will enable users to maximise the productivity dividend.

The audience for this research is both academic and commercial. Teachers and researchers in technical communication will benefit from the publication of *The DITA Style Guide* as a reference source for further work in the field of structured authoring and *semantic authoring.* Implementers, project managers and technical communication practitioners will find *The DITA Style Guide* assists the documentation and vocational learning processes through suggested guidelines and standards.

My motivation to develop *The DITA Style Guide* is rooted in my career-long interest in the use of technology in technical documentation. The production of the Artefact and Exegesis, and the later donation of the Artefact to the DITA community as an open source resource, adds to the body of knowledge on the application of DITA principles in technical communication, with the aim of contributing to the success of the DITA methodology.

References

Bibliography

- Abel, S. (2006, July 30). 10 DITA Lessons Learned From Tech Writers in the Trenches. The Content Wrangler. Retrieved January 25, 2011, from http://thecontentwrangler.com/2006/07/30/10_dita_lessons_learned/.
- About OER Commons. (n.d.). Open Educational Resources. Organisational, . Retrieved December 2, 2009, from http://www.oercommons.org/about#about-open-educational-resources
- About SourceForge. (n.d.). SourceForge. Commercial, . Retrieved November 30, 2009, from http://sourceforge.net/about
- About W3C. (n.d.). World Wide Web Consortium (W3C). Organisational. Retrieved November 30, 2009, from http://www.w3.org/Consortium/
- Agnosticism. (n.d.). In Wikipedia. Retrieved May 15, 2010, from http://en.wikipedia.org/wiki/Agnosticism
- Ament, K. (2003). Single sourcing: Building modular documentation (1st ed.). New York: William Andrew.
- André, J. (1998). Petite histoire des signes de correction typographique. Cahiers GUTenberg, (31), 45-59.
- André, J., & Richy, H. (1999). Paper-Less Editing and Proofreading of Electronic Documents. EuroTeX'99 Proceedings. Retrieved July 8, 2010, from http://www.irisa.fr/imadoc/articles/1999/heidelberg.pdf.
- Apache Derby: Writing guidelines. (n.d.). Apache Derby. Retrieved October 10, 2009, from http://db.apache.org/derby/manuals/guidelines.html.
- Arnold, J. (2005). The PhD In Writing Accompanied By An Exegesis. Journal of University Teaching and Learning Practice, 2(1), 36-50.

- Barker, T. T. (2003). Writing Software Documentation: A Task-Oriented Approach. Allyn and Bacon Series in Technical Communication (2nd ed.). New York: Longman.
- Berners-Lee, T. (2002, April 17). Using Style Sheets - Style Guide. World Wide Web Consortium (W3C). Retrieved October 9, 2009, from http://www.w3.org/Provider/Style/StyleSheets.html.
- Bleiel, N. (2008). tcworld 2009 Conference Proceedings - Using Journalistic Principles to Improve User Assistance. tekom.
- *Capitalization - Heading and publication titles*. (n.d.). Wikipedia. Retrieved January 9, 2009, from http://en.wikipedia.org/?title=Title_case#Headings_and_publication_titles
- Carey, M., & Rouiller, S. (2008, November 12). DITA Short Description Guidelines. Presented at the Silicon Valley DITA Interest Group, Presentation, San Francisco. Retrieved April 30, 2008, from http://svdig.ditamap.com/Shortdesc.ppt
- CNN. (1999, February 25). Internet Time: Will the 'beat' go on? CNN News. Retrieved November 30, 2009, from http://www.cnn.com/TECH/computing/9902/26/t_t/internet.time/
- DADirect. (n.d.). YouTube - A&R *Espresso Book Machine*. Retrieved January 9, 2009, from http://au.youtube.com/watch?v=MXP0E7i0bfU
- Damrau, J. (2005). Developing a Corporate Style Guide. STC Proceedings. Retrieved August 17, 2009, from http://www.stc.org/ConfProceed/2005/PDFs/0073.pdf
- Day, D., Priestley, M., & Schell, D. (2005). Introduction to the Darwin Information Typing Architecture. IBM developerWorks. Retrieved February 9, 2009, from http://www.ibm.com/developerworks/xml/library/x-dita1/
- DeLoach, S. (1998). Microsoft Manual of Style (Review). Technical Communication: Journal of the Society for Technical Communication, 55(3), 285-306.
- Doyle, B. (2007, April 3). HATs Off to XML Help. EContent - Digital Content Strategies and Resources . Commercial, . Retrieved December 26, 2009, from http://www.econtentmag.com/Articles/ArticleReader.aspx?ArticleID=35804
- Dubinko, M. (2009, November 22). How Xanadu Works: technical overview. Micahpedia. Retrieved November 26, 2009, from http://dubinko.info/blog/2009/11/22/how-xanadu-works/
- Eberlein, K. J. (2007 12). Art of the short description | DITA XML.org. DITA XML.org. Retrieved March 7, 2008, from http://dita.xml.org/art-short-description

- Eberlein, K. J. (2009, August 21). Order for lang ref files (OASIS DITA TC correspondence). Retrieved from http://markmail.org/message/r67bf2jystkp34ir.
- *E Ink Corporation* Web Site. (n.d.). . Retrieved January 9, 2009, from http://www.eink.com/
- Elgan, M. (n.d.). *Why Amazon's Kindle is revolutionary*. Computerworld. Retrieved January 9, 2009, from http://www.computerworld.com.au/index.php/id;748883531;pp;1;fp;4194304;fpid;1
- Ellison, M. (2004, February). Seven Plus or Minus Two - Put to the Test.pdf (application/pdf Object). Institute for Scientific and Technical Communicators. Retrieved November 13, 2009, from http://www.istc.org.uk/Events/Conference/Papers/ Seven%20Plus%20or%20Minus%20Two%20-%20Put%20to%20the%20Testpdf.
- Ellison, M. (2007, April). The Magical Number: Seven, Plus or Minus Two. SoundOff, 7(2). Retrieved January 25, 2011, from http://stage.stc-psc.org/newsletter/fall2006.html#The_Magical_Number.
- Farkas, D. (1999). The Logical and Rhetorical Construction of Procedural Discourse. Technical Communication: Journal of the Society for Technical Communication, 46(1), 42-54.
- Flann, E., & Hill, B. (2004). The Australian editing handbook (2nd ed.). Milton, Queensland: John Wiley and Sons.
- Freelance Editorial Association. (n.d.). *Types of Editing*. Retrieved January 9, 2009, from http://www.edsguild.org/types.htm
- Gardiner, D. (2010, May). Discovering XML for editing. The Canberra Editor, 19(4), 6-10.
- Hackos, J. (1994). Managing your documentation projects. Wiley Technical Communication Library (1st ed.). New York: John Wiley and Sons.
- Hackos, J. (2006). Do We Really Need All that Glue? DITA XML.org. Retrieved March 8, 2008, from http://dita.xml.org/do-we-really-need-all-glue
- Hackos, J., & Hedlund, T. (2001). Making a business case for single sourcing. Denver, Colorado: Center for Information-Development Management. Retrieved May 15, 2010, from http://www.comtech-serv.com/pdfs/Business_Case_for_Single_Sourcing.pdf
- Hackos, J., & Stevens, D. (1997). Standards for Online Communication. New York: John Wiley and Sons.

- Hale, C. (1996). Wired Style - Principles of English Usage in the Digital Age (1st ed.). San Francisco: Publishers Group West.
- Hart, G. (2000, March). The style guide is "dead": long live the dynamic style guide! Geoff-Hart.com. Retrieved August 25, 2009, from http://www.geoff-hart.com/articles/2000/dynamicstyle.htm
- Hollis Weber, J. (2007, October). Developing a Departmental Style Guide. Technical Editors' Eyrie. Retrieved August 18, 2009, from http://www.jeanweber.com/newsite/?page_id=30
- Horn, R. E. (1998). Structured Authoring as a Paradigm. In Instructional Development: State of the Art. Educational Technology Publications. Retrieved March 10, 2010, from http://www.stanford.edu/~rhorn/a/topic/stwrtng_infomap/artclStWrAsParadigm.html
- Horn, R. E. (1999). Two Approaches to Modularity: Comparing the STOP Approach with Structured Writing. Journal of Computer Documentation.
- Houlihan, D. (2008). The technical communicator's transformation: Publishing on time and on quality. Boston: Aberdeen Group.
- Houlihan, D. (2010). Benchmarking content reuse in technical communication. Boston: Aberdeen Group.
- Howard, J. R. (1969). The Flowering of the Hippie Movement. The Annals of the American Academy of Political and Social Science, 382(1), 43-55.
- Hudson, N. (1993). Modern Australian Usage (p. 294). Melbourne: Oxford University Press.
- Hughes, M. (2009, September 18). Architecting UA Topics for Reuse. Presented at the UA Europe Conference, Conference, Cardiff, Wales. Retrieved September 30, 2009, from http://www.uaconference.eu/sessions.html#reuse
- Hyttinen, L. (2008, May). Creating a Style Guide - A Case Study of Sandvik Mining and Construction Hard Rock Mining Project. Pro Gradu Thesis, University of Tampere. Retrieved from http://tutkielmat.uta.fi/pdf/gradu02978.pdf
- Illich, I. (1974). Deschooling Society. London: Calder and Boyars.
- IBM Style. (2009). IBM UT Style and Word Usage Council.

- International Standard ISO/IEC 26514: Systems and software engineering - Requirements for designers and developers of user documentation. (2008, June 15). International Standards Organisation.
- International Standard ISO/IEC 5776: 1983: Graphic technology - Symbols for text correction. (1983). International Organization for Standardization.
- Johnson, T. (2009, December 27). If No One Reads the Manual, That's Okay. I'd Rather Be Writing. Retrieved January 14, 2010, from http://www.idratherbewriting.com/2009/12/27/if-no-one-reads-the-manual-thats-okay/
- Jones, D. (1997). Technical Writing Style. Allyn and Bacon Series in Technical Communication. New York: Longman.
- Keen, A., & Bell, E. (2007, August 10). Andrew Keen v Emily Bell | Comment is free | guardian.co.uk. The Guardian. Retrieved August 10, 2009, from http://www.guardian.co.uk/commentisfree/2007/aug/10/andrewkeenvemilybell
- Krechmer, K. (1999). Technical Communications Standards: New Directions in Innovation. Presented at the First IEEE Conference on Standardization and Innovation in Information Technology, Aachen, Germany. Retrieved February 1, 2010, from http://www.csrstds.com/siit.html
- Kulik, B., & Nigloschy, D. (2003). *XML Centric Workflows: New Opportunities for Publishers*. Retrieved January 9, 2009, from http://www.idealliance.org/papers/dx_xml03/papers/03-06-03/03-06-03.pdf
- Leadbeater, C. (2008). We-Think (1st ed.). London: Profile Books.
- Lockwood, K. (2005). Style: The essential guide for journalists and professional writers (3rd ed.). Melbourne: News Custom Publishing.
- Lynch, P. L., & Horton, S. (n.d.). Rhetoric and Web Design. Web Style Guide Online 3rd Edition. Retrieved January 25, 2011, from http://webstyleguide.com/wsg3/9-editorial-style/4-rhetoric-and-web-design.html.
- Lynch, P. L., & Horton, S. (2009). Web Style Guide: Basic Design Principles for Creating Web Sites, Web Style Guide (3rd ed.). New Haven, USA: Yale University Press. Retrieved from http://webstyleguide.com/index.html

- Manning, S. (n.d.). *XML: To Train, or not to train*. The Rockley Report. Retrieved December 17, 2008, from http://www.rockley.com/TheRockleyReport/V1I4/People%20Processes%20and%20Change.htm
- McDaniel, E. (2005). Consider the source: Structured authoring for XML-based documentation. Presented at the UNC College and University System Exchange (CAUSE) Conference, Raleigh, North Carolina: University of North Carolina. Retrieved February 23, 2010, from http://www.unc.edu/cause05/presentations/mcdaniel/mcdaniel.pdf
- McGovern, G. (2009, November 2). Why web links are calls to action. New Thinking. Retrieved February 23, 2010, from http://www.gerrymcgovern.com/nt/2009/nt-2009-11-02-Web-links-action.htm
- Microsoft® Manual of Style for Technical Publications, Third Edition, 2004, ISBN 0-7356-1746-5
- Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review, 63(2), 81-97.
- Muegge, U. (n.d.). *Controlled language: the next big thing in translation?* Translation Directory. Retrieved March 10, 2008, from http://www.translationdirectory.com/articles/article1359.php
- MythTV: Manual of Style. (n.d.). MythTV. Retrieved January 27, 2011, from http://www.mythtv.org/wiki/MythTV:Manual_of_Style.
- Netcraft Web Server Survey. (2009, February). Netcraft. Retrieved November 30, 2009, from http://news.netcraft.com/archives/2009/02/18/february_2009_web_server_survey.html
- Nicholas Negroponte Interview on Lateline [videorecording]. (1996, March 19). Lateline (Australian Broadcasting Corporation). Sydney.
- OASIS. (2007, August 1). Transitional text. DITA Version 1.1 Architectural Specification. Organisational, . Retrieved December 27, 2009, from http://docs.oasis-open.org/dita/v1.1/OS/archspec/transitionaltext.html
- OASIS DITA Adoption TC Web site. (n.d.). OASIS Open. Non-commercial, . Retrieved December 1, 2010, from http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dita-adoption.
- Ogden, C. (1930). Basic English: A General Introduction with Rules and Grammar. London: Paul Treber and Company.

- O'Hara, F. M. (2001). A Brief History of Technical Communication. In STC Proceedings 2001. Presented at the 48th International STC Conference, Chicago: Society for Technical Communication. Retrieved February 23, 2010, from http://www.stc.org/confproceed/2001/PDFs/STC48-000052.pdf
- O'Hara, K., & Sellen, A. (1997). A Comparison of Reading Paper and On-Line Documents (pp. 335-342). Presented at the Proceedings of CHI '97, Human Factors in Computing Systems, Atlanta. Retrieved July 13, 2010, from http://scholar.google.com.ezproxy.lib.swin.edu.au/scholar?q=kenton+o%27hara&hl=en&lr=&btnG=Search.
- O'Keefe, S. (2006, April 12). Wednesday at WritersUA: Structured authoring: taking the plunge. Scriptorium. Retrieved May 15, 2010, from http://www.scriptorium.com/blog/2006/04/wednesday-at-writersua-structured-authoring-taking-the-plunge.html
- O'Neill, J. (2002). A global style guide: Working together around the world. STC 2002 Conference Proceedings. Retrieved August 26, 2009, from http://www.stc.org/confproceed/2002/PDFs/STC49-00024.pdf
- Peters, P. (2004). The Cambridge Guide to English Usage (1st ed.). Cambridge: Cambridge University Press.
- Parkes, M. (1993). Pause and Effect: An Introduction to the History of Punctuation in the West (1st ed., p. 327). Los Angeles: University of California Press.
- Petelin, R., & Durham, M. (1992). The professional writing guide (1st ed.). Warriewood, NSW: Business and Professional Publishing.
- Perlin, N. (2007, January). Writing becomes industrial. Intercom, 54, 17-19.
- Price, J., & Korman, H. (1993). How to Communicate Technical Information: A Handbook of Software and Hardware Documentation. Boston: Addison-Wesley.
- Priestley, M., Anderson, R. E., & Hackos, J. (Eds.). (2007). DITA Language Specification v1.1. OASIS.
- Publication Manual of the American Psychological Association, 5th Edition, Washington DC, 2001, APA
- Purchase, S. (1998). The Little Book of Style. Canberra: AusInfo (Department of Finance and Administration).
- Radio National style guide. (2004, March 1). Australian Broadcasting Corporation. Commercial, . Retrieved January 27, 2011, from http://style.radionational.net.au/glossary.

- Raya, R. (2004). *XML in Localisation: Use XLIFF to translate documents*. IBM developerWorks. Retrieved January 9, 2009, from http://www.ibm.com/developerworks/xml/library/x-localis2/
- Reason, P., & Bradbury, H. (Eds.) (2001). Handbook of Action Research: Participative Inquiry and Practice. London: Sage Publications.
- Reshef, S. (2009). UoPeople Student Handbook - Introduction. University of the People. Retrieved December 2, 2009, from http://www.uopeople.org/LinkClick.aspx?fileticket=svtG9FhsImk%3d&tabid=266
- Rijken, D. (2007). Hippies 2.0 in Museum 2.0. In The User is the Content - 10 Scenarios for the Future (pp. 18-20). Antwerp: C.H.I.P.S. vzw. Retrieved November 30, 2009, from http://www.lulu.com/items/volume_64/1148000/1148772/8/print/MeetingBook3.pdf
- Robidoux, C. (2008). Rhetorically structured content: Developing a collaborative single-sourcing curriculum. Technical Communication Quarterly, 17(1), 110-135.
- Rockley, A. (2001). The impact of single sourcing and technology. Technical Communication: Journal of the Society for Technical Communication, 48(2), 189-199.
- Rockley, A., Manning, S., & Cooper, C. (2009). DITA 101 - Fundamentals of DITA for Authors and Managers. Lulu.com.
- Sapienza, F. (2004). Usability, structured content, and single sourcing with XML. Technical Communication: Journal of the Society for Technical Communication, 51(3), 399-408.
- *Serial comma*. (n.d.). Wikipedia. Retrieved January 9, 2009, from http://en.wikipedia.org/wiki/Serial_comma
- Self, T. (n.d.). The Mysterious Acronym Tag. HyperWrite. Commercial, . Retrieved July 6, 2010, from http://www.hyperwrite.com/Articles/acronym_tag.aspx.
- Self, T. (2008). Structured Writing in XML Participant Guide (1st ed.). Sydney: TACTICS Consulting. (Commercial training workbook)
- Self, T. (2008). *What if Readers Can't Read?* HyperWrite. Retrieved January 9, 2009, from http://www.hyperwrite.com/Articles/showarticle.aspx?id=84
- Self, T. (2009). DITA and the challenges of single-source article publishing. In *Communication, Creativity and Global Citizenship: Refereed Proceedings of the Australian and New Zealand Communications Association Annual Conference*. Presented at the

ANZCA Annual Conference, Brisbane. Retrieved July 2, 2010, from http://www.cpe.qut.edu.au/conferences/2009/anzca/proceedings/Self_ANZCA09.pdf.

- Self, T. (2009, March). Improved Glossary and Terminology Handling Features in DITA 1.2. HyperWrite. Commercial. Retrieved November 25, 2009, from http://www.hyperwrite.com/Articles/showarticle.aspx?id=86
- Self, T. (2010, October). Writing to STOP. CIDM Information Management News, 10(10). Retrieved February 10, 2011, from http://www.infomanagementcenter.com/enewsletter/2010/201010/second.htm.
- Self, T. (2010, December). Reducing Context in Modular Documents. Best Practices Newsletter (CIDM), 12(6), 133-140.
- Self, T. (2010). Context-agnostic writing in modular documents. In *Media, Democracy and Change: Refereed Proceedings of the Australian and New Zealand Communication Association Conference 2010*. Presented at the ANZCA Conference 2010, Canberra. Retrieved February 10, 2011, from http://www.anzca.net/conferences/conference-papers/94-anzca10proceedings.html.
- Self, T. (2011, March). Can you explain that again? DITA for beginners. tcworld, 19-21.
- Sieghart, M. A. (2007, May 25). Hey man, we're all kind of hippies now. Far out. Times Online. Newspaper, . Retrieved November 30, 2009, from http://www.timesonline.co.uk/tol/comment/columnists/article1837763.ece
- Simpson, P. (1935). Proof-Reading in the Sixteenth, Seventeenth and Eighteenth Centuries (1st ed.). London: Oxford University Press.
- Snooks and Co. (2002). Style manual: for authors, editors and printers (6th ed.). Brisbane: Wiley Australia.
- Structured Authoring at Swinburne University of Technology. (n.d.). Swinburne University of Technology, Melbourne, Australia. Education, . Retrieved February 22, 2011, from http://courses.swinburne.edu.au/subjects/Structured-Authoring-HATC424/local.
- Style Sheet for People and Place. (n.d.). Monash University. Retrieved September 19, 2009, from http://elecpress.monash.edu.au/pnp/author/stylesheet.pdf.
- Sun Technical Publications. (2003). Read Me First! A Style Guide for the Computer Industry (2nd ed.). Mountain View, California: Prentice Hall PTR.
- Swain, H. (2009, November 30). Online university of hope. The Age, 14.

- The Associated Press. (2009). The Associated Press Stylebook 2009 Fully Revised and Updated (43rd ed.). New York: Basic Books. Retrieved from http://www.apstylebook.com/
- The Chicago Manual of Style. (1982). (13th ed.). Chicago: University of Chicago Press.
- The Chicago Manual of Style Citation Quick Guide. (n.d.). The Chicago Manual of Style Online. Retrieved November 11, 2009, from http://www.chicagomanualofstyle.org/tools_citationguide.html
- The Mozilla Manifesto. (n.d.). Mozilla Organisation. Organisational, . Retrieved November 30, 2009, from http://www.mozilla.org/about/manifesto.en.html
- The Newspaper Style Manual... (1999, December 4). Lingua Franca. Australian Broadcasting Corporation. Retrieved January 26, 2011, from http://www.abc.net.au/rn/arts/ling/stories/s70685.htm.
- Thesis Statements. (n.d.). The Writing Center, University of North Carolina. Student handouts, . Retrieved November 8, 2009, from http://www.unc.edu/depts/wcweb/handouts/thesis.html
- Tips on Writing Your Thesis Statement. (n.d.). Jackson School of International Studies Writing Center. Academic, . Retrieved November 8, 2009, from http://depts.washington.edu/pswrite/thesisstmt.html
- Tracey, J. R., Rugh, D. E., & Starkey, W. S. (1965, January). Sequential Thematic Organization of Publications (STOP): How to Achieve Coherence in Proposals and Reports. University of Washington. Retrieved May 9, 2009, from http://faculty.washington.edu/farkas/TC510/TraceySTOPReport.pdf
- Van Buren, R., & Buehler, M. F. (1980). *The Levels of Edit*. Pasadena, California: Jet Propulsion Laboratory, 1980 Retrieved December 13, 2008.
- Vazquez, J. (2009). Practical DITA (1st ed.). Durham, NC: Write Spirit.
- W3C Quality Assurance Web Site. (n.d.). World Wide Web Consortium (W3C). Retrieved October 9, 2009, from http://www.w3.org/QA/.
- Walker, J., & Taylor, T. (n.d.). The Columbia Guide to Online Style (2nd ed.). Columbia University Press.
- Wallace, D., & Hughes, J. (1995). Style Book - A Guide for New Zealand Writers and Editors (5th ed.). Wellington: GP Publications.
- Weber, Steven (2005). The Success of Open Source. Harvard: Harvard University Press.

- Web Style Guide (Version 4). (n.d.). Monash University. Academic, . Retrieved May 1, 2009, from http://www.monash.edu.au/staff/web/.
- Weiss, E. H. (n.d.). Improving International Technical Communication by Containing Prose Paragraphs.pdf (application/pdf Object). Retrieved November 11, 2009, from http://www.edmondweiss.com/PDF/Improving%20International%20%20Technical%20Communication%20by%20Contaning%20Prose%20Paragraphs.pdf
- Weiss, E. H. (1999). Bits, Atoms, and the Technical Writer: The Rhetoric of STOP. Journal of Computer Documentation.
- Weiss, E. H. (2005). The Elements of International English Style : A Guide to Writing Letters, Reports, Technical Documents, and Internet Pages for a Global Audience. New York: M.E. Sharpe.
- Wikipedia: Manual of Style. (n.d.). Wikipedia. Retrieved January 27, 2011, from http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style.
- Wilkinson, R. T. (1987). Proof-reading: VDU and paper text compared for speed, accuracy and fatigue. Behaviour and Information Technology, 6(2), 125-133. doi: 10.1080/01449298708901822.
- Williams, J. D. (2003). The implications of single sourcing for technical communicators. Technical Communication: Journal of the Society for Technical Communication, 50(3), 321-327.
- WritePoint. (2010). DITA challenges survey results (November 2009). Commercial. Retrieved January 18, 2010, from http://www.writepoint.com/site/downloads/WritePoint_DITASurveyChallenges_Results.pdf
- XML Cover Pages: *DITA 1.1 Appproved as an OASIS Standard*. (2007, August 13). The Cover Pages. Retrieved June 8, 2009, from http://xml.coverpages.org/ditaV11-Standard.html
- Yeo, S. (2009, September 2). Order for lang ref files (OASIS DITA TC correspondence). Retrieved from http://lists.oasis-open.org/archives/dita/200909/msg00041.html.
- Zydron, A. (2008, March 4). Best Practice for Using the DITA CONREF Attribute for Translation. OASIS. Retrieved September 29, 2009, from http://www.oasis-open.org/committees/download.php/27580/ConrefBestPracticesWhitePaper%5B1%5D.pdf

Appendix

A

Academic Papers

Topics:

During the course of the project, research observations were analysed and developed into academic conference papers. Some are yet unpublished, but published, peer-reviewed papers are:

- *DITA and the Challenges of Single-Source Publishing* (ANZCA 2009 Conference, Brisbane)
- *Context-Agnostic Writing in Modular Documents* (ANZCA 2010 Conference, Canberra)

Further, *Improved glossary and terminology handling* was published as an official OASIS Whitepaper.

One paper (yet to be published), *Editing Marks for XML Mark-up*, was co-authored with Dr Suku Sinnappan.

DITA and the Challenges of Single Source Publishing

Abstract

DITA is an emerging writing methodology built around XML, and promises major improvements in efficiencies of writing, editing, maintaining, publishing and delivering textual information *(Day et al, 2005)*. Although it is currently predominantly used for IT documentation, DITA can be applied in most fields of publishing, including Web site content management and magazine article publishing, to reduce costs and increase efficiency. DITA is built upon the principle of separation of content and form, which attempts to completely abstract the message of written communication (the *content*) from the reading format in which it is presented to the reader (the *form*). This separation paradigm makes redundant many traditional methods of composing, editing, changing, laying out, printing and publishing. The benefits of a change in approach to writing are numerous. One benefit is the ability to *single-source*, where content source can be maintained in a delivery-agnostic format (DITA), and automatically published to many different types of publications.

This paper explains that the transition to a single-sourcing publishing workflow where content is separated from form will require changes to the way editing is approached in the magazine and journal publishing industry. This paper discusses some of the difficulties in achieving true single-source publishing for article and whitepaper content in future DITA publishing environments, especially when there are different editorial rules for different publications. The paper makes recommendations on changes to the editing practice to lower the barriers to single-sourcing.

The approach taken in this paper is to introduce a case study, explain the concept of the separation of content and form, and to then discuss the challenges that a DITA workflow brings to editing in single-sourced publications, to content re-use, and to single-sourcing with different style requirements.

This paper concludes that for single-sourcing to be feasible for article and whitepaper publishing:

- An open source content style manual needs to be developed to provide agreed trans-organisational mechanical editing standards.
- A greater emphasis must be placed on editing in the developmental stages of an article.
- The presentational format of an article can be largely automated at an organisational level, leading to improved efficiency and consistency within the publishing workflow.

DITA

The *Darwin Information Typing Architecture* (DITA) is an open source, XML-based architecture and methodology for designing, writing, managing, and publishing many kinds of information in print and on the Web. The “Darwin” in the name pays homage to Charles Darwin, the famed naturalist most responsible for the theory of evolution. DITA incorporates principles of specialisation, adaption and inheritance in document structures that are reminiscent of Darwinian theory. “Information Typing” refers to the focus on categorisation of information. “Architecture” indicates that DITA is not just an XML standard; it is an approach, a workflow, a methodology, and a philosophy. Although the first version of the DITA standard was only published in 2005, DITA is already being adopted by technical publication departments in large corporations, including Sun, IBM, Adobe and Nokia (*XML Cover Pages, 2007*).

DITA is lauded as a highly flexible and capable single-source publishing architecture, allowing a range of content re-use options. DITA proponents claim that topics written in DITA by a supplier can be incorporated into a manufacturer's technical publications. On initial examination, it also appears that a magazine article, written as a DITA topic, can be published to a number of Web sites and in a number of different printed publications such as magazines and newsletters.

Single-sourcing is an attractive ideal for publishers, because it can dramatically reduce the cost of writing, and maintaining the currency of, information content. An individual piece of text may be used in a dozen different publishing contexts (on a Web site, in a Help system, in a User Guide, in an Administrator Guide, in a product brochure, etc), but any edits to that text source need occur only once, with the changes flowing to all output publications. The principle in DITA which enables single-sourcing is that of *separation of content and form* through semantic mark-up.

In this paper, the author will test the single-sourcing claim in the context of magazine articles and whitepapers by introducing a case study, then identifying the challenges in achieving single-sourcing, and finally recommending an approach to editing to maximise the chances of effective re-use for future XML-based magazine article publishing.

Case Study

In November 2008, an article was written by the author entitled *What if Readers Can't Read? (Self, 2008)*, intended for publication in technical communication society publications and on industry Web sites. The article, based on a keynote presentation delivered at a Technical Communicators Association of New Zealand (*TCANZ*) event in New Zealand, was accepted for publication in the following online and traditional publications:

- *Southern Communicator* magazine (Quarter 1, 2009)
- *HyperWrite Web Site* (www.hyperwrite.com)
- WritersUA Web Site (www.writersua.com)
- STC Intercom magazine (February, 2009)
- STC France Web site (www.stcfrance.org)

The base DITA *content model* has three information types: concept, task and reference. The article was written as a DITA concept information type, in one topic. Prior to submission for publication, the article underwent a number of edits, including a third-party review.

Intercom magazine, a publication of the Society for Technical Communication (*STC*), requested that the article be submitted in Microsoft Word® format. The Word document was generated from the DITA concept topic using the DITA Open Toolkit (DITA OT). The DITA

OT is an open source collection of software tools used by new DITA adopters. DITA adopters tend to migrate from the DITA OT to commercial tools as their documentation projects grow larger.

The Word document was reviewed by an STC Editor (Liz Pohland), whose changes are summarised in the following table.

Table 5: Analysis of Nature of STC Intercom Edits

DITA Mark-up	**Word® Output**	**STC Edits**	**Nature of Edit**
<title>What if readers can't read?</title>	What if readers can't read?	What If Readers Can't Read?	Capitalisation
<shortdesc>This topic discusses...	This topic discusses...	This article discusses...	Wording
<shortdesc>... workers, and...	workers, and	workers and	Serial comma
resounding <b>Yes</b>	Resounding **Yes**	Resounding "Yes"	Formatting Style
<p>What evidence?...	What evidence?	What evidence do we have?	Writing Style
...with a Web edition...	With a Web edition	With a web edition	Capitalisation
A study by <xref...>Springer...	A study by Springer	A study by Springer (www...)	Externalising links
Source: www.theage.com.au, October 2008	Source: www.theage.com.au, October 2008	(www.theage.com.au)	House style
...mentioned teaching...	mentioned teaching	mentioned using eBooks for teaching	Writing style

DITA Mark-up	Word® Output	STC Edits	Nature of Edit
<tm>Facebook</tm>	Facebook™	Facebook	House style
...wrote:</p><lq>Grammar rules used to be...	...wrote: Grammar rules used to be:	...wrote: "grammar rules used to be..."	Writing style, Punctuation
11%	11%	11 percent	House style
neighbours	neighbours	neighbors	AU to US spelling
texting. 2,000 messages...	texting. 2,000 messages...	texting. Two thousand messages...	House style
to realise that...	to realise that...	to realize that...	AU to US spelling
our audience <term>LCD</term> (lowest common denominator)...	our audience LCD (lowest common denominator)...	our LCD audience...	Writing style
an article titled <cite>Hype Alert...	an article titled Hype Alert...	An article titled "Hype Alert..."	Formatting style
<q>To google</q> is now a verb.	"To google" is now a verb.	To "google" is now a verb.	Writing style
habits. The ABC...	habits. The ABC...	habits. ABC...	AU to US usage, and acronym usage
In one 3 minute and 51 second video...	In one 3 minute and 51 second video...	In one short (under 4 minutes) video...	Writing style

Separation of content and form

The use of semantic mark-up in DITA, where text elements are marked up based on their meaning, allows the content to be completely separated from its rendition and display to the reader. For example, a term is marked up as a `<term>` and a citation as a `<cite>`, and no information about how those elements will be displayed is stored in the content. Stylistic (display) rules are applied when the DITA content is *transformed* into a reading format, such as HTML or paper. In a DITA workflow, documents are created as collections of modular, re-usable topic files, and mechanisms allow not only the format to be separated from the content, but also the context. The same topic may be a section in the context of one publication, but a sub-section in the context of another. The intermingling of content, format and context in a style-based document workflow essentially eliminates the possibility of re-use. Once a paragraph is styled as having a 13 cm left margin, it cannot be used on paper 12 cm wide. A phrase marked up in italic won't render as italic on a reading device that doesn't support italic. But a citation identified as a citation in a DITA topic can be processed to italic by one transformation process, to bold red by a different transformation process, and to synthesised voice by another transformation process.

Most popular styles guides mix content (or *editorial*) style rules (eg, "use active voice for instructions") with presentational (*form*) style instructions (eg, "use italics for product names"). The pie-charts in *Figure 2: Breakdown of Content (Editorial) and Presentational Style Rules in Four Style Guides* (see page 190) (*Snooks and Co, 2002*, *The Chicago Manual of Style, 1982*, *Wallace and Hughes, 1995*, *Lockwood, 2005*) show a simplistic calculation of the proportion of content and presentation style rules in three common publishing industry style guides, and a contrasting newspaper style guide, based on the number of pages devoted to each type of rule.

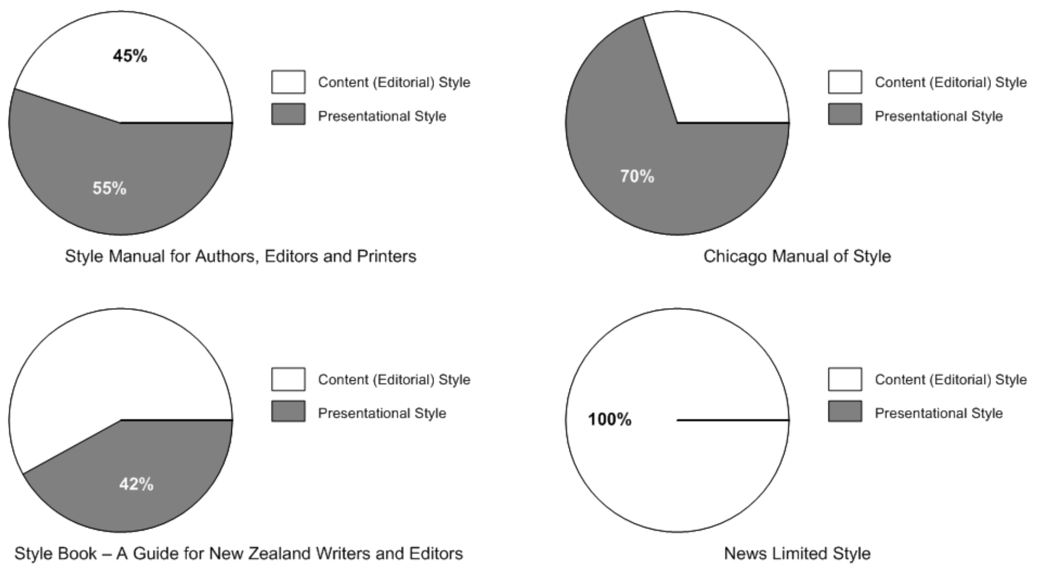


Figure 43: Breakdown of Content (Editorial) and Presentational Style Rules in Four Style Guides

Unlike article publishing organisations such as STC, the separation of content and form in the newspaper publishing process, particularly at News Limited, is complete. Journalists submit stories to the news content management system without any presentational style information. Those stories (which are modular in nature) are single-sourced to Web editions of the paper, traditional paper editions, syndicated newspapers, database archives, and RSS XML feeds.

The Challenges of Editing in Single-Sourced Publications

According to the *Freelance Editorial Association* *(n.d.)*, there are four different categories of editing:

- developmental (editing during the content creation process)
- substantive (improvements after the writer has completed the manuscript)
- copy (correcting errors and enforcing style manual rules)
- proofreading (checking the final version for typographical and layout errors).

Other sources categorise developmental and substantial as being synonyms, and some prefer the term *line editing* to *copy editing*. Some forms of editing, such as photo editing, largely involve the selection and rejection of alternative options.

The *Chicago Manual of Style (1982)* divides the editorial process into *mechanical editing* and *substantive editing*. Mechanical editing involves such matters as capitalisation, spelling, agreement of subjects and verbs, punctuation, and use of numbers. Substantive editing involves re-writing, re-organising, and reworking of the writing style to conform with the guidelines in a style manual.

The respected JPL publication *The Levels of Edit (Van Buren and Buehler, 1980)*, has a more comprehensive *editing typology*, which provides more subtle distinctions in nine types of edits:

- Co-ordination: planning and preparing a manuscript for further editing
- Policy: ensuring the manuscript conforms to company policy (and document framework)
- Integrity: ensuring cross-references, page numbers and paragraph and list numbering is correct
- Screening: checking spelling, punctuation, missing material, and subject/verb matching
- Copy Clarification: identifying hyphenation points and marking the manuscript to denote typesetting requirements
- Format: specifying typographical and layout requirements
- Mechanical Style: ensuring the manuscript conforms to style manual rules
- Language: checking the spelling, expression, punctuation, syntax, fluency, parallelism, terminology and appropriateness of titles
- Substantive: deals with the coherence of the manuscript and the meaningfulness of the content.

These nine types are used in five different levels of editing, as shown in the table below.

Table 6: Types of Edit and Levels of Editing

Type of Edit	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Level 5**
Co-ordination	X	X	X	X	X
Policy	X	X	X	X	X
Integrity	X	X	X	X	
Screening	X	X	X	X	
Copy Clarification	X	X	X		

Type of Edit	Level 1	Level 2	Level 3	Level 4	Level 5
Format	X	X	X		
Mechanical Style	X	X			
Language	X	X			
Substantive	X				

Other editing functions not categorised may include reducing the length of a manuscript.

Because of its authority, for the purposes of this paper, we will use the *Van Buren and Buehler* editing *typology* (see page 191): co-ordination, policy, integrity, screening, copy clarification, format, mechanical style, language and substantive. The STC edits can be organised into the editing *types* as follows.

Table 7: Organisation of STC Intercom Edits into Editing Typology Types

Nature of Edit	**Editing Typology *Types***
Capitalisation of headings	Mechanical Style
Wording	Substantive
Serial comma	Language
Formatting style	Format
Writing style	Substantive
Externalising links	Mechanical Style
House style	Mechanical Style
Punctuation	Language
AU to US spelling	Language
Acronym usage	Language

The edits fitted most comfortably into just four of the nine *types*: language, mechanical style, substantive, and format.

The Challenge of Content Re-use

In a modular DITA documentation workflow, documents are assembled from a pool of standalone topics, and those topics can be re-used in many different publications. *Write once and once only* (WOOO) is an underlying principle of modular documents; this approach results in lower creation and maintenance costs, and reduces the time required to produce different publications.

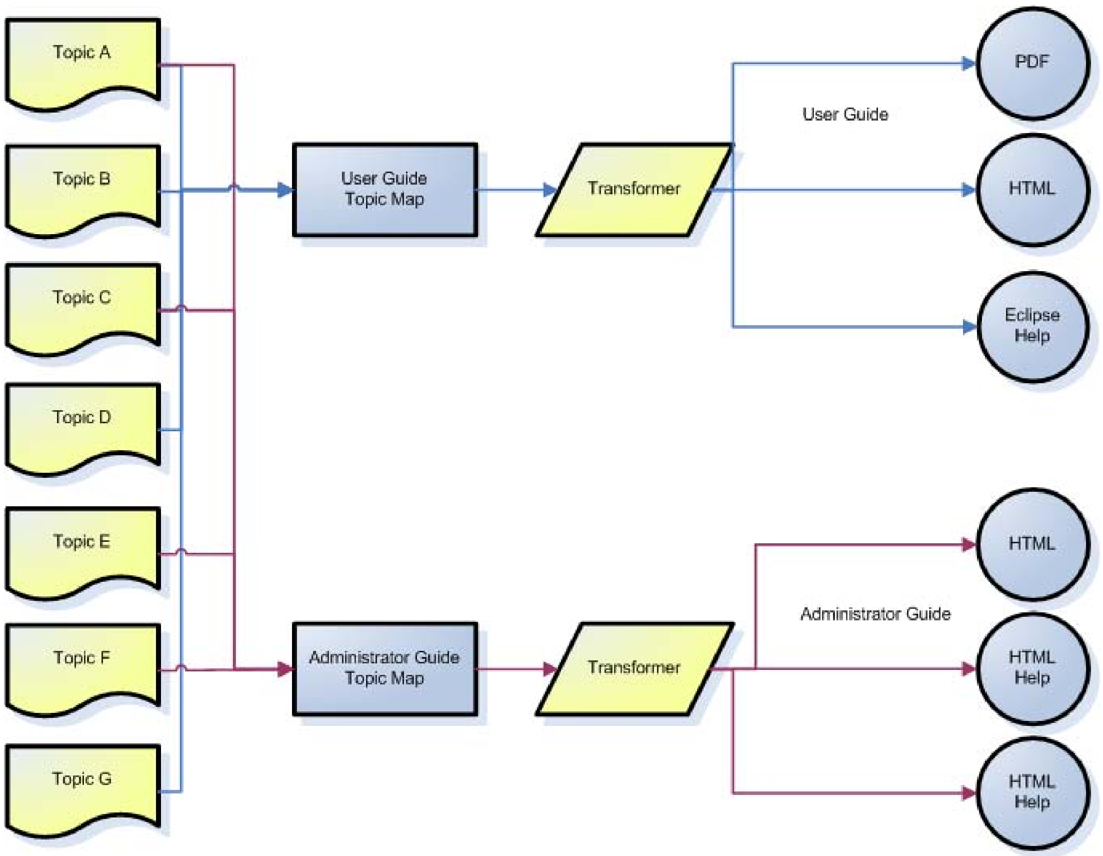


Figure 44: Modular Single-Sourcing Schematic

When DITA is used for magazine article or whitepaper content, the context suits a single topic rather than a multi-topic structure. Single-sourcing is therefore limited to producing different variations of output deliverable from the same source topic, as shown in *Figure 45: Non-Modular Single-Sourcing Schematic* (see page 194).

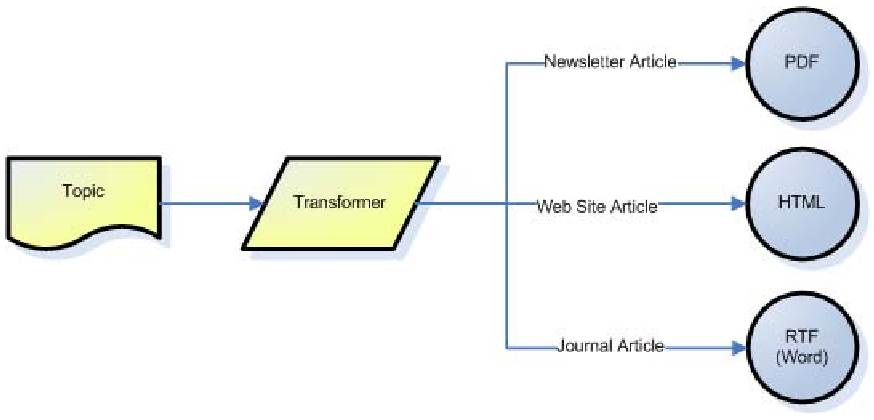


Figure 45: Non-Modular Single-Sourcing Schematic

If the main benefit of single-sourcing is the reduced cost of maintaining the content, does it really matter that some publication-specific changes are made prior to print publication? The answer to this question is probably “no, it doesn't matter”, but the fact that virtually all printed publications are also published online does complicate the matter. Print journals such as *Intercom* are *snapshots* of what a document was at a point in time, while online content is often *continually published*, in that there is not a fixed publishing date, and the content may be updated at any time. Perhaps this was the wrong question, in any case, because DITA represents the future of publishing, whereas *snapshot* print publishing represents the past. Future publishing models are likely to include continuous publishing to paper media, through devices such as the Espresso *(DADirect, n.d.)*, and to non-paper media including E-Ink *(E-Ink Corporation Web Site, n.d.)*.

The Challenge of Single-Sourcing with Different Style Requirements

A DITA workflow effectively separates the content of a document from its presentation, format and delivery (“the separation of content and form”). Authoring mark-up of the document is based on the semantics of the content elements, rather than how the content is to be presented to the reader. Publishing is largely an automated process, where DITA semantic mark-up is transformed, using software style rules, to a reading format. Some of the types of edits described by *Van Buren and Buehler* are concerned with formatting and presentation; in a DITA workflow, this sort of editing is a one-off verification of the transformation style rules rather than a per-document task.

The editing process in a DITA workflow is therefore simpler than that of a (traditional) style-based document workflow. However, the editing task that confirms that a document conforms to the requirements of a company's policies, or to the requirements of a style guide, is still required.

The delivery publications for the *What if Readers Can't Read* article have different requirements. For example, the *HyperWrite Web site* requires its content to be in Australian English, while the STC *Intercom* magazine requires its content to be in US English. A number of conflicting requirements of style between those two deliverables are shown in the table below.

Table 8: Comparison of HyperWrite and STC Publication Requirements

***HyperWrite Web Site* Requirements**	**STC *Intercom* Magazine Requirements**
Sentence case for section headings	Title (*maximal*) case for section headings
Australian English spelling and words	US English spelling and words
Inclusion of serial comma	Avoidance of serial comma
Sentence can start with numeric form of large number	Sentence should start with spelled form of large number
Percentage symbol used to express percentage fractions	"percent" (spelled) used to express percentage fractions

Some edits were subjective, in that the editor might prefer, for example, the phrasing “in one short (under 4 minutes) video” over the original “in one 3 minute and 51 second video”. Other edits reflected the different requirements of online and print media, such as the need for hyperlink targets to be separated from the link text for print media.

The changes suggested by the *Intercom* editor make the article as published the *Intercom* different to the version published on the HyperWrite Web site. True single-sourcing would see the one article source *master* being used for all publications, not modified versions or derivations of that single source. Once an article is edited from its original source, any changes or updates will have to be repeated throughout any derivations, thus denying the cost-saving efficiencies of true single sourcing.

Minimising Publication-Specific Edits

For single-sourcing to be practised, the edits specific to a particular publication must be eliminated, or at best, reduced.

The *Intercom* edits made to the *What if Reader's Can't Read* article fitted into the following four editing typology *types*:

- language (spelling and punctuation changes)
- mechanical style (changes to conform to policies and style guides)
- substantive (improvements to meaningfulness, often subjective)

- format (typographical and layout presentational changes - media-specific).

Language edits are usually contentious only when the differences are cultural or dialectical, and such differences can be handled through either DITA or complementary XML technologies such as XML Localization Interchange File Format (*XLIFF*) *(see Raya, 2004)*. Mechanical style edits are only problematic when different style guides are in conflict; this conflict may be overcome by the introduction of a common style guide, developed in an open source ideological framework. Ideally, subjective edits (that is, changes to suit the personal preferences of an individual editor) can be rejected, although any such diminishment of an editor's role is likely to invoke resistance. Format edit types for an individual article are not applicable to an automated publishing workflow, so such changes would not arise in a wholly DITA environment.

Language (Spelling) Differences in a Single-Source Environment

DITA includes a mechanism for marking up words or phrases (*inline elements*) for special processing in the publishing phase. For example, the word “realise” could be marked up as <ph translate="yes">; and then extracted (along with any other Australian English phrases) into a list of words to be translated into US English (“realize”). The translated phrases could then be re-injected into the document skeleton to form a translated document. (The XLIFF technology makes this sort of extraction and re-injection of phrases a technically trivial matter.) For this approach to work, the document would need to pass through a mini-translation process, and the author would have to be sufficiently disciplined to remember to mark up such phrases. The publisher would also need XLIFF extraction and translation software.

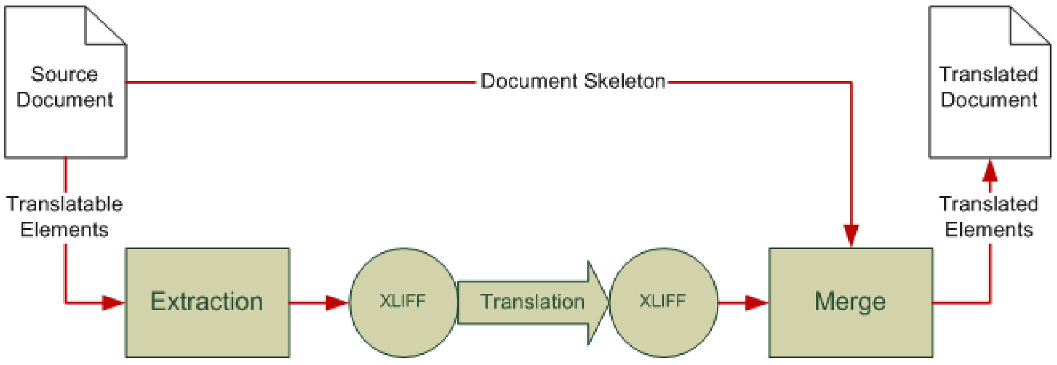


Figure 46: Schematic diagram showing XLIFF in the localisation process

Currently, the labour cost of the mark-up and translation would probably exceed the cost of manually editing the content for each publishing output. Feature-rich, DITA-aware XLIFF tools are not yet readily available, but when such tools reach the market, the reduced mark-up costs would likely make this mini-translation approach cost-effective.

Another option would be to use DITA's content referencing mechanism to place contentious words in a separate file, *transclude* those phrases into the article topic, and then provide two different versions of the separate file for different language versions of the article.

As well as spelling changes, colloquial wording changes might be necessary for different readerships. For example, in the USA, the *octothorpe* symbol (#) is colloquially described as a *pound sign*, while in New Zealand the same symbol is described as a *hash symbol*, and in the UK as the *square*. It would be optimistic to expect an author to be familiar with the colloquial nature of many words. In some cases, word use changes from region to region within the same country, such as the Queenslander Australian use of the word "scallop" to mean what Victorian Australians refer to as a "potato cake". Dialectical differences are not a new problem, however, and *controlled language* (such as *Simplified Technical English*, *ASD-STE100*) is normally used to minimise confusion in terms in technical documents *(Muegge, n.d.)*.

Mechanical Style (Writing and Wording Style) Differences in a Single-Source Environment

Different style manual rules often conflict, so an article written to one style guide may break the rules of another.

For example, the *Manual of Style (Snooks & all)* calls for the separator in numbers greater than three digits to be a space. The *What if Readers Can't Read* article uses a comma as a separator in the number "2,000". The *Style Book (Wallace and Hughes, 1995)* suggests (but does not directly mandate) the use of a comma as a separator. The *Chicago Manual of Style (1982)* dictates a comma separator for general numbers, but a space separator for "scientific copy". Although it is possible that numbers could be semantically identified in DITA, and then processed according to the particular style guide rules, this is likely to be impractical.

It seems, therefore, that the conflicting rules of style manuals represent an obstacle on the path to DITA single-sourcing across organisations with different style manuals. If all publications used a common style manual, the obstacle would dissolve.

Substantive Edits (Subjective Wording Changes) in a Single-Source Environment

If one editor prefers a particular phrasing of a sentence, but another prefers an alternative, single-sourcing cannot practically occur. A poor workaround is to hold off subjective changes until the latest possible point in the publication process, and then repeat those changes if or whenever the source document changes.

Editors are likely to resist any restrictions on their ability to make subjective changes. Interestingly, the problem is not with subjective edits as such, but with conflicting subjective edits.

Format Edits (Media-Specific Changes) in a Single-Source Environment

Any special requirements of a particular media can be easily catered for by changes in the transformation process that generates deliverable output documents from DITA source files. The issues of differing media requirements can be addressed by improving the transformation processing. The *What if Readers Can't Read* article was transformed to Word format for submission to the STC *Intercom* magazine using the standard DITA OT DITA-to-RTF rules. These rules could have been customised to format DITA cross-references so that the text of the link was unadorned (no underline, no colour) and the link target (a Web URL) written out after the link text and enclosed in brackets.

The Same Songsheet

It might be stating the obvious, but for single-sourcing with DITA to be effective, all parties to the single-sourcing have to be DITA or XML environments. Currently, very few article publishing organisations are using DITA, although many are using XML. The *Australian Journal of Communication* is a typical publisher of academic papers. Its submission requirements are:

> *All contributions should conform to current APA documentation style and to guidelines for inclusive language. Contributions should also be accompanied by a 50-100 word abstract. Contributors whose papers are accepted for publication will be required to submit a final copy of their paper, in hard copy and via e-mail to <e-mail address>. Diagrams and tables should be submitted in a separate MS Word document.*

Most of the efficiencies in moving to a DITA workflow will be lost if Microsoft Word® is a key part of the publishing cycle. Single-sourcing in a Word or similar non-structured, style-focussed technology is largely hypothetical, because a non-structured document technology does not lend itself to content re-use through the separation of content and form. The importance of single-sourcing will arise in the future when publishing is universally XML-based *(see Kulik and Nigloschy, 2003)*. (This paper is concerned with those future scenarios.)

The separation of content and form in a DITA process must be matched by a separation of content rules and form rules that define a publication's style.

Assuming that all parties in a future publishing scenario were working in an XML environment, or a DITA environment in particular, *interchangeability* will grow in importance. At the centre of interchangeability practice is standards; it is logical to conclude that single-sourcing in a DITA environment will rely on a number of standards, including editing and writing style standards.

DITA is not yet common in a publishing workflow, but paper-centric tools are. Technological changes such as E Ink devices, on-demand paper publishing and printing (through equipment such as the Espresso “book ATM”), digital distribution of books and magazines, and multimedia e-books *(see Elgan, n.d.)* will drive the demand for smarter and more appropriate publishing solutions, such as DITA-based workflows.

Section Requirements in Style Guides

Some style guides stipulate certain sections that must be included in a document. For example, the APA Style Manual requires the following sections in academic papers *(Source: http://owl.english.purdue.edu/owl/resource/560/01/)*:

- Title Page
- Abstract
- Main Body
- References

DITA's ditamap structure, along with *conditional publishing* capabilities, makes generation of different required sections for different publications a relatively trivial matter, but it does require that the granularity required of the document to meet different section requirements is analysed and implemented carefully.

Depending on the particular requirements, it might be necessary for the author of an article to create a specific ditamap for each publication in which the content is to be used.

Recommendations

In the case study, four types of edits that impede the realisation of single-sourcing in a future DITA-based publishing environment were identified. The impediments can be overcome through the following changes to editing practice.

Table 9: Recommendations for Changes to Editing Practice

Edit Type	**Recommendation**
Language Edits	Use XLIFF or transclusion to cater for different spelling and word usage in different outputs. The option of forcing a controlled vocabulary on authors of magazine articles should not be adopted, as it is likely to unreasonably constrict creativity and result in bland, uninteresting documents.
Mechanical Style Edits	Eliminate publication-specific changes by adopting a standard, open source, trans-organisational content style manual,

Edit Type	Recommendation
	and restrict corporate style guides to issues relating to formatting presentation. (Refer to *Figure 48: Possible Post-DITA Style Guide Cascade* (see page 201).)
Substantive Edits	Remove the substantive editing task and place a greater emphasis on the developmental editing stage. Abandon post-developmental subjective editing changes.
Format Edits	Address format edit changes at the transformation rules stage, at an organisational level rather than at the article or publication level.

Publishers in a DITA workflow should use custom XSL transformation routines, rather than the default DITA OT transformers, to ensure that format *edit types* are not required beyond a one-off verification of the publication style rules after the transformer is created. This would address to issues such as hyperlink references in printed output. Provided the link target and the link text are semantically identified in the source, references can be created to suit the publication's individual formatting and layout preferences without affecting the XHTML output.

The most important and significant of the above recommendations is to adopt a standard, open source, trans-organisational, international content style manual.

Open Source Standard Content Style Manual

An editing process uses a number of authorities for establishing the rules for the wording, structure and presentation of documents. Rules are built through a *cascade* of authorities. For example, a typical style ruleset might require spelling to conform to the *Macquarie Dictionary*, except when a different spelling is recommended by the *Microsoft Manual of Style, 2004*. A typical style rules cascade is depicted in *Figure 47: Typical Existing Style Guide Cascade* (see page 201).

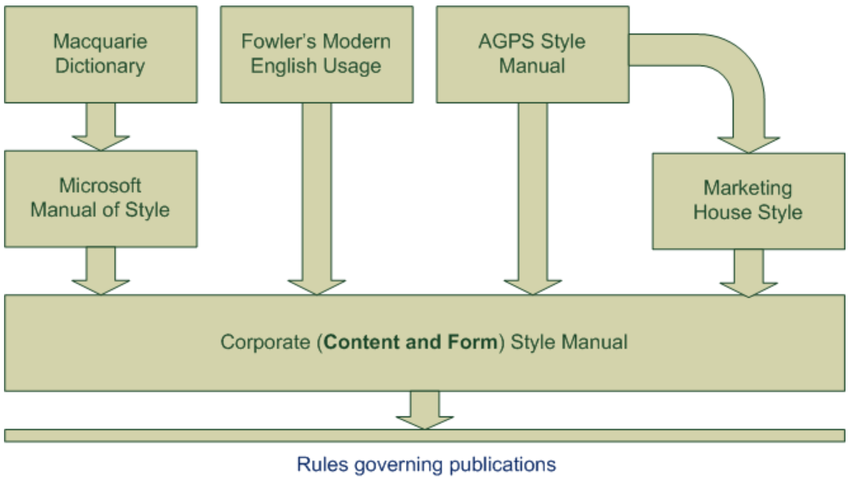

Macquarie Dictionary
Fowler's Modern English Usage
AGPS Style Manual
Microsoft Manual of Style
Marketing House Style
Corporate (**Content and Form**) Style Manual
Rules governing publications

Figure 47: Typical Existing Style Guide Cascade

As shown in *Figure 2: Breakdown of Content (Editorial) and Presentational Style Rules in Four Style Guides* (see page 190), most style manuals used in article publishing mix form (presentational style) rules with content (writing and editorial style) rules, making them difficult to interpret in a semantic mark-up authoring environment.

To improve the prospects of content re-use, an agreed standard or style manual for formalising editing decision-making is vital. Open source development is a proven approach to the creation of trans-organisational and international standards *(Weber: 2005)*. An open source style manual might fit into a style rules *cascade* as shown in *Figure 48: Possible Post-DITA Style Guide Cascade* (see page 201). The standard style manual should be concerned only with writing style (ie, content), and not with presentational style (ie, form).

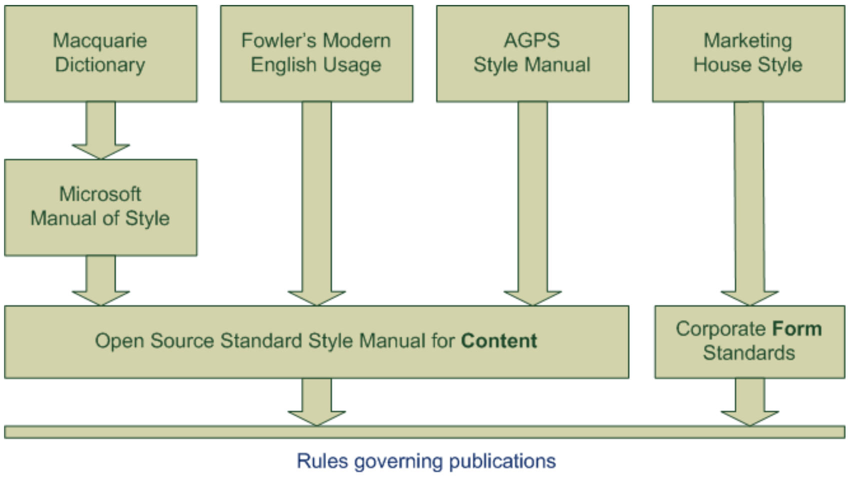

Macquarie Dictionary
Fowler's Modern English Usage
AGPS Style Manual
Marketing House Style
Microsoft Manual of Style
Open Source Standard Style Manual for **Content**
Corporate **Form** Standards
Rules governing publications

Figure 48: Possible Post-DITA Style Guide Cascade

Necessary Changes to *Editing Typology*

The separation of content and form in the publishing process, the adoption of an open source style manual, the increasing importance of continuous publishing, and the use of automated document generation processes will require a re-thinking of the Van Buren and Buehler editing typology. The schematic diagram *Figure 49: Editing in Structured and Unstructured Publishing Models* (see page 202) illustrates a comparison between the editing types in a DITA publishing workflow and existing style-based publishing workflow.

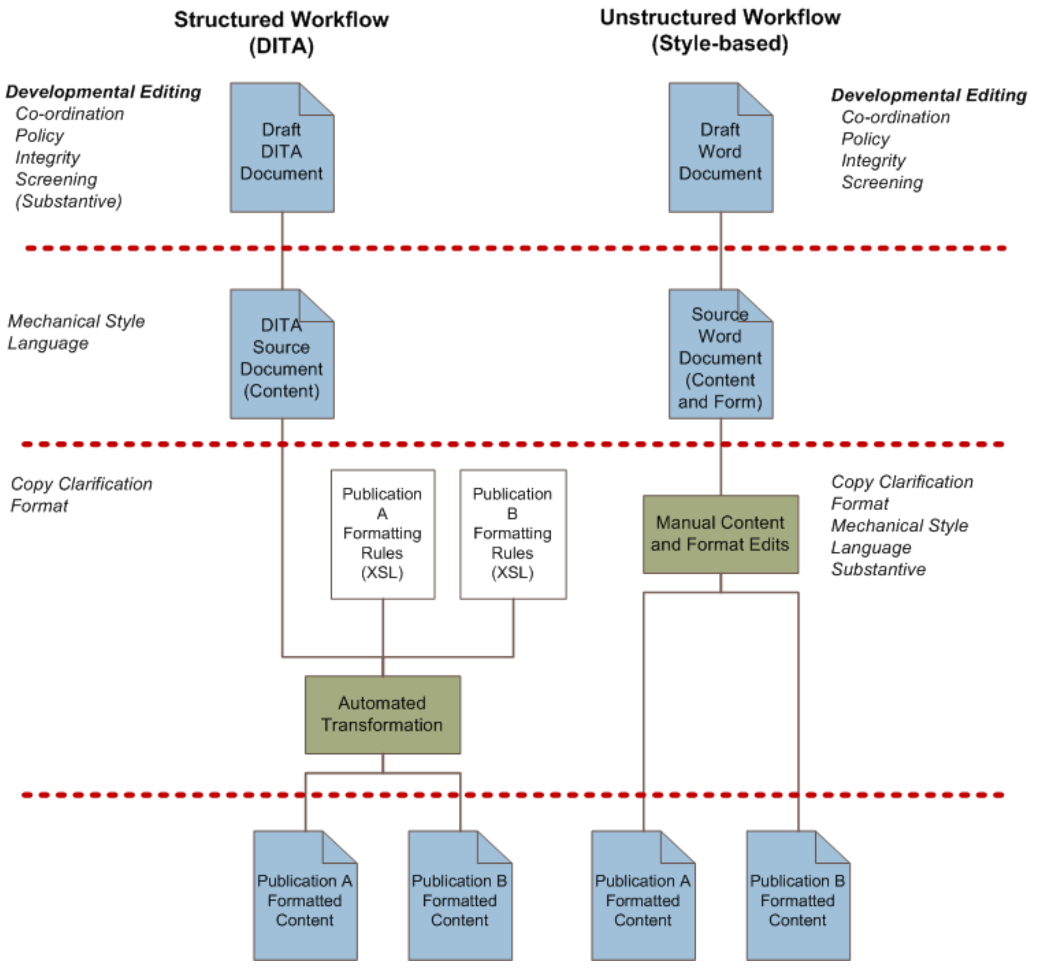

Structured Workflow (DITA)
Unstructured Workflow (Style-based)
Developmental Editing
Co-ordination
Policy
Integrity
Screening
(Substantive)
Draft DITA Document
Draft Word Document
Developmental Editing
Co-ordination
Policy
Integrity
Screening
Mechanical Style
Language
DITA Source Document (Content)
Source Word Document (Content and Form)
Copy Clarification
Format
Publication A Formatting Rules (XSL)
Publication B Formatting Rules (XSL)
Manual Content and Format Edits
Copy Clarification
Format
Mechanical Style
Language
Substantive
Automated Transformation
Publication A Formatted Content
Publication B Formatted Content
Publication A Formatted Content
Publication B Formatted Content

Figure 49: Editing in Structured and Unstructured Publishing Models

Any attempt to standardise punctuation (such as serial commas) and even spelling is bound to meet dogged resistance, as it is natural for people to protect their own language use and conventions. The creation of a standard style manual that crosses organisational culture and national cultural boundaries is not going to be without major challenges. It is important, therefore, that provision for dialectical and spelling differences through a *mini-translation* XLIFF process be developed in parallel with a standard style manual.

Further Research

Further research is required to analyse the acceptance of changes in editing practice amongst editing professionals, and what steps would need to be taken to develop and manage an open source style manual for single-source publishing.

References

- *Capitalization - Heading and publication titles*. (n.d.). Wikipedia. Retrieved January 9, 2009, from http://en.wikipedia.org/?title=Title_case#Headings_and_publication_titles
- DADirect. (n.d.). YouTube - A&R *Espresso Book Machine*. Retrieved January 9, 2009, from http://au.youtube.com/watch?v=MXP0E7i0bfU
- Day, D., Priestley, M., & Schell, D. (2005). Introduction to the Darwin Information Typing Architecture. IBM developerWorks. Retrieved February 9, 2009, from http://www.ibm.com/developerworks/xml/library/x-dita1/
- *E Ink Corporation* Web Site. (n.d.). . Retrieved January 9, 2009, from http://www.eink.com/
- Elgan, M. (n.d.). *Why Amazon's Kindle is revolutionary*. Computerworld. Retrieved January 9, 2009, from http://www.computerworld.com.au/index.php/id;748883531;pp;1;fp;4194304;fpid;1
- Flann, E., & Hill, B. (2004). The Australian editing handbook (2nd ed.). Milton, Queensland: John Wiley and Sons.
- Freelance Editorial Association. (n.d.). *Types of Editing*. Retrieved January 9, 2009, from http://www.edsguild.org/types.htm
- Kulik, B., & Nigloschy, D. (2003). *XML Centric Workflows: New Opportunities for Publishers*. Retrieved January 9, 2009, from http://www.idealliance.org/papers/dx_xml03/papers/03-06-03/03-06-03.pdf
- Lockwood, K. (2005). Style: The essential guide for journalists and professional writers (3rd ed.). Melbourne: News Custom Publishing.
- Manning, S. (n.d.). *XML: To Train, or not to train*. The Rockley Report. Retrieved December 17, 2008, from http://www.rockley.com/TheRockleyReport/V1I4/People%20Processes%20and%20Change.htm
- Microsoft® Manual of Style for Technical Publications, Third Edition, 2004, ISBN 0-7356-1746-5
- Muegge, U. (n.d.). *Controlled language: the next big thing in translation?* Translation Directory. Retrieved March 10, 2008, from http://www.translationdirectory.com/articles/article1359.php

- Publication Manual of the American Psychological Association, 5th Edition, Washington DC, 2001, APA
- Raya, R. (2004). *XML in Localisation: Use XLIFF to translate documents*. IBM developerWorks. Retrieved January 9, 2009, from http://www.ibm.com/developerworks/xml/library/x-localis2/
- *Serial comma*. (n.d.). Wikipedia. Retrieved January 9, 2009, from http://en.wikipedia.org/wiki/Serial_comma
- Self, T. (2008). *What if Readers Can't Read?* HyperWrite. Retrieved January 9, 2009, from http://www.hyperwrite.com/Articles/showarticle.aspx?id=84
- Snooks and Co. (2002). Style manual: for authors, editors and printers (6th ed.). Brisbane: Wiley Australia.
- The Chicago Manual of Style. (1982). (13th ed.). Chicago: University of Chicago Press.
- Van Buren, R., & Buehler, M. F. (1980). *The Levels of Edit*. Pasadena, California: Jet Propulsion Laboratory, 1980 Retrieved December 13, 2008.
- Wallace, D., & Hughes, J. (1995). Style Book - A Guide for New Zealand Writers and Editors (5th ed.). Wellington: GP Publications.
- Weber, Steven (2005). The Success of Open Source. Harvard: Harvard University Press.
- Weiss, E. H. (2005). The Elements of International English Style : A Guide to Writing Letters, Reports, Technical Documents, and Internet Pages for a Global Audience. New York: M.E. Sharpe.
- XML Cover Pages: *DITA 1.1 Appproved as an OASIS Standard*. (2007, August 13). The Cover Pages. Retrieved June 8, 2009, from http://xml.coverpages.org/ditaV11-Standard.html

Improved glossary and terminology handling

DITA 1.2 introduces significant improvements to the way in which terminology is handled. A new glossary entry element provides a flexible semantic structure to store terminology information, and the new indirect linking functionality allows terms occurring in the content to be linked through a key reference or transcluded using a new `abbreviated-form`

> *element. Keys used for key referencing are defined in a new `glossref` specialisation of `topicref` in the map.*

Purpose of a glossary

The purpose of a glossary is to allow readers who encounter terms that they do not understand to look up a definition for the term. Terms may be single words or longer phrases, they may be acronyms or abbreviations, or they may be domain-specific jargon. In a printed publication, a glossary (usually referred to as a *Glossary of Terms*), appears at the front or back of the book content. In an online document, the glossary is often accessed from a link on the document toolbar and typically displays in a separate window or frame from the document content proper. Many online documents make the glossary easier to access by hyperlinking terms in the content to the associated definitions in the glossary.

Purpose of the change to glossary-related elements in DITA 1.2

The new elements introduced in DITA 1.2 will make it easier for publishing tools to automatically hyperlink terms appearing in the text with their respective definitions in the glossary (in online document outputs). The new elements will also make it easier for users to semantically identify terms and their definitions through a new `glossentry` information type.

The `glossentry` is a specialisation of the concept information type.

A new `glossref` element has also been introduced to provide for the possibility of linking terms in the document content with the glossary definitions.

There are three distinct processes in setting up a glossary in DITA.

- Creating individual glossary topics (using the `glossentry` information type)
- Incorporating the glossary topics into a ditamap
- Linking terms in content topics to their glossary definitions.

Benefit to users

The new glossary-related elements will open up opportunities, through improvements in publishing tools, for the following features in output documents:

- Automated pop-up or expansion linking of terms in the content to definitions in the glossary.
- Generation of a rich and comprehensive Glossary of Terms for a printed document.
- Generation of a dynamic Glossary of Terms (perhaps sortable and searchable) for an online document.

Another benefit will be the ability to use a term in an abbreviated form (for example, an acronym) or in a longer form, yet still identify the relationship between the term and its definition.

The `glossentry` information type

The `glossentry` information type contains an individual glossary entry which is made up of the following elements.

Element	Element Name	Provides	Example
Glossary term	`glossterm`	The preferred form of the term.	`<glossterm> USB flash drive </glossterm>`
Definition	`glossdef`	The definition or explanation of one sense of the term.	`<glossdef> A small portable drive. </glossdef>`
Body of the glossary	`glossBody`	Container for more details about a glossary term (such as part of speech or additional forms of the term)	`<glossBody><... /></glossBody>`

In turn, the optional `glossBody` element contains the following elements, used to store more detailed information about the glossary entry.

Element	Element Name	Provides	Example
Surface form of term	`glossSurfaceForm`	The unambiguous full form of the term (which may include the acronym or short form in parentheses). The surface form is suitable to introduce the term in new contexts	`<glossSurfaceForm> USB flash drive (UFD, or flashie) </glossSurfaceForm>`
Part of speech	`glossPartOfSpeech`	The part of speech of the preferred form of the term.	`<glossPartOfSpeech value="noun"/>`
Symbol or icon	`glossSymbol`	Reference to an image file used as a synonym for the term.	`<glossSymbol href="ufd_logo.jpg" scope="local"> <alt> Identification logo for USB flash drives </alt> </glossSymbol>`
Notes on scope of usage	`glossScopeNote`	Information on what the term does or does not apply to.	`<glossScopeNote>Not to be used for other flash memory cards that do not use a USB interface</glossScopeNote>`
Status of term	`glossStatus`	Business rule recording the status of the use of the term. The status is usually explained with `glossUsage`.	`<glossStatus value="obsolete"/>`
Correct usage of term	`glossUsage`	Supplementary information explaining the correct use of the term.	`<glossUsage>Do not use the term in title case (as in "USB Flash Drive") because that suggests a trademark.</glossUsage>`

Element	Element Name	Provides	Example
Variants of the term	`glossAlt`	Container for alternative representations of the glossary term (such as an acronym used for the term)	`<glossAlt><... /></glossAlt>`

In turn, the optional `glossAlt` element contains the following elements, used to store more detailed information about the glossary entry.

Element	Element Name	Provides	Example
Abbreviated term	`glossAbbreviation`	An abbreviated form of the term	`<glossAbbreviation> Flash </glossAbbreviation>`
Acronym	`glossAcronym`	An alternative form of the term as an acronym	`<glossAcronym> UFD </glossAcronym>`
Association with another alternative form	`glossAlternateFor`	Cross-reference to a similar variant of the same term.	`<glossAlternateFor href="#usbfd/memoryStick"/>`
Short form of term	`glossShortForm`	A short form of the preferred form of the term, but not an acronym	`<glossShortForm> flashie </glossShortForm>`
Status of variant	`glossStatus`	Business rule recording the status of the use of a variant of a term. The status is usually explained with `glossUsage`.	`<glossStatus value="prohibited"/>`
Notes on usage of the term	`glossUsage`	Supplementary information explaining the usage of the variant of the term.	`<glossUsage>This is too colloquial.</glossUsage>`
Synonym	`glossSynonym`	A synonym of the glossary term.	`<glossSynonym>memory stick</glossSynonym>`

The `glossgroup` information type may be used to collect multiple glossary entries into a single topic. It serves a similar purpose to the *ditabase* information type, being a nesting container for `glossentry` topics.

Example of a complete `glossentry`

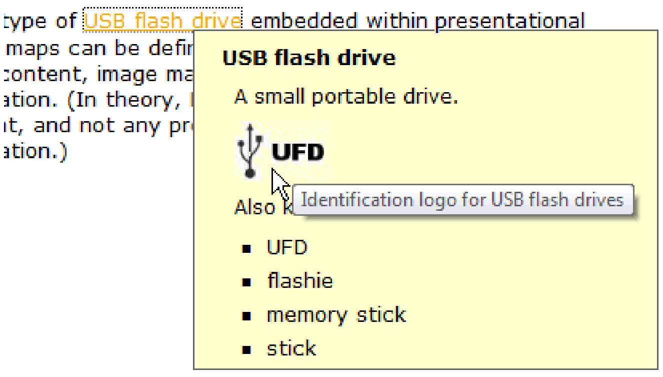
The following is a glossary entry for *USB flash drive*.

```
<glossentry id="usbfd">
  <glossterm>USB flash drive</glossterm>
  <glossdef>A small portable drive.</glossdef>
  <glossBody>
    <glossSurfaceForm>USB flash drive (UFD, or
flashie)</glossSurfaceForm>
    <glossPartOfSpeech value="noun"/>
    <glossUsage>Do not provide in upper case (as in
"USB Flash Drive") because that
    suggests a trademark.</glossUsage>
    <glossSymbol href="ufd_logo.jpg" scope="local">
```

```
       <alt>Identification logo for USB flash
drives</alt>
    </glossSymbol>
    <glossScopeNote>Not to be used for other flash
memory cards that do not use a USB
      interface.
    </glossScopeNote>
    <glossAlt>
      <glossAcronym>UFD</glossAcronym>
      <glossUsage>Explain the acronym on first
occurrence.</glossUsage>
    </glossAlt>
    <glossAlt>
      <glossShortForm>flashie</glossShortForm>
      <glossUsage>Usually describes low cost
devices.</glossUsage>
    </glossAlt>
    <glossAlt id="memoryStick">
      <glossSynonym>memory stick</glossSynonym>
      <glossUsage>This is a colloquial
term.</glossUsage>
    </glossAlt>
    <glossAlt>
      <glossAbbreviation>stick</glossAbbreviation>
      <glossAlternateFor href="#usbfd/memoryStick"/>
    </glossAlt>
    <glossAlt>
      <glossAbbreviation>flash</glossAbbreviation>
      <glossStatus value="prohibited"/>
      <glossUsage>This short form is
ambiguous.</glossUsage>
    </glossAlt>
    <glossAlt>
  </glossBody>
</glossentry>
```

How a glossary entry might be rendered

Publishing tools may provide different methods of rendering glossary elements. The following shows an example of how a publishing tool may render a glossary entry in HTML form.



type of USB flash drive embedded within presentational
maps can be defi
content, image ma
ation. (In theory,
it, and not any pr
ation.)

USB flash drive

A small portable drive.

UFD

Identification logo for USB flash drives

Also k

- UFD
- flashie
- memory stick
- stick

Figure 50: Hypothetical example of how a DITA glossary entry may be rendered in HTML.

Incorporating a glossary in a ditamap

Glossary topics (both `glossentry` and `glossgroup`) can be incorporated into a ditamap using a standard `topicref` element or through a new `glossref` element.

The `glossref` element has a required `keys` attribute used to help link terms in content topic (*inline terms*) to their glossary definitions. Rather than linking an inline term to a particular glossary topic file name, you can link the term to the keys attribute of the relevant `glossref` entry. This *indirect addressing* mechanism allows greater flexibility for single-sourcing, where the glossary definition might be different for different collections of topics.

The diagram below shows the glossref being used for indirect linking of terms.

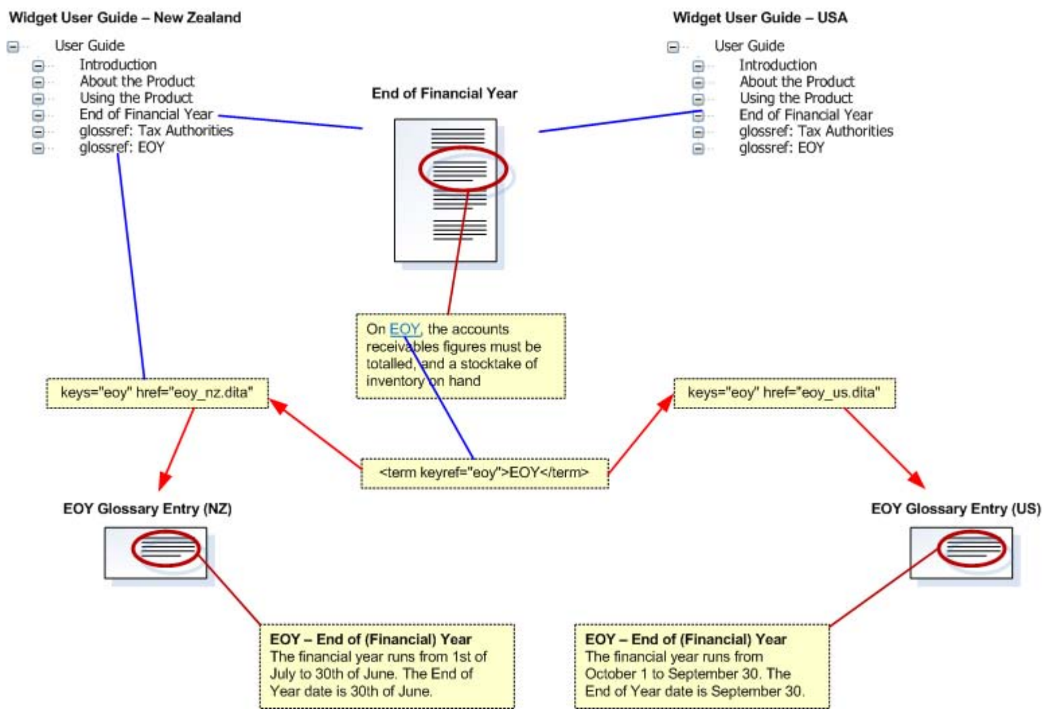

Widget User Guide – New Zealand
User Guide
Introduction
About the Product
Using the Product
End of Financial Year
glossref: Tax Authorities
glossref: EOY
End of Financial Year
Widget User Guide – USA
User Guide
Introduction
About the Product
Using the Product
End of Financial Year
glossref: Tax Authorities
glossref: EOY
On EOY, the accounts receivables figures must be totalled, and a stocktake of inventory on hand
keys="eoy" href="eoy_nz.dita"
keys="eoy" href="eoy_us.dita"
<term keyref="eoy">EOY</term>
EOY Glossary Entry (NZ)
EOY Glossary Entry (US)
EOY – End of (Financial) Year
The financial year runs from 1st of July to 30th of June. The End of Year date is 30th of June.
EOY – End of (Financial) Year
The financial year runs from October 1 to September 30. The End of Year date is September 30.

Figure 51: Indirect Referencing of Glossary Topics using keys

An important difference between `glossref` and `topicref` is that `glossref` won't result in the glossary topic being inserted in the output at the position the reference appears in the ditamap.

Creating a glossary section

A glossary section in an output document is defined by referencing, through normal `topicref` elements, all the glossary (`glossentry` and `glossgroup`) topics in a normal ditamap. The topics must be manually sorted into alphabetical order unless the authoring tool offers an automated topic sorting feature.

```
<map>
.
 <topicref href="abs.dita" />
 <topicref href="awd.dita" />
.
 <topicref href="wrx.dita" />
.
</map>
```

Linking terms in content topics

The linking of a term occurring in-line in a normal content topic to its relevant `glossentry` definition is a matter of associating the term (semantically identified in a `term` element) with the hypertext reference (`href`) or the key reference (`keyref`) of the glossary topic.

For example, the mark-up `<term keyref="eoy">EOY</term>` establishes the association between the term “EOY” and the glossary topic with a `keys` attribute of `eoy`. The related map entry may be `<glossref keys="eoy" href="eoy_nz.dita"/>`.

In addition to using the `term` element as the basis for linking, a new `abbreviated-form` element provides an alternative method of using glossary entries by substituting the element for the glossSurfaceForm or glossAcronym variants from a glossary entry on rendering. It is expected that a processor will substitute the first occurrence of an `abbreviated-form` element with the `glossSurfaceForm` variant in the glossary entry and subsequent occurrences with the `glossAcronym` variant of the glossary entry. The `abbreviated-form` element acts like a content reference but uses the `key` rather than the `href` as the reference.

For example, mark-up of:

```
<p>An <abbreviated-form keyref="abs"/> helps a driver to stop. For
this reason many find an <abbreviated-form keyref="abs"/> useful.</p>
```

might be rendered in HTML as:

```
<p>An Anti-lock Braking System (ABS) helps a driver to stop. For
this reason many find an ABS useful.</p>
```

where the glossref in the map was `<glossref keys="abs" href="abs.dita"/>`, and the `abs.dita` glossary entry topic was:

```
<glossentry>
  <glossterm>Anti-lock Braking System</glossterm>
  <glossBody>
    <glossSurfaceForm>Anti-lock Braking System
(ABS)</glossSurfaceForm>
    <glossAlt>
      <glossAcronym>ABS</glossAcronym>
    </glossAlt>
  </glossBody>
</glossentry>
```

About `keys` and `keyref` attributes, and indirect linking

Whenever a topic has a reference to other content, it makes the topic less reusable because of the dependency on the target being still available and still relevant. The `keys` and `keyref` attributes provide a simple redirection scheme that leverages existing attributes and map architectures to provide support for redirectable conrefs, topicrefs, xrefs, links, terms, and other reference elements and attributes. They also provide a simplified architecture for managing variable or volatile content (such as product names) which need to be easily swapped out when a topic is reused in new contexts.

A topic's keys are always defined in the map. A `keys` attribute can contain one or more keys. Keys resolve to the resources given as the `href` value on the topicref element with the matching key.

Summary

The new glossary features in DITA 1.2 offer many opportunities for processing glossaries and terms into rich features in output documents. The enhancements in DITA include new `glossentry` and `glossgroup` information types, a new `glossref` map element, a new `abbreviated-form` alternative to the `term` element, and the addition of a `keys` attribute to the `term` element to allow for indirect linking.

Hippies in cyberspace

The open source movement is driving many intellectual and technological innovations. As a social movement, open source has much in common with the hippy movement of the 1960s, sharing in particular a utopian view of sharing, trust, collaboration and collectivism as a pathway to a more equitable society. A new generation of hippies is inhabiting cyberspace, profoundly influencing the direction of the information society and the way in which knowledge is stored and shared. Education is a new frontier for the open source movement. To be able to meet the challenges of such an important change, an understanding of the hippy philosophy is key.

Imagine all the people, sharing all the world.

John Lennon (Imagine, 1971)

I believe that a desirable future depends on our deliberately choosing a life of action over a life of consumption.

Ivan Illich (Deschooling Society, 1974)

This is just the beginning, the beginning of understanding that cyberspace has no limits, no boundaries.

Nicholas Negroponte (CNN, 1999)

One of the icons of the socially tumultuous 1960s was the hippy. The hippy sub-culture movement started in the Haight-Asbury district in San Francisco in the early 1960s, possibly spawned from community discontent over a proposed freeway through the district. The early hippy visionaries were utopians, who believed a society radically different from the status quo could be created, incorporating ideals of peace, shared ownership (through co-operatives), personal freedom, and love, while abandoning middle-class values such as materialism and authoritarian institutions. The Haight-Asbury hippy community had disintegrated by 1968, but by then a broader movement had been established which still exists today.

In *Flowering of the Hippie Movement (Howard, 1969)*, the hippy movement was described as having “a fairly well thought-out alternative to conventional society”, built on an implicit assumption that “the example which they set in their own communities would induce change in the rest of society”. The hippy movement did not develop beyond being a fringe social movement, although some credit the movement for accelerating social advances including women's rights, anti-discrimination, and conservation *(Sieghart, 2007)*.

Figure 52: Contemporary hippy at the Rainbow Gathering in Russia

Certainly, the Internet bears signs of a hippy philosophy. Born in California at the end of the 1960s as *ARPANET*, the Internet is built upon notions of shared ownership, co-operation, freedom, trust, and a decentralised, minimalist regulation. There is no one organisation governing the Internet. The addressing standards are the responsibility of *ICANN (Internet Corporation for Assigned Names and Numbers)*, the core networking protocol standards are the responsibility of the *IETF (Internet Engineering Task Force)*, and hundreds of other organisations govern the networks, protocols and applications that constitute the Internet world. Both ICANN and IETF are supra-national, non-profit organisations. ICANN and IETF membership is open to all; likewise, meetings are free to attend, and open to all. Operations of the organisations are transparent.

Virtually all of the standards organisations that operate at a lower level than ICANN and IETF are also open to all, and transparent in operation. The *W3C (World Wide Web Consortium)*, which maintains Web and XML standards, is typical of these collaborative, open organisations; in many ways, it is run along the same lines as a co-operative. In fact, the W3C describes itself not as an organisation, but as a "community" *(About W3C, n.d.)*. One of the W3C's guiding principles is "Web for All", where the social value of the Web is acknowledged, with a goal of making the *benefits [of sharing knowledge] available to all people, whatever their hardware, software, network infrastructure, native language, culture, geographical location, or physical or mental ability.*

The open culture of Internet standards organisations is part of a larger *open source (or open standard) movement*. *Open source* can perhaps be defined as a set of principles, practices and philosophies that promote collaborative development of products and knowledge. The term is most commonly applied to the source code of software that is available to the general public with relaxed or non-existent intellectual property restrictions. Open source communities use licensing arrangements so that a technology can be communally-owned, but still protected by copyright law. Open source products are developed through incremental individual effort or through collaboration.

XML is an open standard (managed by the W3C) designed for the structured storage of all human knowledge. XML itself is a meta-language; that is, it is a set of rules for creating other languages, or *XML applications*. Nearly all XML standards applications are open source projects. Two XML applications commonly used for technical publications are DITA and DocBook, both of which are open standards. These standards are entrusted to *OASIS (Organisation for the Advancement of Structured Information Standards)*. As with most open standards organisations, decisions within OASIS are made collectively by groups of peers, and the deliberations are as open and transparent as possible. For example, the minutes of meetings, whitepapers, discussions and proceedings of open standards committees are generally available in the public domain.

Many software development projects are open source. Amongst the most well-known open source projects are:

Apache HTTP Server	the most popular Web server in the world, driving approximately 100 million Web sites *(Netcraft Web Server Survey, 2009)*
OpenOffice	a suite of general office software tools, including word processor, spreadsheet, presentation and database tools
Mozilla Firefox	the Web browser product of the Mozilla Organisation

The non-profit organisations, or *communities*, through which these collaborative software projects are developed use the same sort of terms to describe their objectives as did the early hippy visionaries. The Apache Software Foundation describes it workings as "a collaborative and meritocratic development process", and explains that the Foundation "is governed by the community it most directly serves”. The mission statement of OpenOffice.org describes an aim “to create, as a community, the leading international office suite that will... provide access to all functionality and data...”. The Mozilla Organisation has a manifesto which includes phrases such as “community of people who believe”, “openness, innovation and opportunity”, “worked together”, “benefits everyone”, “public good”, and “enriching the lives”. One of the four goals of the manifesto echoes the hippy objective of inducing social change by example: “provide a framework for other people to advance this vision of the Internet” *(The Mozilla Manifesto, n.d.)*.

> *As open source has begun over the last several years to attract more public attention, it has taken on a peculiar mantle and become a kind of Internet era Rorschach test. People often see in the open source software movement the politics they would like to see - a libertarian reverie, a perfect meritocracy, a utopian gift culture that celebrates an economics of abundance instead of scarcity, a virtual or electronic existence proof of communitarian ideals, a political movement aimed at replacing obsolete nineteenth-century capitalist structures with new "relations of production" more suited to the Information Age. Steve Weber (Weber, 2005)*

In the essay *Hippies 2.0 in museum 2.0*, Dick Rijken claims a direct connection between the hippy ideals of the sixties and the open source movement of the new millennium *(Rijken 2007)*.

> *They formulated their dreams in the sixties. They tried to make them real in the seventies. They got frustrated in the eighties when it all fell apart. They felt alienated in the nineties, but gathered tremendous material wealth as their careers progressed. They slowly got used to the digital life in the early years of the new millennium, as they learnt from their offspring. And, finally, they got it. In fact, they took over, as they became the perfect amateurs. They’re well educated, articulate, critical, wealthy, and, now, they’re retired. But not bored. Coming from an era where ‘quality’, ‘meaning’, ‘substance’ and ‘depth’ were important values, they express themselves and reflect on the world around them. Big time!*

Rijken identifies a new generation of hippy culture; a movement he calls “hippies 2.0”.

While younger generations spend their time chatting about nothing in particular and collecting unknown friends, hippies 2.0 are generating high quality content on a mind-boggling scale. They have time, they are independent, and they won't take 'no' for an answer.

Nicholas Negroponte, the noted futurist and author of *Being Digital*, once observed that if the Internet were a country, it would be the nicest place on earth *(Nicholas Negroponte Interview on Lateline, 1996)*. By and large, cyberspace is a friendly, co-operative place, but like every neighbourhood, it is also a reflection of the spectrum of humanity. Crooks, con artists, paedophiles, spies, thieves, vandals, pickpockets and government agents "travel" the Internet alongside volunteers, community workers, shopkeepers, students, nurses, traders, politicians, programmers and children.

The new generation of "hippies" have been quietly moulding the nature of cyberspace society, and upsetting the norms of capitalism in the process. The biggest encyclopaedia in the world is open source, created and maintained by volunteer "amateurs". The SourceForge open source software repository has 230,000 open source software projects (as at February 2009), contributed to by more than two million people *(About SourceForge, n.d.)*. Linux-based open source operating systems are now widespread in both business and consumer devices. And of course, the Web, the biggest open source project of all, continues to penetrate through all layers of human endeavour.

The imperative for knowledge and access to be free has highlighted the *gratis versus libre* ambiguity in the word "free", such that the term is now often refined into two: *free-as-in-beer* and *free-as-in-freedom* (or *free-as-in-speech*). The open source movement embraces both "frees".

Some *hippies 2.0* projects are overtly utopian. The *One Laptop Per Child (OLPC)* Association project, founded by MIT Media Lab's Nicholas Negroponte, has as its mission:

To create educational opportunities for the world's poorest children by providing each child with a rugged, low-cost, low-power, connected laptop with content and software designed for collaborative, joyful, self-empowered learning.

Photo: Mike McGregor

Figure 53: OLPC XO-1 Beta-1 Prototype

When it was formed in January 2005, the intention of the OLPC project was to develop a USD100 laptop computer that could be donated to children in developing countries. By October 2009, 1.125 million laptops had been distributed to children and teachers in countries including Sierra Leone, Rwanda, Ghana, Uruguay, Columbia, and Mexico.

The OLPC initiative was from the start an educational project, not a technical project. Many other educational projects are sprouting, as if following the examples and frameworks set by others.

The *University of the People (UoPeople)* is a non-profit online educational institution, with three fundamental values *(Reshef, 2009)* that have a *hippies 2.0* resonance:

- inclusion (access to higher education should be a right for all, not a privilege for the few)
- peer-to-peer teaching
- improvement of the world begins with the individual.

UoPeople's teaching model is built upon open source technologies, including the Moodle learning management system, and the breathtakingly large quantity of free online educational resources. The UK Open University's Knowledge Media Institute is one of a number of organisations providing free access to course materials. According to Peter Scott, director of the Institute, “so much high-quality material is now on the Web that traditional university models are no longer the only arbiters of quality” *(Swain, 2009)*.

The *International Federation for Learning, Education, and Training Systems Interoperability (LETSI)*, is another group of utopians focussed on collaboration in the teaching and learning space. LETSI believes that open standards are key to enabling learning applications that integrate and interoperate. LETSI is small in comparison to *Open Educational Resources (OER) Commons*, founded as recently as February 2007. Amongst the free resources provided by OER are full university courses, comprising readings, videos of lectures, assignments, lecture notes, assessment tasks and text books. The OER community sees equitable access to high-quality education as a global imperative, and aims to “grow a sustainable culture of sharing among educators at all levels” *(About OER, n.d.)*. Amongst OER's 120 major content contributors are Harvard University Law School, Massachusetts

Institute of Technology, Open University (UK), NASA, and Yale University. As at November 2009, there were over 16,000 post-secondary learning materials available from the OER Web site *(ibid.)*. Even the legal contracts licensing the free use, re-purposing and distribution of OER materials are open source; typically Creative Commons licences.

When confronted with the encroachment of open source projects, commercial information technology organisations have chosen between *agin 'em* and *with 'em*. Universities may have to make similar choices in the future. Microsoft chose the *agin 'em* option when confronted with OpenOffice's success in elevating the ODF document format as an international standard mandated by some governments. By contrast, IBM has deliberately adopted a *with 'em* approach, and has donated a number of its technologies to the open source movement. One such example is DITA, an XML-based technology and methodology for writing technical documents.

An assumption is often made that open source products are of a lesser quality than their commercial equivalents, on the basis that the freedom for anyone to contribute means that unqualified people will make poor quality contributions. If one accepts that collaboratively developed content such as Wikipedia is of a lesser quality than a commercial equivalent, a question must be asked about the balance between open source and quality. At what point does poor quality negate the benefit of low (or no) cost?

In a debate in The Guardian, Andrew Keen (author of *The Cult of the Amateur*) wrote, “As the traditional media gatekeepers lose their power, the very idea of cultural authority is undermined, meaning that everybody (ie: nobody) can legitimately determine aesthetic standards or truths.” Keen goes on to quip that “the internet is producing the cult of the amateur, a dumbing-down of culture, in which innocence is replacing expertise as the determinant of value” *(Keen & Bell, 2007)*.

The argument that open source means lower quality is not a strong one. The people who get involved in open source projects are typically experts in their fields, and are volunteering their time and efforts for personal, not commercial, reasons.

While some such as Keen are pessimistic about the increasing power of the “amateur” in the information society, others are optimistic.

> *Because they love it, they are amateurs (‘amare’: to love). They don’t get paid (they don’t need the money), but they do what needs to be done (they do need the challenge)... In their eyes, ‘professionals’ are losers who will only work when someone else pays them, thus sacrificing sacred independence. Not them. No one tells them what to do. Never. (Rijken, 2007)*

The open nature of open source projects leaves the work of contributors open to scrutiny and challenge, and the charters, conventions and manifestos of open source projects provide for quality assurance mechanisms. For example, Wikipedia has mechanisms for challenging statements of fact and assertions, for grading the quality of an article, for discussing issues of contention, and for viewing an audit trail of changes and rationales. Perhaps more importantly, open source projects include a continuous improvement mechanism not always found in commercial equivalents.

The willingness of people, readers, to produce as well as consume information, without charging for the content they create, runs contrary to most neo-classical economic theory. Charles Leadbeater, author of *We-Think*, suggests that the motivation to create and share content is a desire for peer recognition *(Leadbeater, 2008)*. He goes further to say that the collaborative Web may be a resuscitation of the pre-industrial *village common*, communally-owned land or resource. This willingness to share is vitalising creativity as a social activity, and is challenging people from all fields to rethink their own roles in education, communication, industry and commerce. The hippies are changing society, in cyberspace.

Your mind is like a parachute, it doesn't work unless it's open.

Frank Zappa

References

- About OER Commons. (n.d.). Open Educational Resources. Organisational, . Retrieved December 2, 2009, from http://www.oercommons.org/about#about-open-educational-resources
- About SourceForge. (n.d.). SourceForge. Commercial, . Retrieved November 30, 2009, from http://sourceforge.net/about
- About W3C. (n.d.). World Wide Web Consortium (W3C). Organisational. Retrieved November 30, 2009, from http://www.w3.org/Consortium/
- Howard, J. R. (1969). The Flowering of the Hippie Movement. The Annals of the American Academy of Political and Social Science, 382(1), 43-55.
- Keen, A., & Bell, E. (2007, August 10). Andrew Keen v Emily Bell | Comment is free | guardian.co.uk. The Guardian. Retrieved August 10, 2009, from http://www.guardian.co.uk/commentisfree/2007/aug/10/andrewkeenvemilybell
- Leadbeater, C. (2008). We-Think (1st ed.). London: Profile Books.
- Netcraft Web Server Survey. (2009, February). Netcraft. Retrieved November 30, 2009, from http://news.netcraft.com/archives/2009/02/18/february_2009_web_server_survey.html
- Reshef, S. (2009). UoPeople Student Handbook - Introduction. University of the People. Retrieved December 2, 2009, from http://www.uopeople.org/LinkClick.aspx?fileticket=svtG9FhsImk%3d&tabid=266
- Rijken, D. (2007). Hippies 2.0 in Museum 2.0. In The User is the Content - 10 Scenarios for the Future (pp. 18-20). Antwerp: C.H.I.P.S. vzw. Retrieved November 30, 2009, from http://www.lulu.com/items/volume_64/1148000/1148772/8/print/MeetingBook3.pdf
- Sieghart, M. A. (2007, May 25). Hey man, we're all kind of hippies now. Far out. Times Online. Newspaper, . Retrieved November 30, 2009, from http://www.timesonline.co.uk/tol/comment/columnists/article1837763.ece
- Swain, H. (2009, November 30). Online university of hope. The Age, 14.
- The Mozilla Manifesto. (n.d.). Mozilla Organisation. Organisational, . Retrieved November 30, 2009, from http://www.mozilla.org/about/manifesto.en.html
- Weber, Steven (2005). The Success of Open Source. Harvard: Harvard University Press.

Context-Agnostic Writing in Modular Documents

Abstract

In the field of technical communication, there is a migration (driven by the need for efficiency) from style-based, document-centric writing approaches to topic-oriented, modular authoring writing techniques *(Houlihan, 2008, p. 4)*. One of the challenges of working within a modular, structured, semantic authoring environment is the imperative to remove context from the writing to enable re-use. In a modular writing environment, a single source or repository of topical information modules are assembled into different publications for delivery in different reading formats. The ability to re-use the same content modules is especially important for organisations managing large documentation suites, such as motor vehicle manufacturers producing documents with sizable proportions of common content.

This paper argues that by adopting context-agnostic writing techniques for topic-based modular documentation, technical writers can improve content re-use and achieve greater efficiency through the technical documentation life cycle without significantly compromising quality. Context-agnostic writing techniques focus on the separation of context from content. This paper will be of interest to people working in the field of technical communication, and also to people engaged in other areas of written communication practice.

Changing Approaches to Documentation

Technical documentation is a broad term which encompasses computer software and hardware user guides, corporate policy and procedure manuals, Help systems, scientific and medical publications, engineering manuals, machinery instructions, reference materials, and many other forms of non-fiction, corporate, product and business discourse. The word "technical" in the term is sometimes misleading, because the term also encompasses some forms of organisational communication. Technical documentation is commonly produced by professional technical writers. For many years, technical writers have followed a document-centric, linear, narrative writing paradigm, treating a manual as a self-contained and isolated work *(Rockley, 2001)*. Before computerisation, technical writers wrote drafts in longhand before sending them for typing, rewriting, editing, reviewing and typesetting. When word processing software tools were adopted by technical writers, document-centric authoring programs such as WordPerfect®, Word and FrameMaker® allowed the same style-based, document-centric paradigm to be used, but with the technical writer taking over the former roles of typists, typesetters, layout artists, and in some cases, printers *(O'Hara, 2001)*.

When topic-based, modular writing techniques became known in the early 1990s, the concept of *single sourcing* became practicable *(Robidoux, 2008, p. 111)*. Single sourcing is "a method for developing re-usable information" *(Ament, 2003, p. xiii)*, where re-usable document modules are assembled to form publications, with different combinations of modules resulting in separate publications. *Modular writing* is a technique that makes single-sourcing possible. Modular, or *topic-based*, writing, is a style of document design and architecture where content is structured into independent small modules (*topics*) which can be assembled into one or many larger texts, such as a books, Web sites and Help systems. The advent of documentation technologies based on *eXtensible Mark-up Language (XML)*, at the start of 21st century, expanded the possibilities for single-sourcing and re-use, and led to the adoption of the philosophy of *separation of content and form* through semantic mark-up *(Sapienza, 2004, p. 400)*. The *Darwin Information Typing Architecture (DITA)* is an XML-based documentation methodology developed explicitly for technical documentation. DITA is primarily a semantic authoring mark-up language, incorporating the ideas of topic-based, modular architecture, semantic mark-up, and XML-based standard information structures. (XML is a set of standards for the categorisation, storage and retrieval of all forms of structured information.)

This paper argues that by adopting context-agnostic writing techniques for topic-based modular documentation, technical writers can improve content re-use and achieve greater efficiency through the technical documentation life cycle without significantly compromising quality. Context-agnostic writing techniques focus on the separation of context from content. This separation is a further extension of the philosophy of the separation of content and form. By minimising context at the topic level, and abstracting it to the document or publication level, the opportunities for topic re-use are maximised.

Modular Writing

Topic-oriented, modular writing is an approach or a technique, rather than a technology in its own right. The modular writing methodology is simply enabled by technologies such as XML. The change in writing technique from document-centric to modular can be a challenge to many technical writers. A 2009 survey of technical writers using DITA observed that the biggest challenge for writers is caused by the topic-oriented paradigm, and that DITA necessitated "making a major change in the document-centric mindset" *(WritePoint, 2010: 12)*.

It is not just the topic-oriented paradigm that presents a challenge to writers. Being able to re-use, re-purpose and re-assemble topics into different publications in different forms requires many other changes in writing *(Williams, 2003)*. The separation of content and form is alien to those writers accustomed to working in format-driven environments such as Microsoft Word, where *WYSIWYG (What You See Is What You Get)* is a fundamental principle. Rich authoring environments in semantic mark-up languages use an alternative *WYSIOO* approach - *What You See Is (just) One Option (O'Keefe, 2006)*.

Separation of Content, Form, and Context

The use of semantic mark-up in DITA, where text elements are marked up based on their meaning, allows the content to be completely separated from its rendition and display to the reader. For example, a term is marked up as a `<term>` and a citation as a `<cite>`, and no information about how those elements will be displayed is stored in the content. Stylistic (display) rules are applied when the DITA content is *transformed* into a reading format, such as HTML or paper. In a DITA workflow, documents are created as collections of modular, re-usable topic files, and mechanisms allow not only the format to be separated from the content, but also the context. The same topic may be a section in the context of one publication, but a sub-section in the context of another. The intermingling of content, format and context in a style-based document workflow essentially eliminates the possibility of re-use. Once a paragraph is styled as having a 13 cm left margin, it cannot be used on paper 12 cm wide. A phrase marked up in italic won't render as italic on a reading device that doesn't support italic. But a citation identified as a citation in a DITA topic can be processed to italic by one transformation process, to bold red by a different transformation process, and to synthesised voice by another transformation process.

One of the more difficult changes to writing technique when moving from linear to modular writing is the removal of as much context as possible from the text. For example, the use of phrases such as “as shown above” and “in the following diagram” will not be valid if the referenced content is not included in all output publications. Well-written topics (and smaller blocks of text) with minimal context can be re-used in many publishing contexts. Writing in such an approach can be referred to as *context-agnostic* or *context-neutral* writing.

The *agnostic* descriptor has been introduced into information technology terminology to make the distinction between systems that are unaware of the platform on which they run, and those that are aware of the platform but adopt a non-preferential, neutral stance with respect to platform (*neutral*). In other words, *agnostic* means no position (unknowable), while *neutral* is a deliberate position. Terms such as *tool-agnostic*, *schema-agnostic*, *implementation-agnostic*, *industry-agnostic*, *technology-agnostic*, *system-agnostic*, and *language-agnostic* illustrate the usage.

In technical and marketing literature, agnostic often has a meaning close to “independent” - for example, “platform agnostic” or “hardware agnostic”. Agnosticism, (n.d.)

In this paper, I prefer the term *context-agnostic* because it describes the scenario where the author is unaware of the contexts in which a content module will be used.

Context-Agnosticism: Utility, not Poetry

Petelin & Durham (1992, pp. 20-22) describe three types of context that need to be considered by a writer: “the context of culture, the context of organisations, and the brief context of particular situations”. These three types of context need to be separated from content in the writing workflow for the content to be suitable for re-use in multiple contexts. Separating cultural context might mean avoiding culture-specific terms, such as “bonnet” and “hood”.

Separating organisation context might mean avoiding product names. Separating situational context might mean standardising writing style.

O'Neill (2002) identified the following types of contextual information that needs to be minimised in order to produce truly global, modular information:

- product and company names
- organisation-specific terminology
- different ways of writing
- inconsistent standards application in authoring tools.

Writing content that is abstracted from its delivery, presentation, structure and context may result in a sacrifice of the look-and-feel of the end product. However, compromise in technical documentation, in Web design, and in most other fields of communication is normal *(Hackos, 1994, pp. 214-215)*. Content-agnostic writing is a typical information architecture decision which does sacrifice some aesthetics for re-usability and efficiency. As put succinctly by Ellen McDaniel in a conference paper on structured authoring in XML, context-neutral authoring is about “utility, not poetry” *(McDaniel, 2005, p. 9)*.

The view that writing without knowledge of context is a valid approach is not universally accepted. Many of the arguments for and against writing for single-source publishing were summarised by *Robidoux (2008, pp. 110-111)* in a paper discussing the challenges of developing a curriculum for teaching single-sourcing to technical writers.

Although there is evidence that single-sourcing and content re-use reduces writing costs *(Houlihan, 2010)*, there is not yet a large body of evidence to suggest that context-agnostic writing leads to equivalent (or superior) quality of the reading experience itself. However, there are some indications that the benefits of modular writing, such as the ability to customise the modules delivered (online) to suit the needs of an individual reader, do lead to a better reading experience. For example, *Hackos & Hedlund (2001, p. 2)* point to a case study where, from the perspective of a reader (or information “user”), information can be “delivered in a different way that makes more sense and is more usable in the context of its use”.

My experience with single-sourcing has led me to the belief that choosing a context-agnostic writing approach is a sound, pragmatic decision. Context-agnostic writing does not mean the removal of context, but the separation of context from content.

Techniques for Reducing Context

This paper suggests that in order to write in a context-agnostic way, the following techniques can be used:

- removing or minimising context phrasing (eg, “Have your vehicle inspected...”)
- avoiding terms specific to organisations, industries, geography and culture

- using filtering to selectively remove conditional content from the deliverable document
- removing branding and other context from graphics
- removing sequence context and enumeration from topics, and automatically applying sequence in the publishing process
- externalising other context (by storing context in document maps rather than topics, and using devices such as relationship tables, variables and indirection).

Simplistic Removal of Context Phrasing

Some contextual phrasing, such as product and company names, can be removed from text without significant compromise to its readability.

For example, a sentence in the User Guide for (the fictitious) ProductA of “Use ProductA to create a project file to manage all the files in your system” can be re-written as “Use this product to create a project file to manage all the files in your system”. The re-written sentence is then able to be re-used in the User Guide for the similar ProductB.

Likewise, a sentence in a car owner's manual of “your Subaru Impreza is fitted with a supplemental restraint system” can be re-written as “your car is fitted with a supplemental restraint system”. This change removes the context binding the sentence to Subaru Impreza cars only, so that the sentence may be re-used elsewhere, such as in the owner's manual for a Saab 9_2 (a re-badged Subaru Impreza WRX). However, there is more context that can be removed from the same sentence. The re-written sentence mentions “car”, which restricts the use of the sentence to documents concerning cars. Using “vehicle” instead would broaden the potential use of the sentence to documents concerning cars, trucks and boats.

Writers working in a modular writing environment do not need to know where a topic they are writing is intended to be used, or may one day be used. In fact, it may benefit the writer not to know what a topic is intended for, making it easier to write without a context. It is also good practice to be generic as possible. In the example above, a writer knowing that Subaru doesn't currently make boats or trucks may be tempted to use “car” rather than the more agnostic “vehicle”. However, using the narrower “car” context may close off some future re-use opportunities.

In the following example, reference is made to a Subaru dealer; this context limits the application of this paragraph to Subaru manuals (assuming a *variable* has not been used). A context-neutral alternative sentence might read: “Have your vehicle inspected at your local dealer.”

If your vehicle has sustained impact, this may affect the proper function of the Subaru advanced frontal airbag system. Have your vehicle inspected at your SUBARU dealer.

In some cases, the overuse of generic terms may cause misinterpretation of the text. When the context is removed in the example above, a reader may interpret the phrase “your local dealer” as meaning the nearest motor vehicle trader, rather than the intended meaning of the nearest Subaru service centre. Care has to be taken to find the right balance so that the removal of context does not result in the removal of meaning.

DITA provides other mechanisms to allow better management of terminology such as product and company names through features such as variables. This permits document-specific terms to be identified in the topics, and then substituted with a contextually applicable alternative when the document is published to a deliverable format.

Supra-Organisation Specific Terminology

Although not mentioned by *O'Neill (2002)* in the list of types of contextual information, supra-organisation specific terminology is also a common type of contextual information that would need to be removed to maximise re-use potential. For example, referring to tax authorities using the US term “IRA” binds the use of the text to a United States context (US Government Internal Revenue Agency).

The following text (from a *Subaru Impreza MY06 Manual*) is largely context-agnostic.

> *Your vehicle is equipped with a seatbelt warning device at the driver's seat, as required by current safety standards. There is a seatbelt warning light in the combination meter.*

The authors have correctly used “your vehicle”, instead of “your Impreza” or “your Subaru”. Rather than quoting a specific law or safety regulation, the authors have chosen the neutral phrase “current safety standards”. Using more specific (and contextual) language such as “as required by Australian Design Rules” would require the text to be re-written and replaced for every market other than Australia. It might also lead to unnecessary research, such as determining what safety standards are used in Papua New Guinea for the manuals delivered to that country. The reader of the manual, in this case the driver of a Subaru Impreza, just needs to know that the car is fitted with a seatbelt warning device. It could be argued that the phrase “as required by current safety standards” could be removed entirely.

Compare that context-neutral approach with the following paragraph, which can only be understood in the context of a US-based reader:

> *Your vehicle is equipped with a Subaru advanced frontal airbag system that complies with the new advanced frontal airbag requirements in the amended Federal Motor Vehicle Safety Standard (FMVSS) No. 208.*

To a driver in Malaysia, a reference to a (presumably) US Federal standard is irrelevant at best, and confusing at worst. Further, the use of the word “new” restricts the use of the sentence to a time frame; in one year's time, FMVSS 208 will surely no longer be new!

Use of terms specific to a culture or a geographical audience is another example of context-heavy writing. The phrase “trunk lid” may be understood by North American English-speaking audiences, but is not readily understood by many other English-speaking audiences, who know the component as a “boot lid”. Using a variable for the term, using an alternative neutral term (if possible), or including both terms are some methods to dilute the context.

Filtering Out Context

Conditionalising content helps avoid another obstacle to re-use. To avoid having complicated text blocks listing conditions when the content should be used, conditions can be applied to text, and filtering used to selectively remove text from outputs for a particular market, audience or publication.

In the following example, the measurements have been provided in two different units, so that the text can be understood by readers from the US (and other countries using Imperial measurement units) and readers who use the metric system. The text has not been written in the context of the audience using the metric system.

The extender adds approximately 8 inches (200 mm) of length and it can be used for either the driver or front passenger seating position.

Although it might appear that the better outcome might be one where the source contains both measurements, that is not necessarily the case. DITA and other semantic XML approaches incorporate conditional filters which can be used to exclude the non-preferred unit from the output for a particular market. However, even in a country using the metric system there will be drivers who better understand the Imperial system.

Filtering can also enhance readability by removing clutter. In the example below, the comprehensibility of the text is impeded by the condition that the sentence only relates to cars fitted with an immobiliser. Using conditional filtering techniques, this sentence could be excluded from manuals for car models without immobilisers, and the parenthetical clause deleted.

The security indicator light will continue to flash once every three seconds indicating that the system is in the valet mode (only vehicle with an immobiliser).

Removing Context in Graphics

Illustrations, photographs, images and other graphic devices within documents are themselves innately modular, in that they are stored as separate files, and can be used in different contexts in different documents. However, the content of graphics can bind them to context. For example, a photograph with a figure title superimposed on the image can only be re-used if the figure title is relevant in the different re-use context. A photograph with English language call-outs can only be used in an English language document.

Care taken in the creation of images can maximise their re-use potential. The omission of branding from graphics will permit greater re-use. A photo identifying the radiator grille of a car is not significantly enhanced by the inclusion of the car maker's bonnet badge, but including the badge in the photo will render it unsuitable for illustrating any document other than one concerning that car brand.

Removing Sequence

Use of terms that denote sequence is another form of context that must be removed for a text to be context-agnostic. Adjectives such as “previous”, “next”, “earlier”, “aforementioned” and “later”, as in “more information on the product limitations are discussed in later chapters”, should therefore be avoided. Including such restrictive references introduces the risk that the blocks of content won't be in the sequence in all output publications, and that the adjective becomes incorrect and misleading. (It is also possible that the referenced blocks might not be included in all output publications.)

While adjectives such as “following” or “preceding” do apply a context, that fact will not usually present a problem if the referred element is within the same information chunk. (A *chunk* is a block of text describing one idea or step.) For example, if the lead-in sentence will only be re-used in conjunction with the larger block in which it is structured, then the context is confined to *intra-chunk*. Context-agnostic writing only needs to aim to minimise *extra-chunk* context, as this type of context is limiting to re-use.

An example of *intra-chunk* context is:

> *Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.*

In a semantic mark-up language such as DITA, the mark-up itself can be used to bind the lead-in sentence to its antecedents, as in the example:

```
<p>Operation is subject to the following two conditions:
 <ol>
  <li>This device may not cause harmful interference, and</li>
  <li>this device must accept any interference received, including
  interference that may cause undesired operation.</li>
 </ol>
</p>
```

The paragraph element (`<p>`) contains both the word “following” and the items that follow. Whenever the paragraph is re-used, the items will be included in the re-use.

Manually-applied numbers are also problematic, as they lock in a sequence. If enumerations such as chapter, figure and paragraph numbers are required in the output document, they should be automatically generated through the publishing process. (The publishing process converts the topics from their authoring format to a deliverable document in a reading format such as paper or Web.)

While it may initially appear that having no numbered identifiers in the authoring format would make cross-referencing (such as “listed in Table 3 - Codes”) impossible, in fact cross-referencing is a trivial authoring task. DITA incorporates a comprehensive cross-referencing mechanism using metadata as the linking key. This allows references to numbered objects in the output to also be automatically derived during the publishing process.

Cross-references

Although cross-references may be technically easy for a writer to devise, cross-references are heavily burdened with context.

Cross-references to other parts of a publication are context-specific, because to make sense they require a pre-condition that the referenced target (in the example below, the “Activating...” procedure) is included in the output publication.

To exit valet mode, change the setting of your vehicle’s alarm system for activation mode. (Refer to “Activating and deactivating the alarm system” in this section.)

Cross-references are, however, often critical to the navigation logic and the understandability of the text. Rather than remove this context completely, the best approach is to move the context from the topic to the *document map*.

The document map (in DITA, it is known as the *ditamap*) is a *manifest* listing the topic modules that are to be included in a deliverable document, and the hierarchy and sequence in which they will appear. A ditamap is therefore specific to a publication, while topics may be re-used in many publications.

Specifically to allow the shifting of link context from context-agnostic topic to context-specific map, DITA includes a *relationship table*, defined in the ditamap. The relationship table defines the linking relationship between topics in the publication. Because the ditamap defines the collection of topics to be delivered as a publication, this is a more logical place for links between topics in the collection to be defined. When the ditamap is processed to create the deliverable document, the relationships in the relationship table are translated into cross-references that are added at the end of the topic in the output.

There is some evidence that for Web content, cross-reference links placed at the bottom of the page, that is, outside the paragraph text, may be more usable than inline hyperlinks *(McGovern, 2009)* - an added benefit derived from the use of relationship tables.

Externalising Other Context

XML-based modular writing systems offer other methods to *externalise* context, or move the context from topics to the map manifest. One such method is the *keyref* feature offered in DITA.

The concept of keyref is that references to other resources are made with one level of abstraction; the link in the topic refers to a key, and a matching key in the ditamap refers to the resource. This concept is known as *indirection*.

For example, rather than a cross-reference in a paragraph directly referring to another topic, such as in `<xref href="wrx_specs.dita">`, the cross-reference would refer to a key, as in `<xref keyref="model_specs">`. When a direct reference (in the example, to the Subaru WRX model specifications in `wrx_specs.dita`) is used, the paragraph can only be re-used in a WRX publication. However, when an indirect reference is used, the actual target is defined in the ditamap file which specifies the publication. The code in the WRX ditamap might be `<topicref keys="model_specs" href="wrx_specs.dita">`, while in the ditamap used for the Saab 9_2, the code might be `<topicref keys="model_specs" href="9_2_specs.dita">`. Using this indirection technique, the cross-reference in the one topic can refer to entirely different target topics depending on the ditamap in which the topic is used. Indirection can be used for any addressable document component: links, cross-references, glossary definitions, images, and content snippets.

Indirection effectively moves context from the content topic to the document specification (map) level.

Conclusions Drawn About Context-Agnostic Writing

The change in the technical communication field from document-centric writing to topic-based writing, made practical by the development of XML-based documentation architectures such as DITA, will necessitate a change in writing approach to separate content, form and context. The modular DITA architecture in particular provides for the creation of independent content *topics* that are assembled into deliverable publications through document *maps*. While semantic mark-up allows content to be separated from form, features in the document map allow some types of context to be moved from the topic level to the map level.

Adopting context-agnostic writing techniques to topic-based modular documentation will result in greater content re-use opportunities, leading to improved efficiency through the technical documentation life cycle. Those writing techniques include:

- removing or minimising context phrasing
- avoiding supra-organisational specific terms
- using filtering to selectively remove conditional content
- remove branding and other context from graphics
- removing sequence context from topics, and automatically applying sequence in the publishing process
- externalising cross-references through relationship tables, variables and indirection.

Context-agnostic writing does not mean the removal of context, but the separation of context from content and form. While it is not known whether context-agnostic writing will affect the quality of communication by leading to an inferior, superior or equivalent experience for the reader, the concept of re-introducing context customised to the requirements of the individual reader, at the point of document delivery, holds much promise.

The prime benefit of writing modular content suitable for publication in different forms for different purposes in different contexts is efficiency: being able to create more in less time. This benefit is not only applicable to the field of technical communication, but also in other areas of written communication practice where the drive for greater efficiency suggests a re-evaluation of writing processes.

References

- Ament, K. (2003). Single sourcing: Building modular documentation (1st ed.). New York: William Andrew.
- Hackos, J. (1994). Managing your documentation projects. Wiley Technical Communication Library (1st ed.). New York: John Wiley and Sons.
- Hackos, J., & Hedlund, T. (2001). Making a business case for single sourcing. Denver, Colorado: Center for Information-Development Management. Retrieved May 15, 2010, from http://www.comtech-serv.com/pdfs/Business_Case_for_Single_Sourcing.pdf
- Houlihan, D. (2008). The technical communicator's transformation: Publishing on time and on quality. Boston: Aberdeen Group.
- Houlihan, D. (2010). Benchmarking content reuse in technical communication. Boston: Aberdeen Group.

- McDaniel, E. (2005). Consider the source: Structured authoring for XML-based documentation. Presented at the UNC College and University System Exchange (CAUSE) Conference, Raleigh, North Carolina: University of North Carolina. Retrieved February 23, 2010, from http://www.unc.edu/cause05/presentations/mcdaniel/mcdaniel.pdf
- McGovern, G. (2009, November 2). Why web links are calls to action. New Thinking. Retrieved February 23, 2010, from http://www.gerrymcgovern.com/nt/2009/nt-2009-11-02-Web-links-action.htm
- O'Keefe, S. (2006, April 12). Wednesday at WritersUA: Structured authoring: taking the plunge. Scriptorium. Retrieved May 15, 2010, from http://www.scriptorium.com/blog/2006/04/wednesday-at-writersua-structured-authoring-taking-the-plunge.html
- O'Neill, J. (2002). A global style guide: Working together around the world. STC 2002 Conference Proceedings. Retrieved August 26, 2009, from http://www.stc.org/confproceed/2002/PDFs/STC49-00024.pdf
- Petelin, R., & Durham, M. (1992). The professional writing guide (1st ed.). Warriewood, NSW: Business and Professional Publishing.
- Robidoux, C. (2008). Rhetorically structured content: Developing a collaborative single-sourcing curriculum. Technical Communication Quarterly, 17(1), 110-135.
- Rockley, A. (2001). The impact of single sourcing and technology. Technical Communication: Journal of the Society for Technical Communication, 48(2), 189-199.
- Sapienza, F. (2004). Usability, structured content, and single sourcing with XML. Technical Communication: Journal of the Society for Technical Communication, 51(3), 399-408.
- Williams, J. D. (2003). The implications of single sourcing for technical communicators. Technical Communication: Journal of the Society for Technical Communication, 50(3), 321-327.
- WritePoint. (2010). DITA challenges survey results (November 2009). Commercial. Retrieved January 18, 2010, from http://www.writepoint.com/site/downloads/WritePoint_DITASurveyChallenges_Results.pdf

Editing Marks for XML Mark-up

Editing Marks for XML Mark-up - Abstract

Editing (or proof-reading) marks have been used as a shorthand in copy-editing and proof-reading tasks since the 15th Century *(André, 1998)*, and in a similar form to the current International Standard since the 17th Century *(Simpson, 1935)*. The marks are designed for highlighting spelling, typographical, composition and grammatical errors, where the document being reviewed is in the same presentational form in which it will be published.

When documents are created in XML-based semantic mark-up languages such as DITA and DocBook, the content is separated from the presentational form. Copy-editors and proof-readers work not with the final form of the document, but with a *form-agnostic* representation of the document marked up with semantic identifiers. The final form is produced in an automated publishing process, where semantic mark-up is mapped to presentational elements. When working with a form-agnostic document, the editing and proof-reading tasks focus on language issues (spelling, punctuation, grammar, and wording errors) and on semantic mark-up issues.

Editing in XML-based semantic mark-up languages calls for a different approach to editing marks. Addressing this problem is the main motivation for this paper. This paper suggests an alternative set of editing marks for semantic authoring, incorporating those conventional editing marks that remain appropriate, and discarding those marks that are not.

A further impediment to editing in XML-based semantic languages is the challenge of displaying the XML mark-up in paper-based renditions of the document without significantly degrading the readability. This paper identifies two technical approaches to address this challenge.

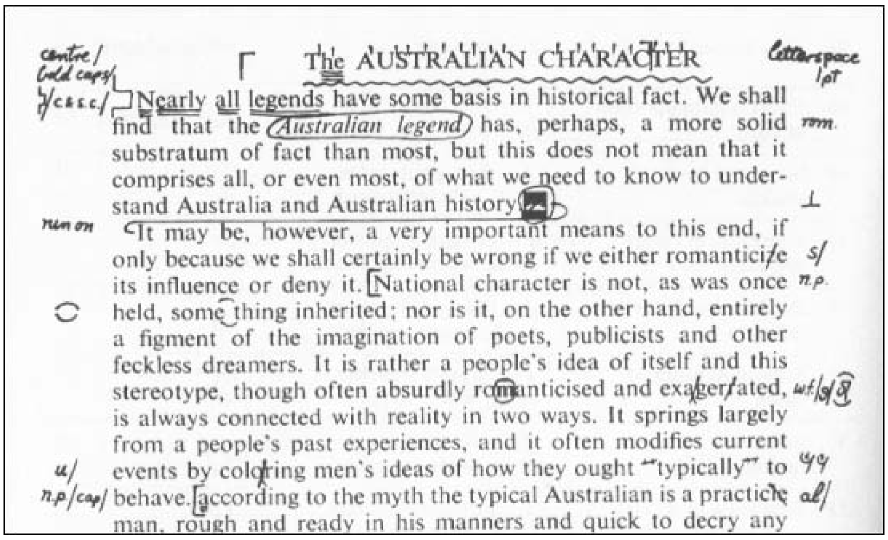
Existing Use of Editing Marks

Editing is an often tedious process in which a document is scrutinised to improve the quality of two fundamental components: content and form.

Editing marks (also known as *proof-reading marks*) are a hand-written mark-up system used by editors of all many of documents. They are commonly used in technical publications. An understanding of the marks is widely known. Most generic style manuals such as the Australian *Style Manual: for authors, editors and printers (Snooks and Co., 2002)* provide a guide to the use of editing marks. (Refer to *Figure 54: Example of Proofing Marks in Practice* (see page 232)).

Some common editing marks indicate changes such as insert, delete, transpose, set in capitals, italicise, bold, spelling, close space, align vertically, move left, and new paragraph. Many of these marks relate to presentational style, and few can be used to indicate errors or required changes in semantic mark-up. (Refer to *Figure 55: Sample proof mark guidelines from Manual of Style (Snooks, 2002)* (see page 234)).

This is a scanned portion of page 148 of the AGPS Manual of Style, 3rd Edition



centre/ bold caps/ THE AUSTRALIAN CHARACTER letterspace 1pt

l/c & s.c./ Nearly all legends have some basis in historical fact. We shall find that the *Australian legend* has, perhaps, a more solid rom. substratum of fact than most, but this does not mean that it comprises all, or even most, of what we need to know to understand Australia and Australian history. ⊥

run on It may be, however, a very important means to this end, if only because we shall certainly be wrong if we either romanticize s/ its influence or deny it. National character is not, as was once n.p. held, some thing inherited; nor is it, on the other hand, entirely a figment of the imagination of poets, publicists and other feckless dreamers. It is rather a people's idea of itself and this stereotype, though often absurdly romanticised and exagerated, w.f./g/ is always connected with reality in two ways. It springs largely from a people's past experiences, and it often modifies current u/ events by colqring men's ideas of how they ought "typically" to behave. n.p./cap/ according to the myth the typical Australian is a practice al/ man, rough and ready in his manners and quick to decry any

Figure 54: Example of Proofing Marks in Practice

Editing marks have been in use since at least the 15th Century, according to *André (1998)* in *Petite histoire des signes de correction typographique*. In the English-speaking world, according to *Simpson (1935)*, a system of editing or proofing marks was being used in printing shops in the 17th Century. As the printing industry grew, editing marks became more widespread and standardised.

André and Richy (1999) explained the purpose of editing (or proof-reading) marks as "to precisely localize where to modify the text, and to precisely indicate where modifications had to be done".

According to the *Freelance Editorial Association (n.d.)*, there are four different categories of editing:

- developmental (editing during the content creation process)
- substantive (improvements after the writer has completed the manuscript)
- copy (correcting errors and enforcing style manual rules)
- proofreading (checking the final version for typographical and layout errors).

The *Chicago Manual of Style (1982)* divides the editorial process into *mechanical editing* and *substantive editing*. Mechanical editing involves such matters as capitalisation, spelling, agreement of subjects and verbs, punctuation, and use of numbers. Substantive editing involves re-writing, re-organising, and reworking of the writing style to conform with the guidelines in a style manual.

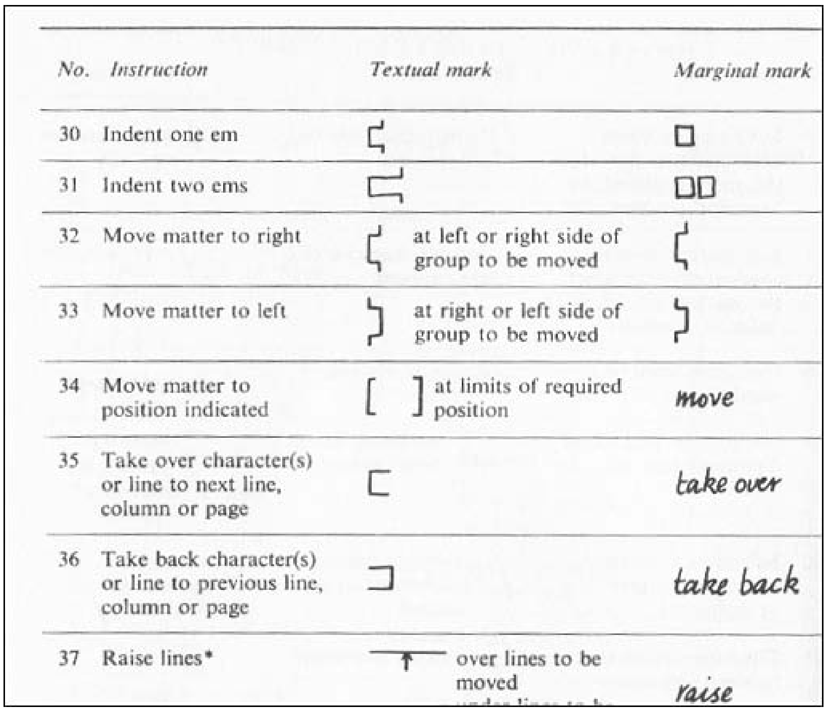
Regardless of the categorisation, editing marks are used in each of the stages of editing.

In this paper, “editing” is used to describe the whole editing process, and “proof-reading” is used to describe the *mechanical editing* checks of the final version of a document prior to publication.

The eradication of grammatical and spelling errors is important to reinforce the quality, credibility and integrity of the document; “if it is printed with spelling mistakes and editorial and design inconsistencies, its authority will - to some extent - be undermined” *(Snooks, 2002, p. 276)*. It is therefore imperative that adequate and efficient proof-reading measures are taken for all documents.

Although computerisation has improved some of the processes, editing remains a labour-intensive endeavour. As explained in *Snooks (2002)*:

> *With the introduction of computer technology, the proofreading process is sometimes reduced to little more than running a spellcheck. This is not proofreading, and it is far from adequate. ... Proofreaders must be able to find and correct errors, understand copy editors' markings and indicate corrections in the proper way.*



No.	Instruction	Textual mark	Marginal mark
30	Indent one em		□
31	Indent two ems		□□
32	Move matter to right	at left or right side of group to be moved	
33	Move matter to left	at right or left side of group to be moved	
34	Move matter to position indicated	[ ] at limits of required position	move
35	Take over character(s) or line to next line, column or page	[	take over
36	Take back character(s) or line to previous line, column or page	]	take back
37	Raise lines*	↑ over lines to be moved	raise

Figure 55: Sample proof mark guidelines from Manual of Style (Snooks, 2002)

Editing marks therefore remain an important tool in the editing process. An international standard for editing marks, *ISO 5776: Graphic technology - Symbols for text correction*, was released in 1983 to formalise long-standing industry conventions. However, while the standard editing marks are useful for correcting traditional documents, they are not effective for correcting XML-based, semantic documents such as those created in DITA.

The Need for New Editing Marks for XML Mark-up

The use of semantic mark-up in DITA, where text elements are marked up based on their meaning, allows the content to be completely separated from its rendition and display to the reader. For example, a term is marked up as a `<term>` and a citation as a `<cite>`, and no information about how those elements will be displayed is stored in the content. Stylistic (display) rules are applied when the DITA content is *transformed* into a reading format, such as HTML or paper. In a DITA workflow, documents are created as collections of modular, re-usable topic files, and mechanisms allow not only the format to be separated from the content, but also the context. The same topic may be a section in the context of one publication, but a sub-section in the context of another. The intermingling of content, format and context in a style-based document workflow essentially eliminates the possibility of re-use. Once a paragraph is styled as having a 13 cm left margin, it cannot be used on paper 12 cm wide. A phrase marked up in italic won't render as italic on a reading device that doesn't support italic. But a citation identified as a citation in a DITA topic can be processed to italic by one transformation process, to bold red by a different transformation process, and to synthesised voice by another transformation process.

Traditional style manuals for editors and publishers have a significant emphasis on presentational aspects of documents. *Self (2009, p.258)* found that up to 70% of the content of popular style manuals, such as *The Chicago Manual of Style (1982)*, was devoted to issues related to presentational style, or form. In semantic mark-up environments, the author, editor and proof-reader have no association with form, as form is separated from content to become a discrete, separate, automated function within the document production process.

Documents written in a semantic mark-up format such as DITA cannot be proofed in the same way as documents written in a presentational format. The semantic mark-up doesn't show the eventual presented form of the document; when printed or displayed for proof-reading, it shows a possible presentational rendition. This is sometimes known as *WYSIOO*, or *What You See Is (just) One Option (O'Keefe, 2006)*. The author, editor and the proof-reader may never see the presentational form of the document until after publication. The different publishing workflow in a semantic mark-up environment is required, and a different approach to editing, proof-reading and editing marks is required. *Gardiner (2010)* suggested that one of the reasons why the promised efficiencies of XML-based publishing hadn't been achieved was because “it is obvious that the [editing] workflow continues parallel the conventional stages of production”.

Studies as far back as 1987 have demonstrated that when proof-reading on-screen instead of on paper, “more proof-reading errors were missed, fewer pages were read, and there was a greater accumulation of fatigue during the reading session” *(Wilkinson, 1987)*. Wilkinson *(ibid.)* concluded that “material be printed for proof-reading”. *O'Hara & Sellen (1997)* found that one of the “major advantages that paper offers in supporting annotation while reading”. Editing and proof-reading XML documents should not preclude the use of paper in the process, so if editing or proof-reading on paper is to continue as part of the publishing process, a system of editing marks has to be maintained. Research into paper-less editing and proof-reading, such as that by *André and Richy (1999)*, tend to rely on electronic pens, tablets and paper. When such paper-less approaches reach maturity, the need for an *XML-friendly* editing mark system (and a method for exposing underlying XML mark-up) will remain.

Personal Observations - Editing an XML Document

One of the authors of this paper has first-hand experience editing and proof-reading documents in DITA XML format, and made a number of observations about the limitations of conventional editing marks for this process.

The author's personal observations were as follows.

> *When editing a document written collaboratively in DITA (the DITA Help Technologies Guide), my first step was to print a copy of the generated PDF version of the DITA document. I find it more convenient to edit on paper. I found, however, that the standard editing marks were not appropriate for this task.*
>
> *The problems that arose are typified by the following examples.*
>
> - *If a phrase was rendered in italics when I was expecting bold, how would I know if that was a mark-up error (such as the author using `synph` when `uicontrol` was semantically correct) or a processing decision? How do I indicate that I want to check this or stipulate this? My role was to ensure the DITA document was accurate, not to adjudicate on the presentational style.*
> - *In some cases, a semantic mark-up did not result in a different presentational style from standard text. How could I check that the element was correctly marked-up if there was no visible indication?*
> - *Topic metadata, and even some topic content, may not be rendered in the printed version. How can I check whether this important information is correctly included?*
> - *If a quotation appeared on the page correctly encapsulated with quotation marks, did this mean that the author had correctly used the `q` element, or had incorrectly hard-coded the quotation marks?*
> - *If I could determine that a chunk of text was incorrectly marked-up, how could I indicate the error with editing marks?*

In summary, the author found that editing marks were inadequate because:

- they were primarily focussed on presentational appearance
- they did not allow for revising semantic mark-up

For editing of paper documents authored in DITA to be practical, the following changes to editing practice are needed.

- A new output format specifically used for editing, where all semantic mark-up is identifiable in a supra-organisational standard way, must be devised.
- A new standard editing mark-up scheme for printed renditions of semantically marked-up documents must be developed.

Editing Marks Proposal

> *A new set of XML-aware editing marks to replace or augment the current style-based editing marks will permit handwritten instructions to be communicated between authors,*

> *reviewers and editors, if used in conjunction with a technique to expose the semantic mark-up in the review document.*

This paper proposes the following editing marks for reviewing XML documents.

Table 10: New XML-specific editing marks

Intention	**In-text mark-up**	**In-margin mark-up**
Apply the element in the margin to the highlighted text (or check that it has been applied).	( ) or circle	<cite> or <cite>?
Apply the attribute in the margin to the highlighted element (or check that it has been applied).	⫯ or ( ) or circle	@product=Lite
Insert a link at the marked insertion point to the topic (or other element) name nominated in the margin.	⋏	↷*[topic name]*
Remove the XML mark-up	( ) or circle	<₰>
Move the opening XML tag to the position marked by the arrow	↑_▲	<>
Move the closing XML tag to the position marked by the arrow	▼_↑	<>

Table 11: Traditional editing marks to continue to use with same meaning

Intention	**Mark-up**
Insert in text the content indicated in the margin	⋏
Delete	₰
Delete and close up	₰

Intention	**Mark-up**
Leave as printed	*stet*
Change to capital letters	*caps*
Change to capital for initial letter and small caps for remainder	*c. & s.c.*
Change to lower case	*l.c.*
Insert superior (superscript) character(s)	˥
Insert inferior (subscript) character(s)	˩
Use ligature or dipthong	⊂⊃
Close up (delete space)	⊂⊃
Insert space	#
Transpose characters or words	⊓⊔*trs*
Begin new paragraph	[*n.p.*
No fresh paragraph; run on from previous paragraph	⟲*run on*
Spell out abbreviation or figure in full	*spell out*
Insert omitted portion of text	*out - see copy*
Substitute or insert a comma	,/
Substitute or insert a semi-colon	;/

Intention	Mark-up
Substitute or insert a full stop	⊙/
Substitute or insert a colon	$\textcircled{:}$/
Substitute or insert a question mark	?/
Substitute or insert an exclamation mark	!/
Surround with parentheses	(/)/
Surround with brackets	[/]/
Insert an apostrophe	$\overset{,}{\curlyvee}$
Insert an ellipsis	.../
Insert a diagonal stroke	⊘
Refer issue... doubtful accuracy	$\textcircled{?}$
Correction is concluded (separator for in-margin mark-up)	/

Table 12: Traditional editing marks not to use

Intention	Mark-up
Change highlighted text to italics	*ital*
Insert more white space	
Change highlighted text to small capitals	*s.c.*
Change highlighted text to bold	***bold***
Change highlighted text to roman (non-italic) type	*rom.*

Intention	Mark-up
Wrong font	*w.f.*
Invert type	୭
Change damaged character	x
Underline	*underline*
Insert space between lines or paragraphs	>\#
Reduce space between lines or paragraphs	( )
Make space appear equal between words	*eq.* \#
Less space between words	*less* \#
Insert letter space	*letter* \#
Move to centre	⌈ ⌉*centre*
Indent text	⊏
Move text block to the right	⊏
Move text block to the left	⊐
Move text to a different position	[ ]*move*
Take (wrap) text over to the next line	[*take over*
Take text back to the previous line	]*take back*
Raise lines	*raise*
Lower lines	*lower*
Correct the vertical alignment	‖
Straighten lines	═

Intention	**Mark-up**
Push space down	⊥
Insert a rule (en, em or 2 em)	⊢⊣
Insert a single quotation mark	$\overset{‘}{\vee}\overset{’}{\vee}$
Insert a double quotation mark	$\overset{“}{\vee}\overset{”}{\vee}$
Insert a leader	⊙

Editing Aids to Expose XML Mark-up

The correctness of XML mark-up can only be checked within a paper-based proof-reading process if the XML mark-up is somehow exposed on the printed page. One of the authors of this paper has experimented with a number of technical solutions based on DHTML, CSS and XSL-T to allow the display of XML mark-up in paper-based renditions of the document, without significantly degrading the readability. Two such technical solutions can be best explained through brief case studies.

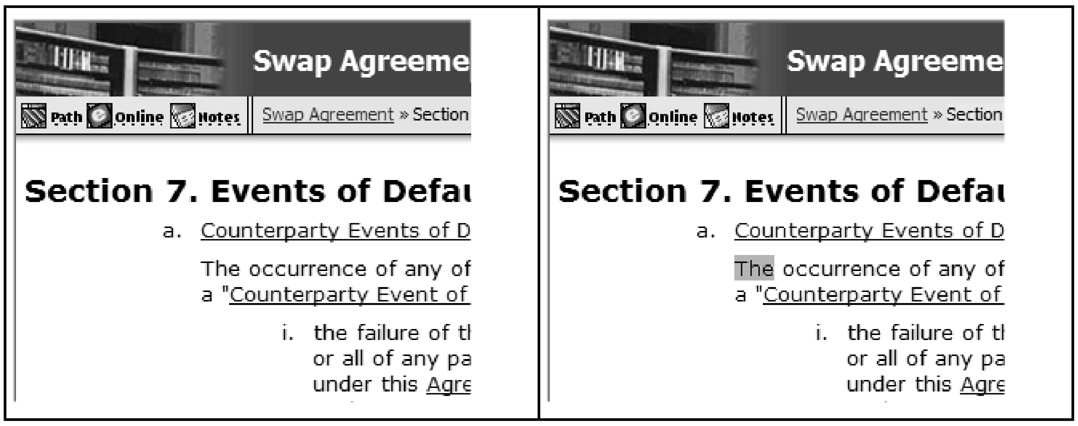
Case Study: Editing-Friendly Web Content through DHTML

The structured nature of XML documents opens up many opportunities for semi-automating some editing and proof-reading tasks, and providing editing aids.

A company producing electronic versions of complex legal contract documents required a quality assurance (QA) process that would effectively eliminate the risk of errors being introduced into documents during the conversion from paper-based (Microsoft Word or Adobe PDF format) to electronic (Microsoft HTML Help, or *CHM*, format). The accuracy of the electronic documents was fundamental to the business.

There was no alternative to a manual proof-reading and QA process to ensure that the automatic conversion steps had been completed correctly. This manual process was tedious and repetitive, and thus prone to human error. For example, the proof-reader were required to check that every use of a defined term (such as “Project Completion Date”) in the document was correctly linked to the definition of that term in the Definitions section. In some cases, similar terms could easily be linked to the incorrect definition, such as “Commonwealth (Bank)” being linked to “Commonwealth (Government)”.

To make the QA process easier for the proof-readers, one of the authors of this paper developed a DHTML (Dynamic HTML) feature which would highlight the elements in the document of interest to proof-readers. For example, unlinked words starting with a capital letter (a possible indicator or a defined term) might be highlighted so that the proof-reader could check whether the word was a term or an undefined proper noun or first word in a sentence (as shown in *Figure 56: Screen captures showing different rendering views* (see page 242)).



Section 7. Events of Defau

a. Counterparty Events of D

The occurrence of any of
a "Counterparty Event of

i. the failure of tl
or all of any pa
under this Agre

Section 7. Events of Defau

a. Counterparty Events of D

The occurrence of any of
a "Counterparty Event of

i. the failure of tl
or all of any pa
under this Agre

Figure 56: Screen captures showing different rendering views

Two different renderings of the CHM document was required: the normal view, and the QA view. To minimise the risk of the QA version being accidentally distributed to end users of the document, and to obviate the need to build two different versions of the document, the DHTML technique used employed an external JavaScript file named `qa.js`. When an HTML topic in the CHM file was displayed, a JavaScript function embedded in the topic was launched to attempt to load the `qa.js` file. If the file was not found, the topic would be displayed normally. If the file was found, functions in the `qa.js` would execute to re-format the topic to highlight the point of interest. Different `qa.js` files could be used to highlight different points of interest for the different QA "sweeps". As end users would never have the `qa.js` file, they would never see the QA view.

The JavaScript code to load the `qa.js` file was:

```
<script>
if(location.href.search(/QA/)!=-1)  {
 document.write('<scr' + 'ipt order="6" language="JavaScript1.2"
src="' + chmfile_path + 'qa.js"><\/scr' + 'ipt>')
 }
</script>
```

While this technique was very effective for on-screen proof-reading, it was not used for paper-based editing. However, it would be possible to print the topics to provide a paper equivalent provided the formatting was suitable for the (usually monochrome) paper medium.

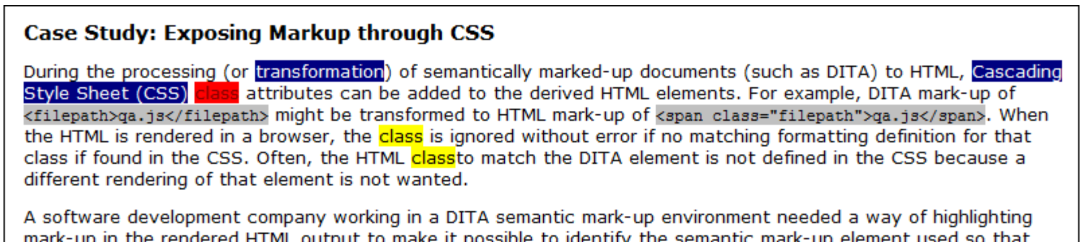
Case Study: Exposing Mark-up through CSS

During the processing (or *transformation*) of semantically marked-up documents (such as DITA) to HTML, *Cascading Style Sheet (CSS)* `class` attributes can be added to the derived HTML elements. For example, DITA mark-up of `<filepath>abc.exe</filepath>` might be transformed to HTML mark-up of `<span class="filepath">abc.exe</span>`. When the HTML is rendered in a browser, the class is ignored without error if no matching formatting definition for that class if found in the CSS. Often, the HTML class to match the DITA element is not defined in the CSS because a different rendering of that element is not wanted.

A software development company working in a DITA semantic mark-up environment needed a way of highlighting mark-up in the rendered HTML output to make it possible to identify the semantic mark-up element used so that consistency of mark-up could be checked as part of the editing process. One of the authors of this paper worked in conjunction with the company to create a CSS file which would highlight the mark-up using a colour-coding schema. The approach developed required two versions of the HTML document to be produced: one for the editing and checking purpose, and one for display to the end user. However, it would be possible to also use this technique in conjunction with the DHTML technique described earlier.

The colour-coding simply applies different colours and/or fonts to different mark-up elements, such as `synph` DITA elements displayed in maroon. (Refer to *Figure 57: Example of CSS used for colour-coding the origin elements of processed HTML* (see page 243).)

An expansion of the technique currently under development is to move away from simple colour coding (which does not suit printing to monochrome media, and has accessibility problems) to richer, exposed mark-up. The expanded technique takes advantage of CSS2 features such as `:before` and `:after` *pseudo elements*. In some cases, CSS is proving to be too limited, and modifying the XSL-T templates that control the processing from DITA to HTML is a more comprehensive approach.



Case Study: Exposing Markup through CSS

During the processing (or transformation) of semantically marked-up documents (such as DITA) to HTML, Cascading Style Sheet (CSS) class attributes can be added to the derived HTML elements. For example, DITA mark-up of `<filepath>qa.js</filepath>` might be transformed to HTML mark-up of `<span class="filepath">qa.js</span>`. When the HTML is rendered in a browser, the class is ignored without error if no matching formatting definition for that class if found in the CSS. Often, the HTML classto match the DITA element is not defined in the CSS because a different rendering of that element is not wanted.

A software development company working in a DITA semantic mark-up environment needed a way of highlighting mark-up in the rendered HTML output to make it possible to identify the semantic mark-up element used so that

Figure 57: Example of CSS used for colour-coding the origin elements of processed HTML

The CSS additions to colour-code the output HTML based on its semantic origin would follow the pattern:

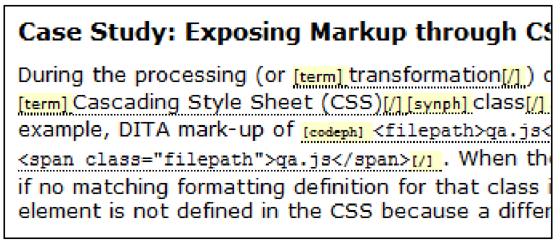
```
.synph {
 color:maroon;
 background-color: red;
}

.term {
 background-color: navy;
 color:white;
}

pre.codeblock, samp.codeph {
 font-size: 120%;
 background-color:gray;
}

span {
 background-color: yellow;
}
```

Using the `:before` and `:after` *pseudo elements*, the output can show the semantic origin in a more sophisticated way, with the principal benefit being that semantic labelling can be printed to monochrome media. (Refer to *Figure 58: Example of CSS pseudo elements used for exposing the origin elements of processed HTML* (see page 244).)



Case Study: Exposing Markup through CS

During the processing (or [term] transformation[/]) c
[term] Cascading Style Sheet (CSS)[/] [synph] class[/]
example, DITA mark-up of [codeph] <filepath>qa.js<
<span class="filepath">qa.js</span>[/] . When th
if no matching formatting definition for that class i
element is not defined in the CSS because a differ

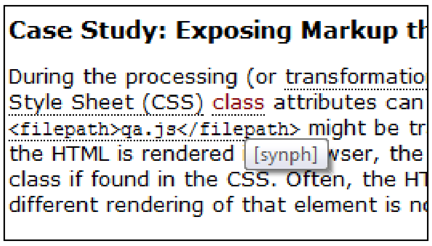
Figure 58: Example of CSS pseudo elements used for exposing the origin elements of processed HTML

The CSS additions to prefix the original DITA element name would follow the pattern:

```
.term {
 background-color: white;
 color:black;
 border-bottom: #000 1px dotted;
}
.term:before{
 content:"[term]";
 background-color: yellow;
 font-size: 70%;
}
.term:after{
 content:"[/] ";
 background-color: yellow;
 font-size: 70%;
}
```

Changing the XSL-T transformation templates can make it possible to show the semantic origin without cluttering up the text with mark-up. An HTML `acronym` element (with a `title` attribute value of the name of the original DITA element) can be injected around the `span` element normally added. This `acronym` element results in a *hover text* or *tooltip* pop-up displaying the semantic origin when the user holds the mouse over the element *(Self, n.d.)*.

A visual indicator that text has mark-up is discrete, dotted underlining. (Refer to *Figure 59: Example of acronym tag used for identifying the origin elements of processed HTML* (see page 245).) This technique is only useful for on-screen reading, as the pop-ups do not appear on paper!



Case Study: Exposing Markup th

During the processing (or transformatio
Style Sheet (CSS) class attributes can
<filepath>qa.js</filepath> might be tr
the HTML is rendered [synph] wser, the
class if found in the CSS. Often, the HT
different rendering of that element is n

Figure 59: Example of `acronym` tag used for identifying the origin elements of processed HTML

The supporting CSS would follow the pattern:

```
acronym {
 cursor: help;
 border-bottom: #000 1px dotted;
}
span {
 color: maroon;
 background-color: yellow;
}
```

Using techniques such as these make it possible to print a rendition of the document with exposed semantic mark-up, in turn making it possible to edit and proof-read XML-based semantic documents on paper.

Editing Marks for XML Mark-up - Conclusion

The standard editing marks used to indicate required changes in a printed document are not suitable for semantic authoring, where different presentational forms of a document can be automatically produced from the same source. Changes to editing practice are required to make editing of semantic documents more effective.

This paper proposes two approaches to make it practical to edit XML-based semantic documents on paper:

- Use an alternative set of editing marks for semantic authoring, incorporating those conventional editing marks that remain appropriate, and discarding those marks that are not.
- Use technical solutions based on DHTML, CSS and XSL-T to allow the display of XML mark-up in paper-based renditions of the document without significantly degrading the readability.

The adoption of these two approaches will together allow paper-based editing and proof-reading of documents to continue into the era of XML-based documents that separate content from form.

Editing Marks References

- André, J. (1998). Petite histoire des signes de correction typographique. Cahiers GUTenberg, (31), 45-59.
- André, J., & Richy, H. (1999). Paper-Less Editing and Proofreading of Electronic Documents. EuroTeX'99 Proceedings. Retrieved July 8, 2010, from http://www.irisa.fr/imadoc/articles/1999/heidelberg.pdf.
- Gardiner, D. (2010, May). Discovering XML for editing. The Canberra Editor, 19(4), 6-10.
- International Standard ISO/IEC 5776: 1983: Graphic technology - Symbols for text correction. (1983). International Organization for Standardization.
- O'Hara, K., & Sellen, A. (1997). A Comparison of Reading Paper and On-Line Documents (pp. 335-342). Presented at the Proceedings of CHI '97, Human Factors in Computing Systems, Atlanta. Retrieved July 13, 2010, from http://scholar.google.com.ezproxy.lib.swin.edu.au/scholar?q=kenton+o%27hara&hl=en&lr=&btnG=Search.
- O'Keefe, S. (2006, April 12). Wednesday at WritersUA: Structured authoring: taking the plunge. Scriptorium. Retrieved May 15, 2010, from http://www.scriptorium.com/blog/2006/04/wednesday-at-writersua-structured-authoring-taking-the-plunge.html
- Snooks and Co. (2002). Style manual: for authors, editors and printers (6th ed.). Brisbane: Wiley Australia.
- Self, T. (n.d.). The Mysterious Acronym Tag. HyperWrite. Commercial, . Retrieved July 6, 2010, from http://www.hyperwrite.com/Articles/acronym_tag.aspx.
- Self, T. (2009). DITA and the challenges of single-source article publishing. In *Communication, Creativity and Global Citizenship: Refereed Proceedings of the Australian and New Zealand Communications Association Annual Conference*. Presented at the ANZCA Annual Conference, Brisbane. Retrieved July 2, 2010, from http://www.cpe.qut.edu.au/conferences/2009/anzca/proceedings/Self_ANZCA09.pdf.
- Simpson, P. (1935). Proof-Reading in the Sixteenth, Seventeenth and Eighteenth Centuries (1st ed.). London: Oxford University Press.

- The Chicago Manual of Style. (1982). (13th ed.). Chicago: University of Chicago Press.
- Wilkinson, R. T. (1987). Proof-reading: VDU and paper text compared for speed, accuracy and fatigue. Behaviour and Information Technology, 6(2), 125-133. doi: 10.1080/01449298708901822.

Appendix

B

Industry Conferences and Publications

Topics:

- *Industry Conferences*
- *Industry Publications*

Since the acceptance of his research proposal in August 2008, the author has shared his findings and proposals by presenting at industry conferences and contributing to industry publications.

Industry Conferences

Presentation	Conference	Location	Date
Workshop: Introduction to DITA	UAConference Europe	Edinburgh, UK	18-19 September 2008
Forum Chair: *OASIS DITA for Help Initiative*	UA Conference Europe	Edinburgh, UK	18-19 September 2008
Workshop: *Introduction to DITA*	TCANZ Writers Forum 2008	Auckland, NZ	9-10 October 2008
DITA Workshop	Cherryleaf Seminars	London	10 November 2008
DITA Case Study: Proposal Generator	DITA Europe 2008	Munich, Germany	17-18 November 2008
Moderator: *Future trends in Business and Technical Communication*	STC France Chapter Conference	Paris	20-21 March 2009
DITA Workshop	Cherryleaf Seminars	London	26 March 2009
Workshop: Introduction to DITA	WritersUA Conference for Software User Assistance	Seattle, USA	29 Mar - 2 Apr 2009
Forum Chair: *Creating Help with DITA*	WritersUA Conference for Software User Assistance	Seattle, USA	29 Mar - 2 Apr 2009

Presentation	Conference	Location	Date
Introducing the DITA OT (joint presentation with Pam Noreault)	WritersUA Conference for Software User Assistance	Seattle, USA	29 Mar - 2 Apr 2009
Workshop: Hands On DITA	AODC 2009	Melbourne	20-22 May 2009
Writing to STOP	AODC 2009	Melbourne	20-22 May 2009
WinANT Echidna	ComTech Webinar	Online	16 June 2009
Writing to STOP	Scriptorium Webinar	Online	15 July 2009
Chair: *DITA Forum*	UA Conference Europe	Cardiff, UK	17-18 September 2009
Using DITA for Help	UA Conference Europe	Cardiff, UK	17-18 September 2009
Writing to STOP	UA Conference Europe	Cardiff, UK	17-18 September 2009
Writing to STOP	tcWorld 2009	Wiesbaden, Germany	4-6 November 2009
WinANT Echidna: Simplifying and Automating DITA Publishing	DITA Europe 2009	Munich, Germany	16-17 Nov 2009
DITA Update	WritersUA Conference for Software User Assistance	Seattle, USA	21-24 March 2010
WinANT Echidna	WritersUA Conference for	Seattle, USA	21-24 March 2010

Presentation	**Conference**	**Location**	**Date**
	Software User Assistance		
DITA Update	AODC 2010	Darwin	12-14 May 2010
WinANT Echidna	AODC 2010	Darwin	12-14 May 2010
DITA Conversion and Conditional Publishing	Cherryleaf Seminars	London	26 May 2010
Trends in Technical Communication (with Sarah O'Keefe and Ellis Pratt)	Scriptorium Webinar	Online	30 July 2010
An Update on DITA: Features, Tools and Best Practices	European Help Conference 2010	Stockholm	16-17 September 2010
Case Study: Visma Software DITA Documentation	DITA Europe 2010	Vienna, Austria	15-16 November 2010
Acronyms	Université de Bretagne Occidentale Lecture	Brest, France	10 November 2010
User Assistance	Université Rennes 2 Workshop (in conjunction with Université Clermont Ferrand)	Rennes, France	12 November 2010

Industry Publications

- Self, T. (2009, March). Improved Glossary and Terminology Handling Features in DITA 1.2. HyperWrite. Commercial. Retrieved November 25, 2009, from http://www.hyperwrite.com/Articles/showarticle.aspx?id=86
- Self, T. (2010, October). Writing to STOP. CIDM Information Management News, 10(10). Retrieved February 10, 2011, from http://www.infomanagementcenter.com/enewsletter/2010/201010/second.htm.
- Self, T. (2010, December). Reducing Context in Modular Documents. Best Practices Newsletter (CIDM), 12(6), 133-140.
- Self, T. (2011, March). Can you explain that again? DITA for beginners. tcworld, 19-21.

Appendix

C

Ancillary Projects

Topics:

- *WinANT Echidna project*
- *TACTICS Consulting learning materials*
- *Structured Authoring subject at Swinburne University of Technology*

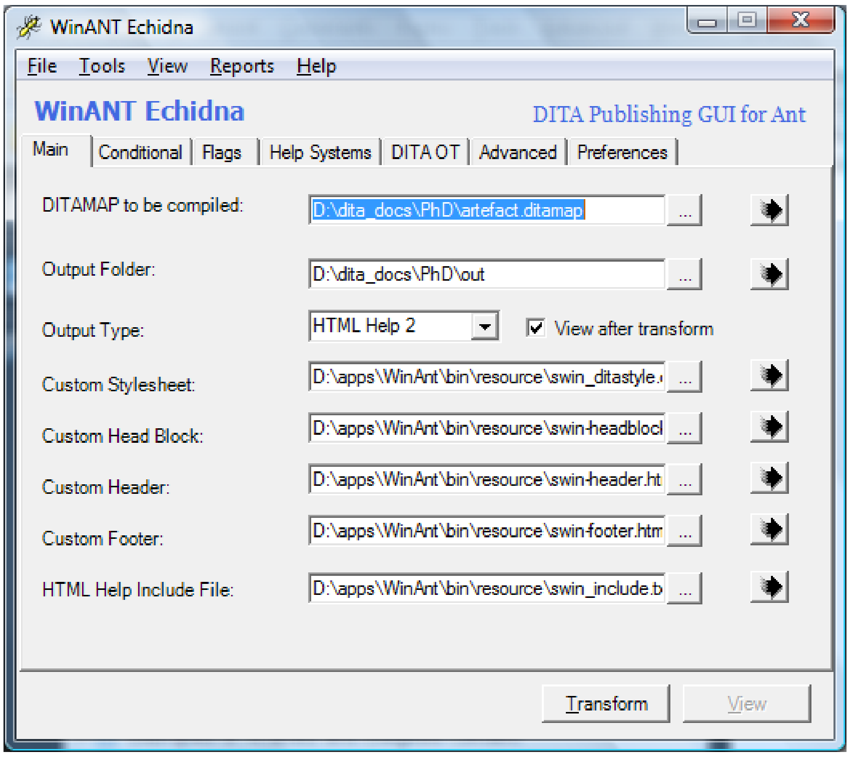
WinANT Echidna project

WinANT Echidna is an open source software tool developed by the author. It serves as a Microsoft Windows interface to the DITA Open Toolkit.

I created the WinANT Echidna program on realising that the standard method of using the DITA Open Toolkit required re-entering configuration information on every build, and working with a command line interface. The initial purpose of WinANT Echidna was to provide a *point-and-click* Microsoft Windows interface to the DITA Open Toolkit.

As I progressed with the development of the Artefact, WinANT Echidna became more important for periodically building modules to provide drafts that were easier to edit on paper. Some of the ideas developed in the Artefact, such as rules for variables and conditional processing, could be easily tested with WinANT Echidna.

WinANT Echidna attracted some interest in the DITA community, and I made the decision to move the software project to open source. This was completed in February 2009, with the project being managed through the SourceForge facility.



DITAMAP to be compiled: D:\dita_docs\PhD\artefact.ditamap
Output Folder: D:\dita_docs\PhD\out
Output Type: HTML Help 2 ☑ View after transform
Custom Stylesheet: D:\apps\WinAnt\bin\resource\swin_ditastyle.
Custom Head Block: D:\apps\WinAnt\bin\resource\swin-headblocl
Custom Header: D:\apps\WinAnt\bin\resource\swin-header.ht
Custom Footer: D:\apps\WinAnt\bin\resource\swin-footer.htm
HTML Help Include File: D:\apps\WinAnt\bin\resource\swin_include.b

Figure 60: WinANT Echidna publishing tool for DITA

I added a series of reporting features to help with the management of the Artefact DITA source, extending the functionality of WinANT Echidna beyond just an interface to the DITA Open Toolkit.

Additional features aimed to simplify the process of creating PDF output were added, again primarily to enable better quality drafts of the Artefact to be produced. This work led to my development of PDF plug-in *customisations* as part of this contribution to the OASIS DITA Adoption Technical Committee.

The WinANT Echidna project therefore influenced the development of the Artefact by allowing ideas to be tested, by providing management reporting, and by enabling production of better quality PDF output. This Exegesis was produced in PDF form, and in HTML-based form, using the features of WinANT Echidna developed alongside the research project.

TACTICS Consulting learning materials

> *My paid consultancy work included the development of learning materials for a commercial introductory structured authoring training course. The training workbook shared some content topics with the Artefact.*

In my capacity as a documentation consultant, I designed a training course for new DITA authors for the TACTICS Consulting training and documentation company in Australia.

The main deliverable of this work was a 92 page workbook, *Structured Authoring in XML (Self, 2008)*, which was completed in late 2008.

The workbook used the Information Mapping methodology, which has some philosophical similarities to DITA.

The TACTICS project influenced the development of the content and structure of the Artefact, and served as an exemplar of content re-use across different publications.

Structured Authoring subject at Swinburne University of Technology

> *My involvement in the curriculum development and the teaching of the Structured Authoring post-graduate university subject provided an opportunity to interact closely with students new to DITA, and to reflect on the questions asked and problems encountered by those new users.*

I developed the curriculum for the Swinburne University of Technology *Structured Authoring (HATC424)* subject from mid-2006 to mid-2007. The subject, which focusses on DITA, forms part of the Graduate Diploma in Technical Communication programme.

The learning materials for the *Structured Authoring* subject were managed in the same topic repository as that used for the Artefact.

During the research project, I taught the *Structured Authoring* subject to students who were precisely the target audience for *The DITA Style Guide*: technical communicators fresh to DITA. I used the opportunities to analyse and reflect on the needs of authors new to DITA. The interaction with the students was useful not only in identifying questions that new DITA authors might ask, but in developing justifications and rationales for the guidelines and rules.

The experience with the *Structured Authoring* project influenced the development of the rules and guidelines of the Artefact.

Appendix

D

Contents of CD-ROM

The CD-ROM accompanying this Exegesis contains the following documents and applications:

- The Artefact (standard published *The DITA Style Guide: Best Practices for Authors*) in ePUB format.
- The Artefact (standard published *The DITA Style Guide: Best Practices for Authors*) in Microsoft HTML Help (CHM) format.
- The standard page layout version of the Artefact in PDF format.
- The Artefact, with exposed rationale and links to DITA Language Reference, in Microsoft HTML Help (CHM) format.
- The Exegesis with links to related Artefact topics in Microsoft HTML Help (CHM) format
- WinANT Echidna application setup files.

Index