



Proposals for DITA machine industry

Jonatan Lundin (jonatan.lundin@citec.se)

rev:A 2008-02-01

the one for the job

crtec

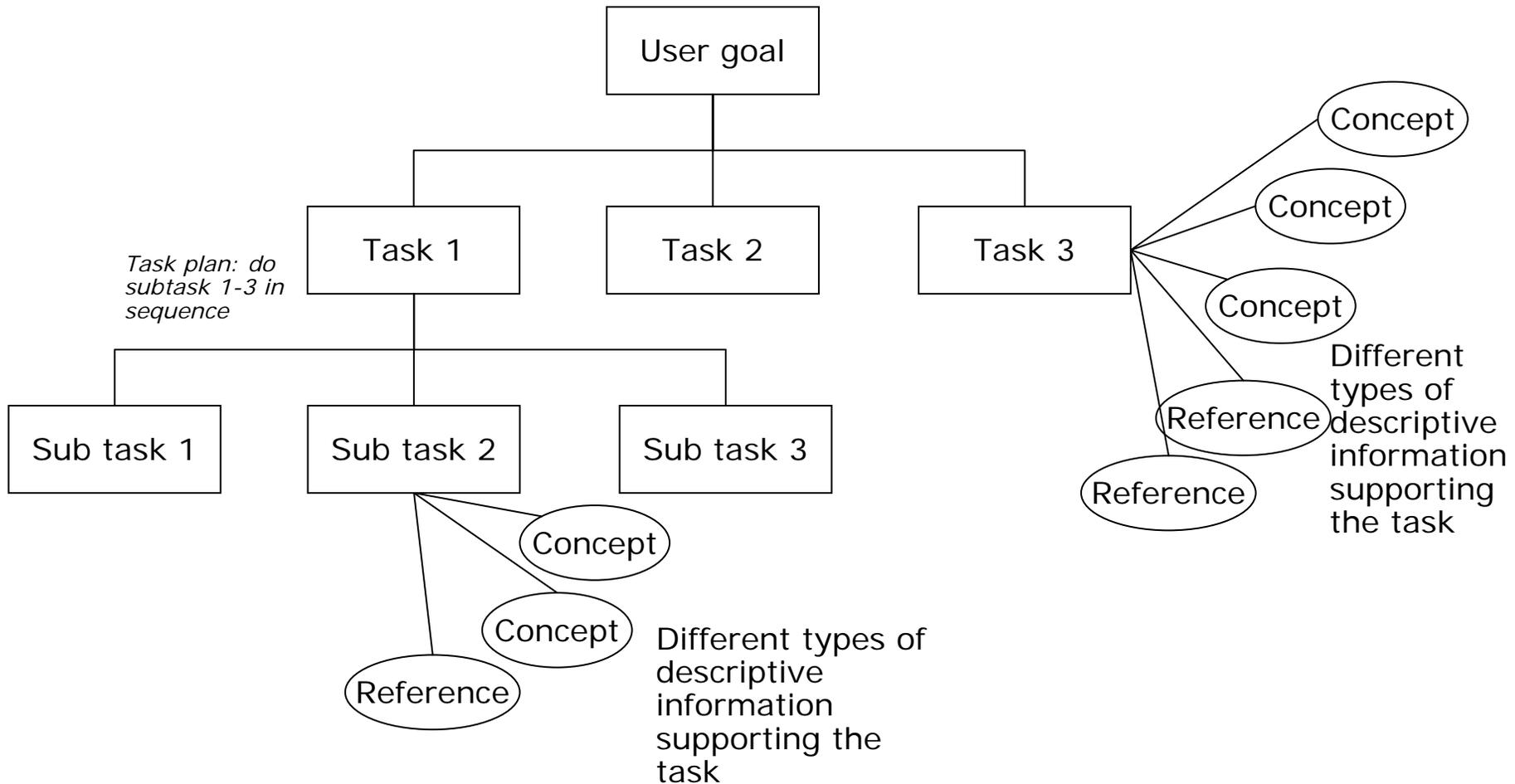
Agenda

- Information typing
- Topic resolution and storing
- Filtering logic
- MathML domain
- Support for task plans
- Misc.
- Navigation behaviours

Information typing

- The information analysis/architecture in DITA starts from a task analysis. Hierarchical task analysis is preferable used. The descriptive information (concept and reference), supporting the different tasks is identified in the task analysis.
- The minimalist approach can be applied in machine industry area even though I think the user is not "learning-by-doing" as much as in human-computer-interaction area (more "learning-by-reading" in machine industry)
- This is different from the ILS (S1000D, PLCS etc) perspective where the starting point is the material breakdown structure (and DMRL). The needed data modules are identified in relation to the structure (and also the maintenance plan)
- Even in this perspective the amount and types of descriptive information is greater than in the software industry. Depending on the nature of the task different types of descriptive information is needed. A manual for a power plant product contains more descriptive information than a software manual.

Information typing (contd)



Information typing (contd)

- Is then concept and reference information types enough in machine industry?
 - Should it be possible to distinguish between, for example, the following descriptive types:
 - Information describing the design of a machine (or part of); what components is it built up from and their designation, interface etc
 - Information describing the operation of a machine (or part) of; the theory of operation or functionality
 - Information describing the application of a machine (or part of); In what contexts is it suited, the purpose
 - Information describing the requirements a machine (or part of) imposes on its environment/context/surrounding
 - Information describing technical data (weight, measures, performance etc)
 - The category element in prolog/metadata element can be used to define the type, but is it enough? Possibility to use a choice list? Different mark-up?
-

Topic resolution and storing

- The needed descriptive topics (concept and reference) derived from task analysis can be associated with a product component object
- The descriptive topics are maybe stored in a CMS. The storage structure in CMS can represent the product data model according to a PLCS (**ISO 10303 AP 239**), model where the topics are stored to its corresponding object in the structure. Or the data model in DITA CMS can be defined using the same model as in a CSDB (S1000D).
- There are ongoing investigations and pilot cases to establish the relationship between S1000D and PLCS within OASIS/OPEN (www.oasis-open.org).
- What is the relation between DITA machine industry SC and PLCS/S1000D in this perspective? The product attribute on topic root element should hold the part no/article no? Or is there need for another metadata?
- The DITA CMS should also have a task oriented structure to hold the task topics

Filtering logic

- DITA 1.1 includes filtering/flagging logic
- This is very useful in machine industry since a product often has variants (product configurations) and thus it is required to filter out a specific product configuration manual from a “master” map.
- Each component in a machine often has a unique identity (part no – can for example be extracted from DMC in S1000D)
- The product and platform attributes on various elements is then used to hold the flagging property?
- Should there be only one (1) attribute for flagging, since there are need for more attributes than product, platform, audience etc? What about market localization (ANSI, IEC etc), company specific information (company-Sweden, company-Germany), detail level (summary, deep), manual type (manual1, manual2 etc)? How many flagging attributes can we identify? S1000D uses only one?

Filtering logic contd

- The ditaval file can represent a specific product configuration (specifies all components - part no/articles - in a product)
- The current functionality represents OR operator
- There is a need to have AND operators (maybe INV)

```
<p product="productA AND productB">
```

The element shall be excluded if ditaval file is:

```
<prop att="product" val="productA" action="exclude"/>
```

```
<prop att="product" val="productB" action="exclude"/>
```

Filtering logic contd

- Possibility to exclude all elements having a value not specified in the ditaval file?
- Meaning; today you can specify which elements you want to exclude, but the opposite should also be possible: Include only the elements specified in the ditaval file (thus exclude all others)

```
<p product="productC">
```

The element shall be excluded if ditaval file is:

```
<prop att="product" val="productA" action="exclude"/>
```

```
<prop att="product" val="productB" action="exclude"/>
```

```
<prop att="product" val="AllOtherValuesThanTheOncesSpecifiedInThisFile  
  action="exclude"/>
```

MathML domain

- Include an equation element (inline and block) with the same possibilities as the image element.
- Include the possibility to build equations using the Mathematical Markup Language (MathML™)
- Tool support?

Support for task plans

- Task plan element in task topics? Starting from a Hierarchical Task Analysis (HTA) means identifying task plans.
- The DITA map can represent a hierarchical task tree.
- The collection-type attribute (sequence, sibling, family etc) in map node elements corresponds to task plans (even though more values should be available?).
- The same possibility should be available in a task topic. The different steps in a task topic can maybe performed in any order? Or depending on a context a certain step is optional (“if it is raining don’t do step X” for example)?

Misc.

- Inline element for electrical units (V, A, Hz etc)
- Type element on image element to define type of image (drawing, diagram, screen shot etc)

Navigation behaviours

- What has DITA given the active user (the behavior that minimalism is based on)?
- A DITA manual has a topic hierarchy (the map hierarchy) where each topic has a title that is usually numbered (when formatted as paper) according to the placement in the hierarchy (1.1, 1.1.1, 1.1.2 etc). What is, at a first glance, different from the traditional linear manual?
- OK: the topics are standalone and does not have explicit relation to other topics (no: “as we said before” etc). There are no glue information as well. And we have the short description and relation tables. DITA focuses on task and gives more task oriented manuals (descriptive information is secondary). This is good. But is it enough?
- Does the topic position in the hierarchy have a meaning? Can I place my topic anywhere in the hierarchy and nothing is lost? What meaning does the position have if it matter? After all; the topics are standalone.
- OK; if my map is structured according to the HTA tree then I can not place the topics in a random order?
- The active user does search very much; when he/she got stuck using the product the manual is used and to find the error recovery information and a search process is started. Some say that the user spends half of the time using information on searching.
- Is this taken in to account in DITA? I don't see that happen. If it was then it would be possible to control how the navigation interfaces are built (TOC for example). Now this is controlled outside DITA (formatting/publishing tools).
- If a better support for building and controlling the navigation interfaces was part of DITA then the topic position in the hierarchy wouldn't matter, making the DITA manual truly modularized for the active user.

Navigation behaviours

- The navigation interface (TOC for example) is very important in the active user and minimalist paradigm. The user puts a lot of time in searching in documentation. Different users (depending on age, knowledge etc) have different ways of searching and uses different search questions; thus different navigation interfaces are needed for one manual.
- The traditional TOC is based on the map hierarchy and uses topic titles as entries (or navtitle) in the TOC. The TOC is defined (where it should show up and the way it is arranged) in formatting/publishing tools (DITA OT (XSL-FO), FrameMaker etc). This must be controlled from inside the DITA markup, not outside.
- Today DITA allows you to compile a TOC from multiple DITA maps and you can omit parts of the map using the toc attribute.
- You can not, for example, define how the TOC shall be organized (different from the map hierarchy), have multiple TOCs for the same map or define where in the map hierarchy the TOC should be inserted using the available DITA markup.
- The TOC is not only a formatting issue; it is a very important part of the content. The TOC is a layer above the content (compare to topicmaps).
- The Deck of cards manual is used as example (see other PDF).

Navigation specification template 1

NavSpecTemplate1.navref

```
<navSpecTemplate>
  <title>Table of content</title>
  <qualifier criteria="All" order="AccordingToMapHierarchy"
title="topic"/>
</navSpecTemplate >
```

NavSpecTemplate2.navref

```
<navSpecTemplate>
  <title>Table of content – sorting per type of info</title>
  <title>Diamonds</title>
  <qualifier criteria="Diamonds" order="Descending" title="topic"/>
  <title>Hearts</title>
  <qualifier criteria="Hearts" order="Descending" title="topic"/>
  <title>Clubs</title>
  <qualifier criteria="Clubs" order="Descending" title="topic"/>
  <title>Spades</title>
  <qualifier criteria="Spades" order="Descending" title="topic"/>
</navSpecTemplate >
```

NavSpecTemplate3.navref

```
<navSpecTemplate>
  <title>Table of content – sorting per product parts</title>
  <title>No2</title>
  <qualifier criteria="No2" order="Descending" title="topic"/>
  <title>No3</title>
  <qualifier criteria="No3" order="Descending" title="topic"/>
  <title>No4</title>
  <qualifier criteria="No4" order="Descending" title="topic"/>
  <title>No5</title>
  <qualifier criteria="No5" order="Descending" title="topic"/>
</navSpecTemplate >
```

```
<map>
  <navref href="NavSpecTemplate1.navref"/>
  <navref href="NavSpecTemplate2.navref"/>
  <navref href="NavSpecTemplate3.navref"/>
  <topicref href="Hearts2.dita" navtitle=""/>
  <topicref href="Hearts3.dita" navtitle=""/>
  <topicref href="Hearts4.dita" navtitle=""/>
  ..
  <topicref href="AceOfSpades.dita" navtitle=""/>
</map>
```

Or

```
<topicref href="navSpecTemplate3.navref" format="navref"/>
```

Navigation specification template 2

NavSpecTemplate2.navref

```
<navSpecTemplate>  
  <title>Diamonds (Installing topics): </title>  
  <qualifier criteria="Diamonds" order="Descending" title="topic"/>  
</navSpecTemplate >
```

All topics in map meeting the criteria are listed in the node in the table of content. The qualifier element can be nested to make hierarchy.

Example of formatted TOC

Topics (card) sorted per type of information.....	Page
Diamonds (Installing topics):	
Diamonds No2.....	33
Diamonds No3.....	34
Diamonds No4.....	35
Diamonds No5.....	36
Diamonds No6.....	37
Diamonds No7.....	38
Diamonds No8.....	39
Diamonds No9.....	40
Diamonds No10.....	41
Jack of Diamonds.....	42
Queen of Diamonds.....	43
King of Diamonds.....	44
Ace of Diamonds.....	45

Navigation specification template 3

NavSpecTemplate1.navref

```
<navSpecTemplate>  
  <qualifier criteria="CHapterTOC" order="AccordingToMapHierarchy"  
  title="navref"/>  
</navSpecTemplate >
```

```
<map>  
  <navref href="NavSpecTemplate1.navref"/>  
  <topicref href="Hearts2.dita" navtitle=""/>  
  <topicref href="Hearts3.dita" navtitle=""/>  
  <topicref href="Hearts4.dita" navtitle=""/>  
  <navref href="NavSpecTemplate2.navref"/>  
    <topicref href="AceOfSpades.dita" navtitle=""/>  
    <topicref href="AceOfSpades.dita" navtitle=""/>  
    <topicref href="AceOfSpades.dita" navtitle=""/>  
  <topicref href="AceOfSpades.dita" navtitle=""/>  
</map>
```

Enabling two levels of TOC. One (1) main in the beginning of the manual and one (1) TOC in each chapter. The main TOC references each chapter TOC. The chapter TOC includes all child topics relative to the node where the navref is inserted in the map.

NavSpecTemplate2.navref

```
<navSpecTemplate>  
  <title>Chapter TOC</title>  
  <qualifier criteria="All" order="AccordingToMapHierarchy"  
  title="topic"/>  
</navSpecTemplate >
```

Conclusion and ideas

- The navref element should be possible to insert anywhere in the map. This will enable to make TOCs for each chapter in a manual.
- It must be possible to reference a certain node in the map to state where the TOC is to be inserted.
- What metadata on topics should be used for the criteria to select topics to a node in the TOC? It is important to distinguish between different types of information and to what product part the topic is describing (relation). Task, concept and reference is not enough? The different types of information should correspond to the search questions a user states; thus information typing is important beyond the default DITA types.
- The order in which the topics are stored in the map is not important (what does the order and hierarchy tell the user?). Since the topics are standalone it doesn't matter which topics are placed before and after a specific topic in the map hierarchy. You can place the topic anywhere in the map. The important issue is that a certain topic must be easy to find.
- With the navref functionality it would be possible to just reference all topics needed in a flat structure in the DITA map and then define various hierarchies (for example the HTA tree) as navigation interface (TOC/navref).
- Is a metapart in the navref file needed? Also a main title to hold the type of TOC (table of content etc)?