



Proposals for DITA machine industry

- Including a topicmap light version in DITA
(ditatoctemp)

Jonatan Lundin (jonatan.lundin@citec.se)

2008-08-18

Background: DITA and topicmap (1)

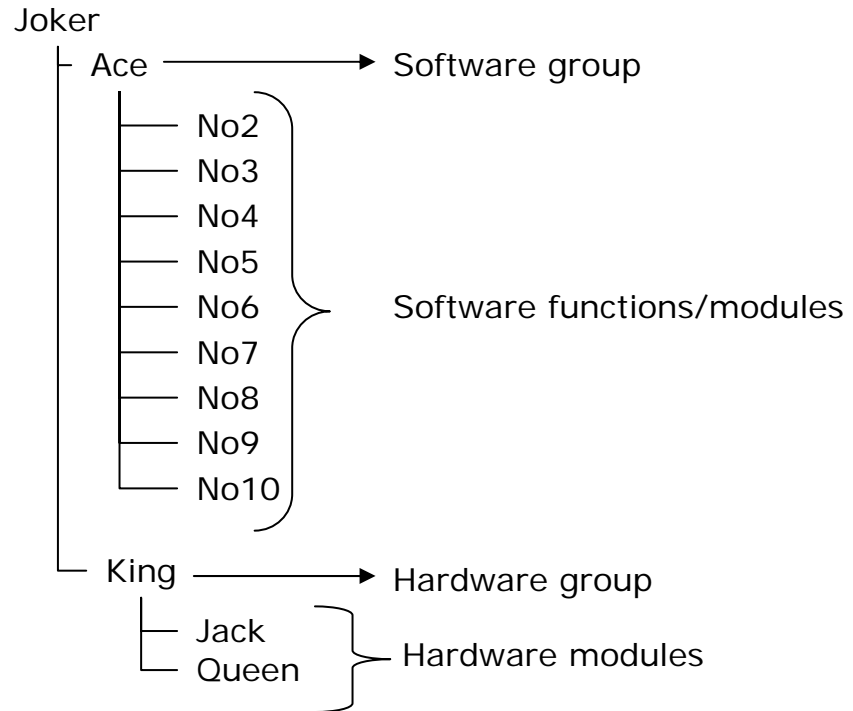
A machine product has the following components (product structure):

The numbers and images in a deck of card is used as an example.

King, Jack and Queen are hardware components and Ace, No2, etc are software components (functions/modules). They all are integrated to form a product.

The examples in this presentation does not invoke different user categories or configuration management issues (version handling, filtering etc). The sequence in the examples shall not be interpreted as the steps in a method for information analysis.

Product structure:



Background: DITA and topicmap (2)

A specific user goal is to install and test the machine product.

The work (to reach user goal) must be carried out in a specific order, analyzed via a hierarchical task analysis (HTA).

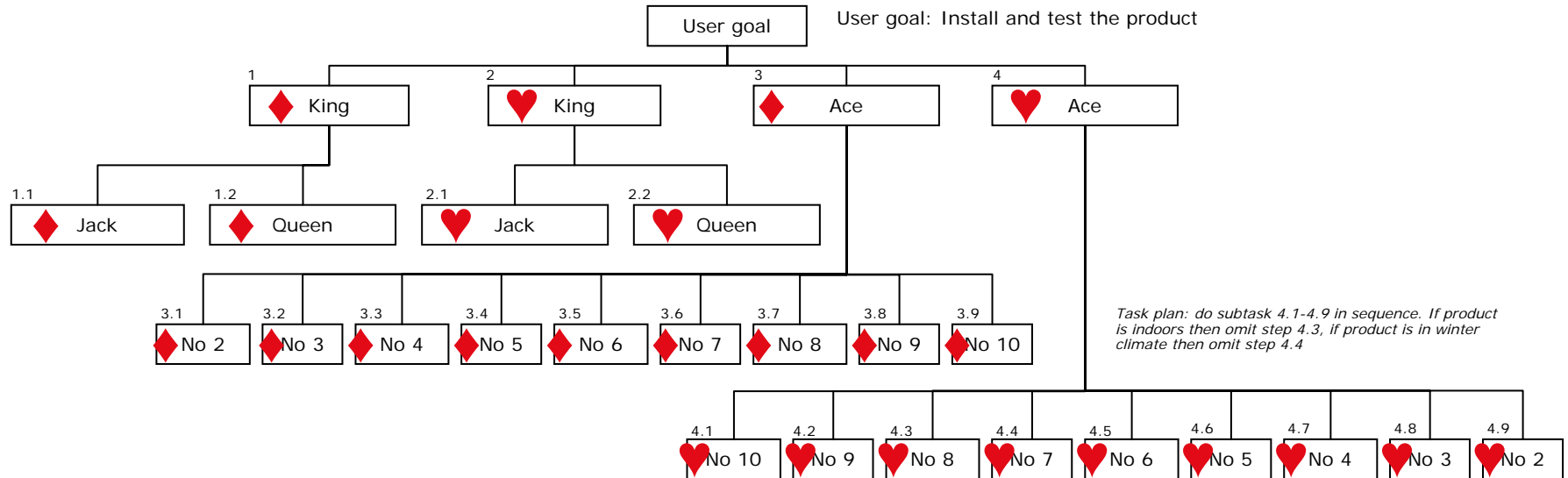
Hierarchical task node types:

♥ = Install

♦ = Test

♦ No 10

Legend
The task is interpreted as
"Test software component 10"



Background: DITA and topicmap (3)

Now we know which steps that must be carried out to install and test the machine product. We also now which components are handled in which order.

But the user needs installation and testing instructions and supporting functionality and design descriptions to be able to perform the work.

The mapping of the types of cards in the deck of cards and the DITA types is as follows:

DITA task topics

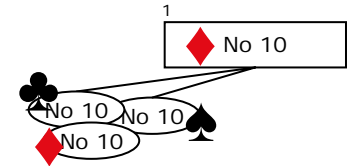
♥ = Installation instructions

♦ = Testing instructions

DITA concept topics

♠ = Functionality descriptions
(describes how something (subject) works – theory of operation)

♣ = Design descriptions
(describes how something (subject) is designed)



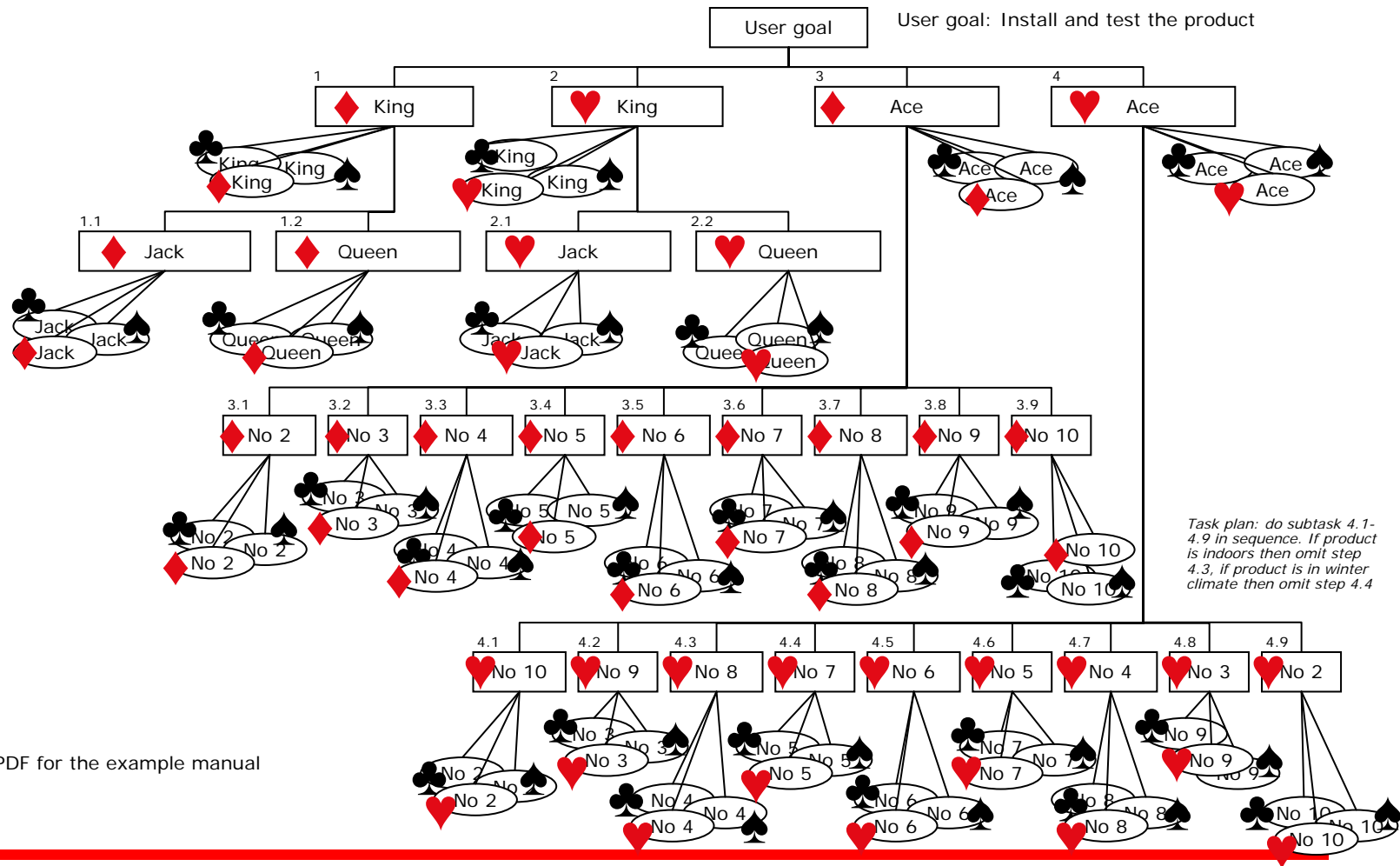
*Legend:
To be able to test the software component 10 the user needs*

- a task topic with testing instructions ("Testing software component 10" – the oval circle with "No 10" and the diamond icon),
- a concept topic describing the functionality of component 10 and
- a design description describing the design of component 10.

Background: DITA and topicmap (4)

The information need analysis result looks like this:

For each task node we need a task topic (of certain type) and descriptive information to support the task



See deck of cards PDF for the example manual

Background: DITA and topicmap (5)

Now we know which topics we need to create and write. How should they be packaged to the user? In a manual of course.

A DITA map is created which has the following hierarchy (according to the hierarchical task analysis).

Task and descriptive information are mixed. Descriptive information are child of it task topic (related to the same component). This is valid only for diamonds – not hearts.

A relation table is created. The linking and collection-type attributes are filled in according to the HTA-analysis. The task-plan information given for node 4 has to be put in the “Ace of Hearts” topic since the collection-type attribute does not support omitting just one topic.

- 1 [King of Diamonds](#)
 - 1.1 [King of Clubs](#)
 - 1.2 [King of Spades](#)
 - 1.3 [Jack of Diamonds](#)
 - 1.3.1 [Jack of Clubs](#)
 - 1.3.2 [Jack of Spades](#)
 - 1.4 [Queen of Diamonds](#)
 - 1.4.1 [Queen of Spades](#)
 - 1.4.2 [Queen of Clubs](#)
- 2 [King of Hearts](#)
 - 2.1 [Jack of Hearts](#)
 - 2.2 [Queen of Hearts](#)
- 3 [Ace of Diamonds](#)
 - 3.1 [Diamonds No2](#)
 - 3.1.1 [Clubs No2](#)
 - 3.1.2 [Spades No2](#)
 - 3.2 [Diamonds No3](#)
 - 3.2.1 [Clubs No3](#)
 - 3.2.2 [Spades No3](#)
 - 3.3 [Diamonds No4](#)
 - 3.3.1 [Clubs No4](#)
 - 3.3.2 [Spades No4](#)
 - 3.4 [Diamonds No5](#)
 - 3.4.1 [Clubs No5](#)
 - 3.4.2 [Spades No5](#)
 - 3.5 [Diamonds No6](#)
 - 3.5.1 [Clubs No6](#)
 - 3.5.2 [Spades No6](#)
 - 3.6 [Diamonds No7](#)
 - 3.6.1 [Clubs No7](#)
 - 3.6.2 [Spades No7](#)
 - 3.7 [Diamonds No8](#)
 - 3.7.1 [Clubs No8](#)
 - 3.7.2 [Spades No8](#)
 - 3.8 [Diamonds No9](#)
 - 3.8.1 [Clubs No9](#)
 - 3.8.2 [Spades No9](#)
 - 3.9 [Diamonds No10](#)
 - 3.9.1 [Clubs No10](#)
 - 3.9.2 [Spades No10](#)
- 4 [Ace of Hearts](#)
 - 4.1 [Ace of Clubs](#)
 - 4.2 [Ace of Spades](#)
 - 4.3 [Hearts No2](#)
 - 4.4 [Hearts No3](#)
 - 4.5 [Hearts No4](#)
 - 4.6 [Hearts No5](#)
 - 4.7 [Hearts No6](#)
 - 4.8 [Hearts No7](#)
 - 4.9 [Hearts No8](#)
 - 4.10 [Hearts No9](#)
 - 4.11 [Hearts No10](#)

Table of content in manual and hierarchy in DITA map

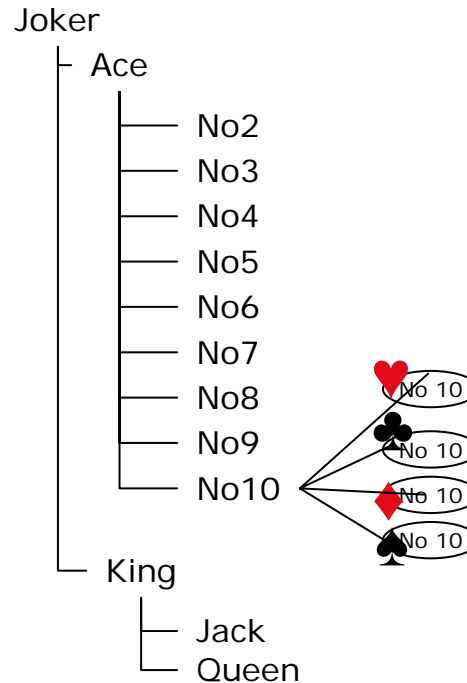
Background: DITA and topicmap (6)

The topics are stored in a CMS according to the product structure.

Task topics could (should) be stored according to the hierarchical task analysis (task oriented structure) in the CMS.

The CMS could also use a topicmap (shown on next slides) "behind the scene" when a technical writer is searching a topic.

Product structure in CMS:



Background: DITA and topicmap (7)

Everyone is happy or? It is known that user of technical documentation has problems finding the relevant information (see Jonatan master thesis). The traditional table of contents and index does not support the user enough. It is simply not possible to have one (1) content structure fitting every user reading the manual?

A solution would be to create a topicmap above the DITA content to support the variety of users.

The topicmap is more powerful than the traditional index, since the topicmap includes robust ways of addressing associations between subjects and also can cope with subject typing.

It is not possible to automatically generate a topicmap out from the DITA content. Well, it is, but the mapping of topicmap features to DITA features does not give an effective topicmap:

- Topicmap topic types=task, concept, reference?
- Topicmap topics=all topics in a DITA map?
- Topicmap occurrences type=DITA topic?
- Topicmap occurrences=all topics in a DITA map?
- Topicmap associations="related-to" (as given from reltable), "child" and "parent" as given from position in DITA map?
- Topicmap role types =what in DITA?

It could be possible to create a topicmap skeleton where certain data was captured from the DITA content (semi-solution) "on-the-fly".

Maybe it would be possible to make the DITA map act like a topicmap using specialization?

A mapping of the DITA content to create a topicmap is given on next slides.

Background: DITA and topicmap (8)

For task oriented information:

- *Topicmap topic types* could be the major parts of the HTA-tree (life cycle phases).

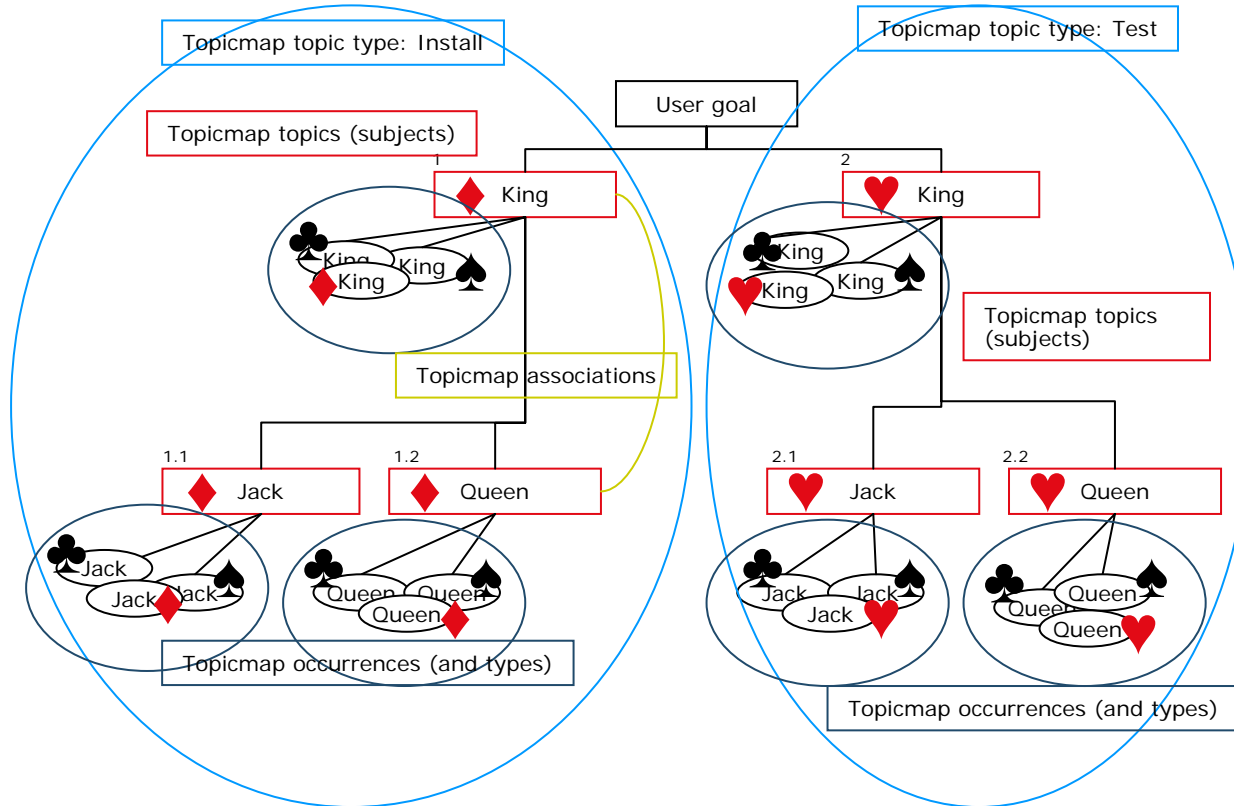
- *Topicmap topics* could be each node in the HTA-tree (of a certain topic type then).

- *Topicmap occurrence type* could be the different type of DITA topics (diamonds, hearts, clubs and spades in this presentation). We need to distinguish between different types of DITA task and DITA concepts. The task/concept/reference-category attribute does this (see other proposal).

- *Topicmap occurrence* could be each needed DITA topic for each HTA-node (of a certain type then).

- *Topicmap associations (and types)* could be the relation between the HTA-nodes (child/parent), the task-plan information (which order to carry out the subtasks) and relation to product components etc, etc.

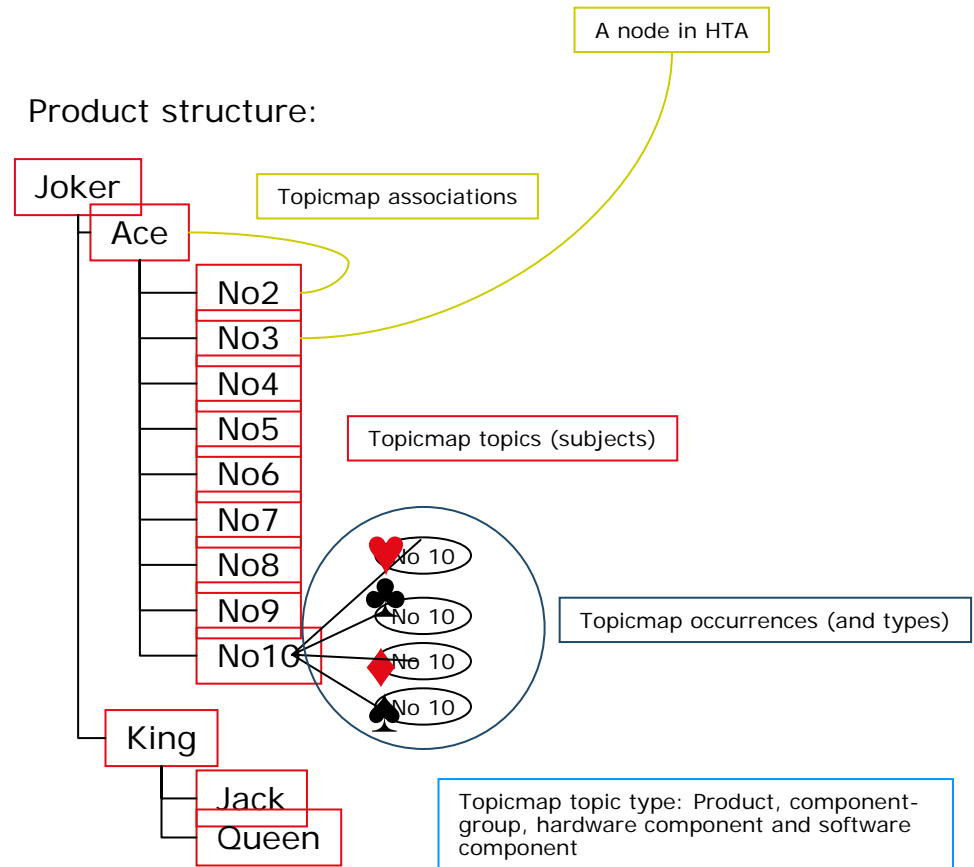
- *Topicmap roles (and types)* could be "Child" and "Parent".



Background: DITA and topicmap (9)

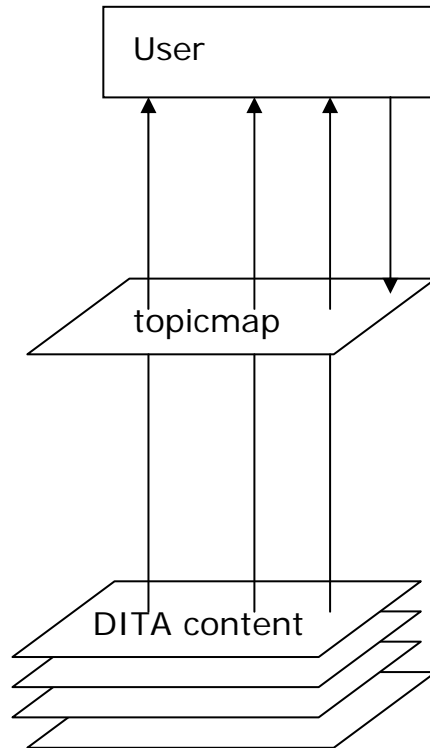
For product oriented (descriptive) information:

- *Topicmap topic types* could be the things the product is built up from; product, component groups, hardware-components and software-components.
- *Topicmap topics* could be each product, component group and component (software/hardware) a specific product consists of.
- *Topicmap occurrence type* could be the different type of DITA topics (diamonds, hearts, clubs and spades in this presentation). We need to distinguish between different types of DITA task and DITA concepts. The task/concept/reference-category attribute does this (see other proposal).
- *Topicmap occurrence* could be each DITA topic that is related to a product, component group and component.
- *Topicmap associations (and types)* could be the relation between the product-nodes (consists-of, is-part-of), the relation to the HTA-nodes etc etc.
- *Topicmap roles (and types)* could be "Child" and "Parent".



An example of a topicmap XML instance based on the presentation is available in the proposal

Background: DITA and topicmap (10)



The use of a topicmap for DITA content does not mean that you can just quickly ad hoc create some DITA topics and hope that the topicmap can make the whole thing usable to the user.

Instead you can put it the other way around and start from the topicmap and based on the result create DITA topics.

This implies that the starting point is to identify what the user is searching/looking for (needing).

The way of working today where you infotype topics is lost when publishing, since the type of information is not presented to the user anywhere (metadata is only used in production phase not in consumer phase). The reader could also benefit from knowing the metadata when searching.

This also implies that the DITA map cannot both cope with the navigation structure and to tell which topics that are included in a specific manual. The DITA map is instead only the tool to tell which topics are included in a manual. Search interfaces are defined with topicmaps, ditatocmp (see proposal) files and indexing of keywords.

Background: DITA and topicmap (11)

The topicmap enables the user to find a topic through various ways. The user can:

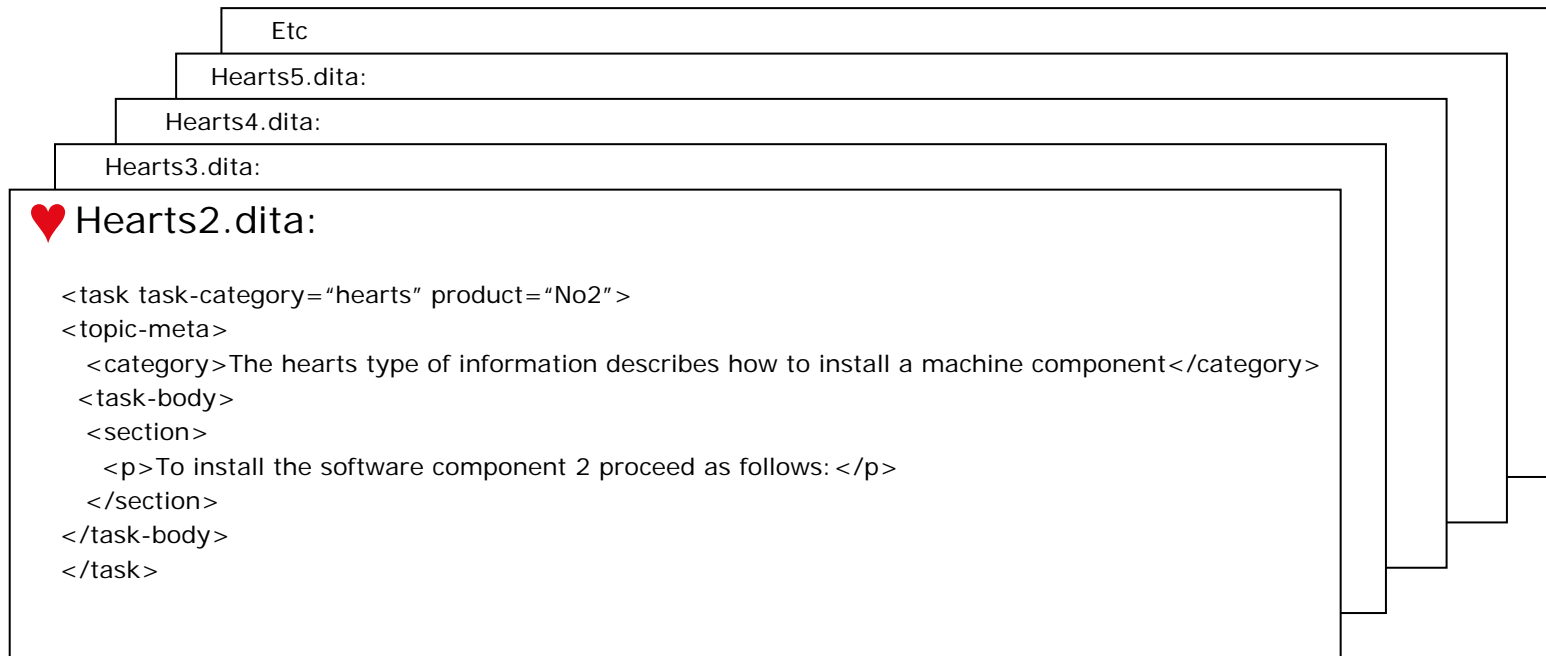
- start from a specific known product component (subject) and find all information (occurrences) relevant to the component.
- Start from the topic type "hardware component" to see a list of all (subjects) hardware components (to select one).
- Select a specific task node and see which sub tasks that must be carried out to perform the actual task node (via associations).
- Start from the life cycle phase "Install" and see all topics relevant in the installation phase. Once a specific node is selected all information relevant (occurrences) to the node can be displayed.
- Start from a specific task node and go to the product components relevant (via associations) to the task node.
- See all occurrences of a specific type (all functionality descriptions for all components).
- Etc

We want something similar in DITA markup! This proposal is about introducing a "DITA topicmap light version"

A topicmap is not suitable for paper documentation? The "DITA topicmap light version" must cope with paper media.

Use cases: ditatoctemp markup (1)

Each card in a deck of cards still corresponds to a DITA topic (one card with markup shown below) in this simulated example. The task-category attribute is included (see other proposal). The product attribute is used to hold the component to which the topic is related even though it might not be the best solution.



Use cases: ditatoc temp markup (2)

We also need a bookmap.

The proposal is to create a “table of contents (TOC)” template. Each TOC template “references” topics of certain types (addressing certain topic attributes). *The solution for how a topic shall be referenced must be further investigated (to give flexibility).*

The different TOCs are generated during publishing. Each ditatoc temp-file is a unique DITA XML file.

It must be possible to have many toc elements in the booklist, each one referencing a ditatoc temp-file (you could have just one ditatoc temp which has the same content as the separate files – but for the sake of reuse many must be an option).

```
<bookmap>
  <frontmatter>
  <booklist>
    <toc href="DITATocTemplate1.ditatoc temp"/>
    <toc href="DITATocTemplate2. ditatoc temp"/>
    <toc href="DITATocTemplate3. ditatoc temp"/>
    <toc href="DITATocTemplate4. ditatoc temp"/>

  </booklist>
</frontmatter>
<chapter>
  <topicref href="Hearts2.dita" navtitle="" />
  <topicref href="Hearts3.dita" navtitle="" />
  <topicref href="Hearts4.dita" navtitle="" />
  ..
  <topicref href="AceOfSpades.dita" navtitle="" />
</chapter>
</bookmap>
```

} Hierarchy
according to
HTA

Use cases: ditatoctemp markup (3)

Each ditatoctemp file has its own semantics (markup).

The first TOC is the traditional TOC which orders the topic according to the map hierarchy.

The second TOC “picks” out all topics of a certain information type and sorts them according to the type.

DITATocTemplate1:

```
<toc-template>  
  <title>Table of content</title>  
  <qualifier order="AccordingToMapHierarchy" title="topic"/>  
</toc-template>
```

DITATocTemplate2:

```
<toc-template>  
  <title>Table of content – topics (cards) sorted according to type of card</title>  
  <tocGroup>Diamonds</tocGroup>  
  <qualifier task-category="Diamonds" product="all" order="Descending" title="topic"/>  
  <tocGroup>Hearts</tocGroup>  
  <qualifier task-category="Hearts" product="all" order="Descending" title="topic"/>  
  <tocGroup>Clubs</tocGroup>  
  <qualifier concept-category="Clubs" product="all" order="Descending" title="topic"/>  
  <tocGroup>Spades</tocGroup>  
  <qualifier reference-category="Spades" product="all" order="Descending" title="topic"/>  
</toc-template>
```

Use cases: ditatoc temp markup (4)

The third TOC "picks" out all topics that describes a certain software component and sorts them according to the component.

The fourth TOC "picks" out all topics that describes a certain hardware component and sorts them according to the component.

DITATocTemplate3:

```
<toc-template >
  <title>Table of content – topics (cards) sorted according to numbers</title>
  <tocGroup>No2</tocGroup>
  <qualifier task-category="all" concept-category="all" reference-category="all"
product="No2" order="Descending" title="topic"/>
  <tocGroup>No3</tocGroup>
  <qualifier task-category="all" concept-category="all" reference-category="all"
product="No3" order="Descending" title="topic"/>
  <tocGroup>No4</tocGroup>
  <qualifier task-category="all" concept-category="all" reference-category="all"
product="No4" order="Descending" title="topic"/>
  ...
  <tocGroup>No10</tocGroup>
  <qualifier task-category="all" concept-category="all" reference-category="all"
product="No10" order="Descending" title="topic"/>
</toc-template >
```

DITATocTemplate4:

```
<toc-template >
  <title>Table of content – topics (cards) sorted according to images</title>
  <tocGroup>Jack</tocGroup>
  <qualifier task-category="all" concept-category="all" reference-category="all"
product="Jack" order="Descending" title="topic"/>
  <tocGroup>Queen</tocGroup>
  <qualifier task-category="all" concept-category="all" reference-category="all"
product="Queen" order="Descending" title="topic"/>
  <tocGroup>King</tocGroup>
  <qualifier task-category="all" concept-category="all" reference-category="all"
product="King" order="Descending" title="topic"/>
  <tocGroup>Ace</tocGroup>
  <qualifier task-category="all" concept-category="all" reference-category="all"
product="Ace" order="Descending" title="topic"/>
</toc-template >
```


Use cases: ditatoc temp markup (5)

Another feature is that it must be possible to define chapter-TOCs. A main TOC at the beginning of the manual list all chapter-TOCs and on which page they are. In each chapter a TOC is generated which lists all topics in that chapter (according to map hierarchy).

This is valid only for paper media.

```
<bookmap>
<frontmatter>
<booklist>
  <toc href="DITATocTemplateMain.ditatoc temp"/>
</booklist>
</frontmatter>
<chapter navtitle="My chapter title1">
  <toc href="DITATocTemplateChapter1.ditatoc temp"/>
  <topicref href="AceOfSpades.dita" navtitle=""/>
  <topicref href="AceOfHearts.dita" navtitle=""/>
  <topicref href="AceOfDiamonds.dita" navtitle=""/>
  <topicref href="AceOfClubs.dita" navtitle=""/>
</chapter>
<chapter navtitle="My chapter title2">
  <toc href="DITATocTemplateChapter2.ditatoc temp"/>
  <topicref href="KingOfSpades.dita" navtitle=""/>
  <topicref href="KingOfHearts.dita" navtitle=""/>
  <topicref href="KingOfDiamonds.dita" navtitle=""/>
  <topicref href="KingOfClubs.dita" navtitle=""/>
</chapter>
</bookmap>
```

DITATocTemplateMain:

```
< toc-template type="main">
  <title>Main table of contents</title>
  <qualifier reference-
category="ChapterTOC"
order="AccordingToTOCHierarchy"
title="toctemp"/>
</toc-template>
```

DITATocTemplateChapter1:

```
< toc-template type="chapterTOC">
  <title>Table of contents for My chapter
1</title>
  <qualifier task-category="all" concept-
category="all" reference-category="all"
product="all"
order="AccordingToMapHierarchy"
title="topic"/>
</toc-template>
```

Use cases: ditatoc temp markup (6)

It could also be possible to include the relations from the reltable in the TOC. A further feature in DITA would be the ability to type the topic relation (relation-type).

Another alternative is to use the navtitle or titlealts values.

You could even include the short description and other metadata in the TOC (valid for paper).

Result in table of content in paper manual:

No3	
Diamonds No3.....	2
<i>Related (done before): Diamond No2.....</i>	3
Hearts No3.....	4
<i>Related (done before): Diamond No2.....</i>	5
Clubs No3.....	6
<i>Related (parent): Ace of Clubs.....</i>	7
Spades No3.....	8
<i>Related (parent): Ace of spades.....</i>	9

TocTemplate:

```
<toc-template >
  <title>Table of content – topics (cards) sorted
according to numbers with topic relations</title>
  <tocGroup>No2</tocGroup>
    <qualifier task-category="all" concept-category="all"
reference-category="all" product="No2"
relation="accordingToReltable" order="Descending"
title="navtitle"/>
  <tocGroup>No3</tocGroup>
    <qualifier task-category="all" concept-category="all"
reference-category="all" product="No3"
relation="accordingToReltable" order="Descending"
title="navtitle"/>
  <tocGroup>No4</tocGroup>
    <qualifier task-category="all" concept-category="all"
reference-category="all" product="No4"
relation="accordingToReltable" order="Descending"
title="navtitle"/>
  ...
  <tocGroup>No10</tocGroup>
    <qualifier task-category="all" concept-category="all"
reference-category="all" product="No10"
relation="accordingToReltable" order="Descending"
title="navtitle"/>
</toc-template >
```

Use cases: ditatoc temp markup (7)

The TocGroup elements are used to group the entries in the TOC. The title for the group could either be typed or fetched from an attribute in DITA topics.

It must be possible to build a hierarchy of tocGroup elements. Each tocgroup element can have one or several qualifier elements and one or several tocGroup elements. An alternative is that each tocGroup has a subject element to hold the subject (no2 etc) and that tocGroup is not an empty element (close tag needed).

An alternative is to only have the qualifier element (or the tocGroup).

The ditatoc temp file must have a main title and maybe also a "tocmeta" element?

The ditatoc temp could also have a description element if the different subjects (defined by the tocGroup) needs explanation.

DITATocTemplate2:

```
<toc-template>
  <tocmeta>Maybe not needed</tocmeta>
  <title>Table of content – topics (cards) sorted according to type of card and component</title>
  <tocGroup navtitle="task-category=diamonds">
    <description>This section lists all topic that describes how to install a particular
component.</description>
    <subject>No2</subject>
      <qualifier task-category="Diamonds" product="No2" order="Descending" title="topic"/>
    <subject>No3</subject>
      <qualifier task-category="Diamonds" product="No3" order="Descending" title="topic"/>
    ...
  </tocGroup>
  <tocGroup navtitle="task-category=hearts">
    <subject>No2</subject>
      <qualifier task-category="Hearts" product="No2" order="Descending" title="topic"/>
    <subject>No3</subject>
      <qualifier task-category="Hearts" product="No3" order="Descending" title="topic"/>
    ...
  </tocGroup>
</toc-template>
```

Comparision: ditatoctemp and topicmap

- The tocGroup element in ditatoctemp is used to express what corresponds to topic type, topic, occurrence type, association type, association, role type and role in topicmap.
- Both topicmap and ditatoctemp can display the type of topic. The category attribute on DITA topic root/category element is used to further specify the DITA type (see other proposal).
- Topicmap can express relations between topics but not in ditatoctemp (if relations between tocGroup elements would be included...). Ditatoctemp can display relations between occurrences (reltable).
- The ditatoctemp could serve as the template for generating a topicmap from DITA content. In this case the ditatoctemp states the topics and topic types etc and relates the occurrences to the topics.

Contact/feedback

Jonatan.lundin@citec.se