empolis

ARVATO KNOWLEDGE MANAGEMENT

White Paper

# The Topic Maps Handbook

For further information, please contact info@empolis.com.

Acknowledgement: This text has been used by the Standards Australia handbook “XML Topic Maps for Knowledge Management”.

Author: Dr. H. Holger Rath

ABOUT THE AUTHOR

This handbook builds on the experience the author, Dr. H. Holger Rath has gathered conducting many topic map tutorials and workshops over the last few years. Dr. Rath is head of the German delegation for the ISO committee responsible for the topic map standard. He is co-editor of ISO’s Topic Map Query Language standard and chair of the OASIS Technical Committee ‘Vocabulary for XML Standards and Technologies’. As Head of Consulting at empolis, a content management and knowledge management solutions provider, he knows both the market requirements as well as the technical possibilities.

empolis GmbH

An der Autobahn 2

33311 Gütersloh

Germany

info@empolis.com

www.empolis.com



Version 1.1 – 2003-02-06

# Table of contents

# Preface

In January 2000, the International Organization for Standardization ISO published the standard ISO/IEC 13250 *Topic Maps*. Topic maps define a model for the semantic structuring of knowledge networks. Dubbed the 'GPS of the information universe', topic maps are a solution for organising and accessing large and continuously growing information pools. They provide a 'bridge' between the domains of knowledge management and information management.

A handbook about 'Topic Maps' should deliver more than just a technical description of the Topic Map standard. It should motivate the reader to understand why the standard was developed and how it can be used, and what the most important application domains are. The handbook should provide a simple but comprehensive example to introduce the main concepts of topic maps and the problem space topic maps address. At that point, the reader will be prepared for a detailed introduction into all concepts and features of the topic map paradigm – of course accompanied by lots of examples and useful remarks. After having learned all about the technology the reader will be eager to learn how to design and create topics maps and what the most important application domains are. A chapter about the whole area of topic map standardisation work should be part of a handbook about an ISO standard. A good handbook should also provide a pointer to the future of this exciting area.

The handbook is divided in seven chapters. Chapter 1 *Knowledge Management and the 'Info Glut'* describes the challenges of knowledge management in the information age, lists some 'historical' approaches to manage 'info glut', and explains why topic maps are needed. Chapter 2 *A simple example* uses a back-of-book index to introduce basic topic map concepts. Chapter 3 *The topic map paradigm* explains topic maps, gives syntax examples, and illustrates the application of these concepts in a topic map about the British Virgin Islands. Chapter 4 *How to create a topic map* presents the topic map cookbook – a systematic approach to topic map design – and hints for the manual as well as automatic generation of topic maps. Chapter 1 *Application scenarios* talks about the most important application domains and who is using the paradigm. Chapter 6 *The family of topic map standards* provides an overview about the available topic map and related standards, their history, and latest developments. Chapter 7 *Conclusions and outlook* finally risks taking a look into the crystal ball and predicts how the topic map future might evolve. Chapter 8 provides a *Glossary* of topic map terms and Chapter 9 a list of *Resources* consisting of literature references as well as consortia and committee names.

# 1 Knowledge management and the 'Info Glut'

## 1.1 Purposes of knowledge management

The main purpose of a *Knowledge Management* (KM) is to support decision making based on information and experience that exist in the organisation. Finding best practices and experts, supporting call centre agents or online shoppers, e-Learning, or just finding the information you are looking for are all typical KM activities. It is the task of KM to speed the take up of organisational learning and to increase the organisation's IQ. The 'knowledge cycle' consisting of knowledge recognition, capture, distribution, utilisation then looping back to re-capture. This is how organisation learning could result in a 'smarter' organisation with higher IQ – the rate at which organisations exploit the knowledge cycle is increasingly seen as a benchmark of company 'health'.

When we look at KM from the technology point of view different components leading to a KM capability can be identified. The 30,000 feet view shows two main components, *information management* and *information access*, accompanied by business connectors to other applications. Information management acts to ensure up-to-date, timely, complete, and consistent information. Intelligent information access ensures that information is available to the right person, to the right level of detail in a uniform and consistent manner.

At the 10,000 feet level information management and information access could be divided into sub-components.

- Information management separates out by the kinds of information managed: *data management* ('raw' data e.g., parts catalogue, product data), *asset management* (e.g., images, graphics), *document management* (monolithic documents e.g., MS-Word, scanned pages), *content management* (structured documents with access to document parts e.g., XML), *metadata management* (data about the information e.g., author, last update, access rights), *link management* (links between information objects), and *knowledge structure management* (explicitly coded knowledge e.g., classification schema, semantic net).
- Information access separates out along access method lines: *search* (on all kinds of information), *navigation* (in link networks, knowledge structures), and *notification* (whenever information is added to the KM system fulfilling a user profile).

The purpose of KM is providing a powerful tool to manage and to access the continuously growing amount of very diverse, highly interlinked, and complex information. KM is the tool to deal with the info glut.

## 1.2 Challenges of the information age

There is no doubt, we are living in the *information age* and we are part of the *information society*. Millions of people all over the world surf the Internet every day – for fun or seriously as part of their job. Large parts – if not the complete – economic system of the industrialised world relies and depends on electronic information stored and managed in computers and interchanged via networks. The amount of electronically available information doubles every two years. It is not a lack of information, it is over abundance of rich information that causes discomfort. It is the info glut, which increasingly is becoming a daily challenge.

The promise of the information society

*Deliver the right information to the right person at the right time.*

has turned to a threat under the avalanche of information that daily crowds our lives.

But what is giving rise to this avalanche? It is the simple fact that we have access to not just a single information resource on any given topic, but to many. Information on a given topic could reside in a company database records, company documents, parts of documents, web pages, images, videos, etc. coming from different sources and different repositories. Increasingly we also need to know the context in which the information resource is embedded to validate its relevance to our enquiry. We need pointers to related information as well. We want to know how the found resource fits in the larger picture. We want to extract knowledge from information in context.

This leads to a more refined objective for the information society:

*Deliver the right information* ***in the right context*** *to the right person at the right time.*

## 1.3 'Historical' approaches to manage info glut

*NOTE : The heading of this section is a little bit provocative to show that many concepts used in 'modern' information technology are in fact quite old. Despite their antiquity they are still in regular use.*

Let us investigate how humans tried to manage the 'info glut' challenge in the past – before computers were available. Information was collected in physical containers called books and books were collected in even larger physical containers called libraries. To locate specific information on a specific topic, we needed to access special kind of information that told us about the books and the kind of information they contained. A library catalogue helped us to find, locate, the books on the shelves related to our topic of interest. To actually retrieve, locate, the needed text portion we used one, often a combination, of four different paradigms: following the hierarchical structure of the table of contents, reading the text from beginning to end, browsing through the pages, and looking up the back-of-book index.

### 1.3.1 Catalogues and subject classification

The library catalogue is a collection of cards (= metadata records) about every book in the library. A metadata record describes the typical attributes of a book. This set of attributes is the result of many years of research in the library sciences community. Typical attributes are book title, author, publisher, publication date, ISBN, and location in the library (probably using some variation of the Dewey or Library of Congress system).

**Example:** Catalogue attributes of this handbook are "Title = The Topic Maps Handbook", "Author = H. Holger Rath", "Publisher = empolis GmbH", "Year of publication= 2003".

An other important attribute is the classification code assigned to the book. The code addresses a node in a (sometimes large) subject classification schema and states as such where the book is about (= which subjects are covered by the book).

**Example:** The topic map standard ISO/IEC 13250:2000 carries the ICS code 35.240.30 "IT applications in information, documentation and publishing" [7]. Code 35 stands for "Information technology, Office machines" and code 35.240 stands for "Applications of information technology".

### 1.3.2 Table of contents

The table of contents is a hierarchical organisation of the chapters and sections of the book with links (= the page numbers) to their locations. It can be seen as a specific kind of subject classification.

*NOTE : It is specific because a table of contents of one book should not be applied to any other book – maybe the highest levels of stereotype professional publications can attempt to have a table of contents that places the subject matter of this volume in the context of other volumes (e.g., legal publications, technical documentation, patents, or standards).*

A chapter or section heading describes roughly the subject of the following text. The nesting of the chapters, sections, sub-sections etc. defines the hierarchical structure – the classification – of the subjects provided in the text.

**Example:** If you want to know something about "sub-classing in topic maps" you could use the table of contents of this handbook and navigate down the path "The topic map paradigm" / "Classes and class hierarchies" / "Class hierarchies – Superclasses / Subclasses". After having found the appropriate heading the page number tells you where to found the related text.




empolis TRANSFORMING INFORMATION INTO VALUE

### 1.3.3 Back-of-book index

The back-of-book index is probably the most powerful paradigm to find, locate, certain information in a book. There are several reasons for this:

- The index is a collection of terms representing the relevant subjects explicitly or implicitly covered by the text.
- The index is a surrogate, a kind of semantic fingerprint of the book's content.
- The index is a result of an intellectual process in which a human (the indexer) selects only those subjects from the text which are – in their opinion – relevant for the target audience.

Whatever is important in a book should be found in its index. All unimportant 'noise' existing in the text has been filtered away. Searching for something in a book can be reduced to searching for it in the index. If it is not in the index you can be quite sure that it is not covered by the book. And if it is in the index, precise pointers to the pages (= page numbers) or sections (= section numbers) guide you to the information you are looking for. See Chapter 2 *A simple example* for an example of a back-of-book index.

Building an index is the task of the book's publisher or author and they get paid for their investment when we buy the book. But building an index for a company's repository with thousands or millions of online resources would be an expensive piece of work with probably no quick ROI. Information retrieval concepts tried to overcome this dilemma with automatically generated indices. The best known and most widely used are so called 'full text' indices. As the name 'full text' implies, a full text index contains all words of the text – except some stop words like "a", "the", "and", "or".

But there is a big difference between searching a purpose built back-of-book index and searching a full text index. Searching a purpose built index is much more rewarding than searching a full text index. The purpose built index is much more precise, because you search only relevant terms, whereas the search in a full text index looks at every word of the text – even at those producing the irrelevant 'noise' – and will therefore return the irrelevant hits mixed with the relevant ones. And even worse, you have to look at every hit before you recognise what is relevant and what is not. Probably everybody experienced the frustration about the enormous number of – useless – hits when performing a full text search on the web or an organisation's repository.

## 1.4 Why topic maps?

The previous sections identified the, historically, three most common forms of information location referencing in use today, catalogue reference, table of contents, and indexing. We also saw that a purpose built index offers a more structured solution to the finding of relevant topic matter than a full text index can. For KM the need is to have information presented in context. If we were looking for a paradigm on which to build an information locating system then the purpose built index would be our model of choice. A purpose built indexing approach, such as we would find in the back of a book, is an information locating paradigm that fits into the KM picture, brings information into a meaningful context, and supports various ways to organise information resources. As you probably guessed: topic maps are such a paradigm.

Topic maps are designed to manage the info glut, build valuable information networks over any kind of information resources, and enable the structuring of unstructured information. A topic map can be seen as an electronic super index, implementing the back-of-book index paradigm and much more.

Topic maps are various technologies in one:

- Complex metadata: A topic map contains information about information resources. It is not part of the information resources, it is created, managed, and stored separately from the resources, but could be closely connected to them.
- Search index: As searching in a back-of-book index is a very precise searching method, searching in a topic map provides better search results as searching in a full text index. A topic map can be seen as an intelligent search index.
- Link network: Topic maps are well-organised link networks helping to avoid the 'lost-in-hyperspace' syndrome.
- Knowledge structure: Topic maps are a base technology for explicit knowledge modelling and knowledge navigation – hence their value to Knowledge Management.

You might argue that comparable technologies, like Semantic Networks , follow similar approaches but did not succeed in the past. So, why should topic map be more successful? There are many arguments why topic maps 'will make it' – here are the most important ones:

- The Internet provides both the infrastructure to deliver information access to everybody's desktop as well as the required organisation of large amounts of information – both not existing when Semantic Networks were developed. So, the world is ready for topic maps – the problems they solve and the infrastructure they need are in place.

- Topic maps are not a proprietary solution but an international ISO standard describing electronic indices, classification schemas, knowledge structures, that are Web enabled, and ready to use in the Internet and Intranets.

- Topic maps are not an academic knowledge representation paradigm. They are built by practitioners, focussing on real-world requirements and on implementable solutions.

- Topic maps not only cover the meta layer above the resources, they are also providing a powerful mechanism to connect resources within the meta layer. They are the bridge between Information Management and Knowledge Management.

### 1.4.1 *Topic map slogans*

Charles F. Goldfarb, the inventor of SGML and father of mark-up languages branded topic maps the *'GPS of the information universe'*. As the Global Positioning System helps avoid getting lost in physical space and is able to guide you to a target point, a topic map says where you are in your information space and where to go to find what you are looking for. As the GPS satellites send down their signals to the GPS devices showing the coordinates and the way to go, the topic map 'sends' down its 'signals' to the topic map 'device' showing the user what he is currently looking at and where he could go to locate specific or related information.

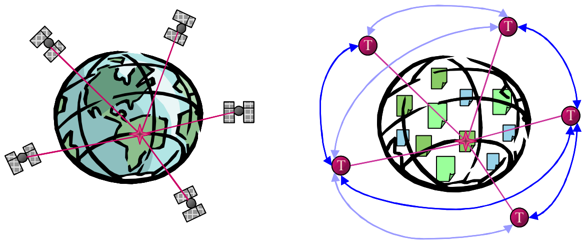


**Figure 1** GPS of the information universe.

Topic maps are the next logical step towards a semantic web. As unified addressing of resources by the Hypertext Transfer Protocol (HTTP) [23] and universal data by the eXtensible Markup Language (XML) [22] are considered the 1st and 2nd revolutions, topic maps are prophesied as the *3rd revolution*.

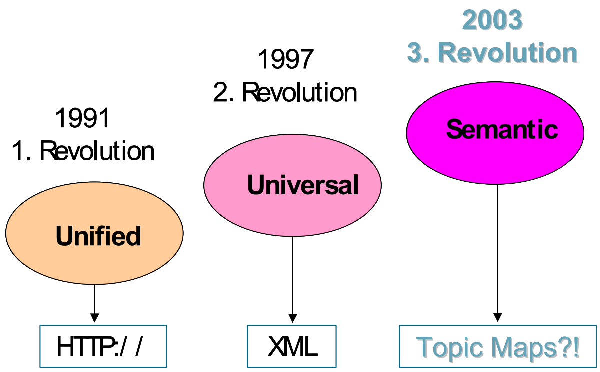



**Figure 2** The 3rd revolution.

### 1.4.2 *Analysts forecast topic maps a bright future*

Both the Gartner Group [4] and the META [14] Group have published reports about topic maps and described them as an 'emerging' technology for content navigation, metadata, and knowledge management. They recommend that 'enterprises that are faced with significant challenges in the organization of information should track the evolution of topic mapping technology.' (Gartner) and 'G2000 users should monitor' (META). Gartner Group predicts – with a 0.7 probability – that 'topic maps will become a mainstream technology by 2003'. META Group foresees that 'the XTM market segment to grow 3x-5x per year, reaching a market value of $1B+ by 2005-07 (tools and services)'.

# 2 A Simple Example

The simple example presented in this chapter is a – fictitious – back-of-book index of the travel guide “Guide to the British Virgin Islands”. A back-of-book index was selected as example because the topic map paradigm was invented to model back-of-book indices electronically (see section 6.1 *Brief history of topic map standardisation*).

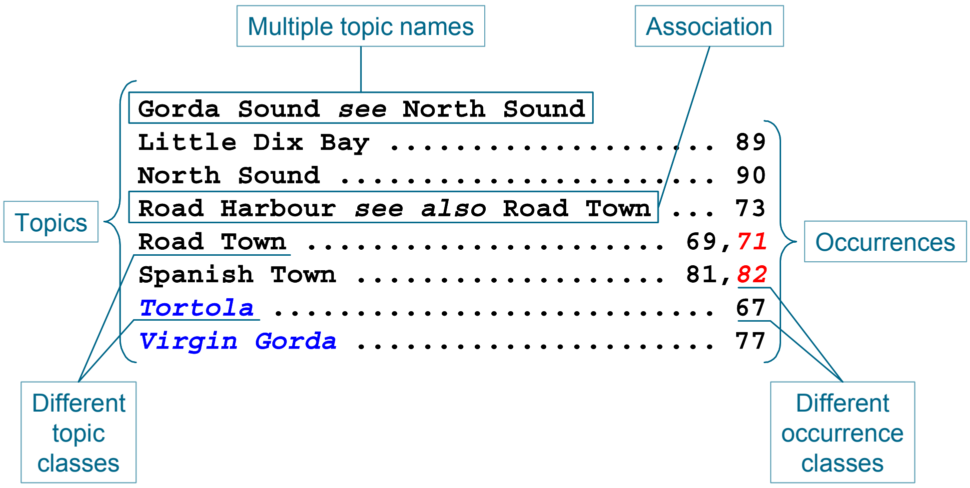



**Figure 3** Back-of-book index example introducing the basic topic map concepts.

The index entries are *topics* representing the concepts (*subjects*) covered by the guide that are relevant to the target audience and selected by a human in an intellectual process. For the reader’s convenience the topics carry a human readable *name*.

The page numbers are pointing to the pages (*resources*) containing relevant information about the topic. These are the *occurrences* of the topic.

Different formatting of topic names (e.g., regular font and *italic blue*) help to differentiate the various kinds of topics (*topic classes*). Topic classes provide a classification to simplify the finding of topics. As our example is about the British Virgin Islands it makes sense to highlight the islands in the index.

Different formatting of page numbers (e.g., regular font and *italic red*) help to distinguish between various kinds of occurrences (*occurrence classes*). Occurrence classes provide some ‘hints’ about the kinds of resources the reader can expect when following the occurrence link. The page numbers 71 and 82 point to city maps of Road Town and Spanish Town; all other page numbers might point to arbitrary text based kinds of resources.

The “see” relationship illustrates *multiple topic names* and indicated that these are just different names (synonyms) for the same subject, which is represented by the topic. Gorda Sound and North Sounds are two names for the same bay – the physical thing – in the north of Virgin Gorda.

The “see also” relationship is different – even if it looks quite similar. A “see also” relationship expresses an *association* between topics. We do not know the kind of association – if you know the British Virgin Islands you would know that Road Harbour is the harbour of Road Town – we just know that they are somehow related and that we might also take a look at the topic Road Town when we are interested in Road Harbour. Associations found in a topic map can be instances of a certain declared class (e.g., the association class “is harbour of city”).

Good printing practice prohibits the use of many different formatting styles in a printed index. But a topic map could have as many topic classes, occurrence classes, and association classes as required by the application.

These are the basic concepts of the topic map paradigm. We were able to easily derive them from a simple back-of-book index:

- *Topics* represent *subjects* and carry *names*.
- *Occurrences* of a topic point to relevant *resources*.

- *Associations* connect related topics.
- *Topic classes, occurrence classes*, and *association classes* help to distinguish different kinds of topics, occurrences, and associations respectively.

But there are further requirements, which cannot be fulfilled by a printed back-of-book index but would improve the usability of an electronic index. Here are some of those requirements:

- The names of a topic should consider the display device (e.g., full name on computer screen, name with some abbreviations on PDA, name with lots of abbreviations on WAP phone).
- Users would like to view only a certain part of the index (e.g., the topics satisfying the user's interest).
- An index should be multilingual and users should be able to select the language they are able to understand.
- An index should cover more than one set of resources at the same time (e.g., PDF pages of a book, web portal, image database, news) and users are able to select the kinds of resources they want to see.
- Two indices with different origin but about the same domain should be easy to merge.

The topic map paradigm covers these requirements with the *concepts variant names, scopes, subject identifiers,* and *merging*. The next chapter fully introduces the topic map paradigm.

# 3 The Topic Map Paradigm

The previous chapter explained the basic topic map concepts by looking at a simple example of a back-of-book index. This chapter introduces the topic map paradigm standardised in ISO/IEC 13250:2000 [10] more systematically. It also gives syntax examples using the XML Topic Map (XTM) format [21].

*NOTE : The XTM specification [21] is the explanatory text of the XTM DTD [20]. It will be incorporated into the next edition of ISO/IEC 13250. See Chapter 6* The family of topic map standards *for further discussion on the evolution of this standard.*

## 3.1 Subjects are the starting point

We learned already that a *subject* is a concept, notion, idea etc. that is worth of becoming a topic in a topic map. The topic map standard does not predefine candidate subjects. Quite the contrary, the definition of *subject* in ISO/IEC 13250:2000 is as general as it could be:

*In the most generic sense, a 'subject' is any thing whatsoever, regardless of whether it exists or has any other specific characteristics, about which anything whatsoever may be asserted by any means whatsoever.*

*ISO/IEC 13250:2000*

The generality of the definition enables topic maps to be applied to any application domain you can think of. It is up to the topic map author to identify the subjects, which are relevant for the application and should be represented in the topic map.

If a subject is relevant for the topic map and should therefore become a topic depends on:

- Application domain: The application domain defines what the topic map is about and has great impact on the selection of subjects. Since knowledge is 'infinite' the application domain has to be limited, bounded, to become manageable.
- User of the topic map: A topic map is – normally – built for the users. Consequently their requirements should be covered and these requirements might differ depending on the user's perspective, context, view, knowledge about the application domain. Careful user requirements analysis help to

establish the appropriate subjects that should appear in the topic map.

- Author of the topic map: A topic map is currently the result of an intellectual process performed by a topic map author. The topic map author is the final arbiter of what is in the map. The selections made depend on his individual knowledge about the application domain and the requirements of the user.

Some examples show how the subjects depend on the target audience. Again, the application domain is the “Guide to the British Virgin Islands”:

- Subjects of user “regular tourist”: hotels, restaurants, beaches, day trips, shopping sites, water sports, weather.
- Subjects of user “sailor”: marinas, anchorages, provisioning sites, currents, weather, winds, reefs, ship wrecks, charter bases.
- Subjects of user “scuba diver”: dive centres, fishes, corals, dive spots, reefs, ship wrecks, currents, water temperatures, visibility.

It is obvious that these lists differ in some subjects but also have some subjects in common. But even if the lists have some subjects in common the interest of the target audience in that subject might be very different. A good example is the subject “RMS Rhone” (a famous ship wreck): a sailor has to know how much water is above the wreck in low tide, whereas the scuba diver wants to know how deep the lowest part of the wreck is in the water and how many corals or fishes are around the wreck.

## 3.2 Building blocks: topics, names, occurrences, associations

The previous section explained that subjects are a result of an intellectual process performed by topic map authors to model information from resources and knowledge of people. It is essential to understand that subjects are the ‘things’ we, the humans, have in mind when we start building a topic map. And these ‘things’, the subjects, are outside of the computer with all its software, files, databases, and networks – the computer does not ‘know’ the subjects. E.g., all the islands, bays, towns, ship wrecks, and reefs of the British Virgin Islands are physically located somewhere in the Caribbean, but they are not inside someone’s computer. They are not part of the topic map; they are concrete parts or abstract concepts of our world.

This section explains how to bring these subjects ‘inside the computer’ and how to embed them in explicitly coded knowledge models, in topic maps.

### *3.2.1 Topics – computerised subjects*

A topic in a topic map represents a subject inside the computer. Whenever a topic is created representing a subject, the subject becomes through the topic a machine interpretable ‘object’. Once in this form further assertions can be made about it in an explicitly coded electronic form, the topic map. This was not possible for the subject, the ‘thing’, itself – because it was outside of the computer.

*NOTE : This explanation sounds rather philosophic and academic – and to a certain degree it is both. But for proper understanding of topic maps the distinction between subjects and topics is very important. The distinction makes it very clear that there is a difference between the original ‘thing’, the subject, and its representation in a computer, the topic. It enables us to narrow down the vast amount of human knowledge to something that is explicit and machine processable.*

Technically speaking, a topic reifies a subject – makes an abstract concept concrete. Reification is the process of creating a topic for a subject.

*NOTE : One might argue that subjects, which are already ‘inside a computer’ (e.g., the web site “*http://www.bviwelcome.com/*”), are already a machine interpretable ‘object’. But the number of subject outside computers is much greater than the number of subjects inside computers. Therefore we should treat all subjects equally and create topics even for the ones, which are already in computers. Section 3.6* Addressable and non-addressable subjects *gives further explanations on this issue and describes how topics can be bound to their subjects.*

A topic has three *characteristics*:

- Names (or more precisely *base names*): Human readable name for the subject. A topic can have zero, one, or many names.
- *Occurrences*: Link to information resources relevant for the topic.
- Playing roles in associations: A topic may be associated with other topics.

A topic can be an instance of zero, one, or more classes. These classes are also topics. For internal bookkeeping reasons, every topic has a – unique – identifier used for addressing purposes. Figure 4 shows some topics, their names, and classes.

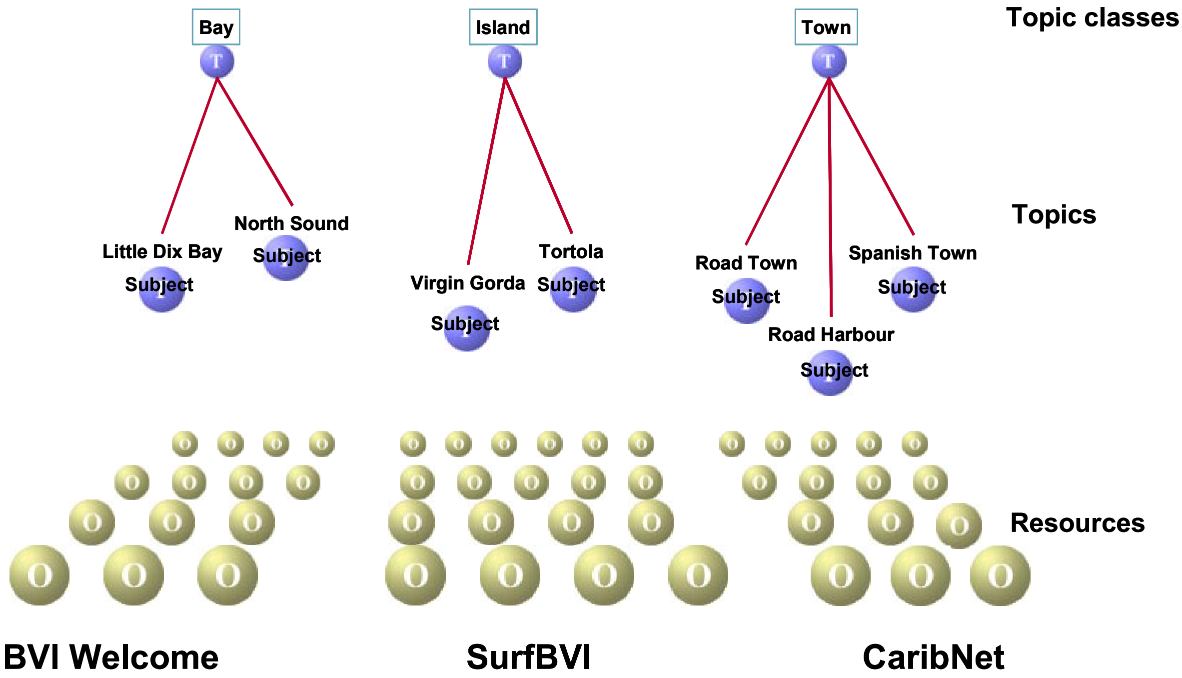



**Figure 4** Topic examples.

Syntax of a topic map with two topics:

```
<?xml version="1.0"?>

<!DOCTYPE topicMap PUBLIC
"-//TopicMaps.Org//DTD XML Topic Map
(XTM) V1.0//EN"
"http://www.topicmaps.org/xtm/1.0/xtm1.
dtd">

<topicMap id="british-virgin-islands"
xmlns="http://www.topicmaps.org/xtm/1.0
/"
xmlns:xlink="http://www.w3.org/1999/xli
nk">

  ...

  <topic id="tortola">
    <instanceOf>
      <topicRef xlink:href="#island"/>
    </instanceOf>

    ...

  </topic>

  <topic id="island"> ... </topic>

  ...

</topicMap>
```

### 3.2.2 *Names – talking to topics*

Names (base names) let humans talk about topics. Base names are one of the three topic characteristics. Without a name a topic would only have an id and that is not appropriate for human communication. Because a topic can have multiple names, we can use them for various purposes – mainly to list synonyms and name translations in different languages or dialects of the subject.

*NOTE : The standard defines the Topic Naming Constraint (TNC), which constraints the use of base names. See section Figure 15* The topic naming constraint (TNC) *for further details.*

Some base name examples:

- “Gorda Sound”, “North Sound” (synonyms)
- “Virgin Islands”, “Jungferninseln”, “Iles Vierges” (translations)

- "Taxi", "cab" (dialects)
- "Radio set", "walkie-talkie", "VHF" (technical terms)

Base name syntax:

```
<topic id="north-sound">
  <baseName>
    <baseNameString>North Sound
    </baseNameString>
  </baseName>
  <baseName>
    <baseNameString>Gorda Sound
    </baseNameString>
  </baseName>
</topic>
```

### *3.2.3 Occurrences – pointing to relevant information resources*

Occurrences bind relevant information resources to topics. Whenever a resource provides information about a topic it should be considered to become an occurrence of the topic. Occurrences are one of the three topic characteristics.

An occurrence can be an instance of one class, but it does not have to be. The class is a topic. Please note, that in contrast to topics occurrences must not be instances of multiple classes. Figure 5 shows some occurrences and their classes.

Technically speaking, an occurrence is either a resource reference that connects the topic with the resource or a string value (resource data) that is the resource.

A resource reference links the relevant resource to the topic using a notation called XLink/XPointer [28][29] URI (Universal Resource Identifier) [6]. This works similar to HTML hyperlinks in the web. Consequently, all resources addressable by a XLink/XPointer URI could be become occurrences in a topic map – means everything you can address in the Internet can become a topic's occurrence.

A resource data occurrence assigns a value (expressed as character string) to the topic. The class of the occurrence denotes the kind of the value. This can be used to assign meta data (property-value pairs) to topics and to subjects (because topics represent subjects).

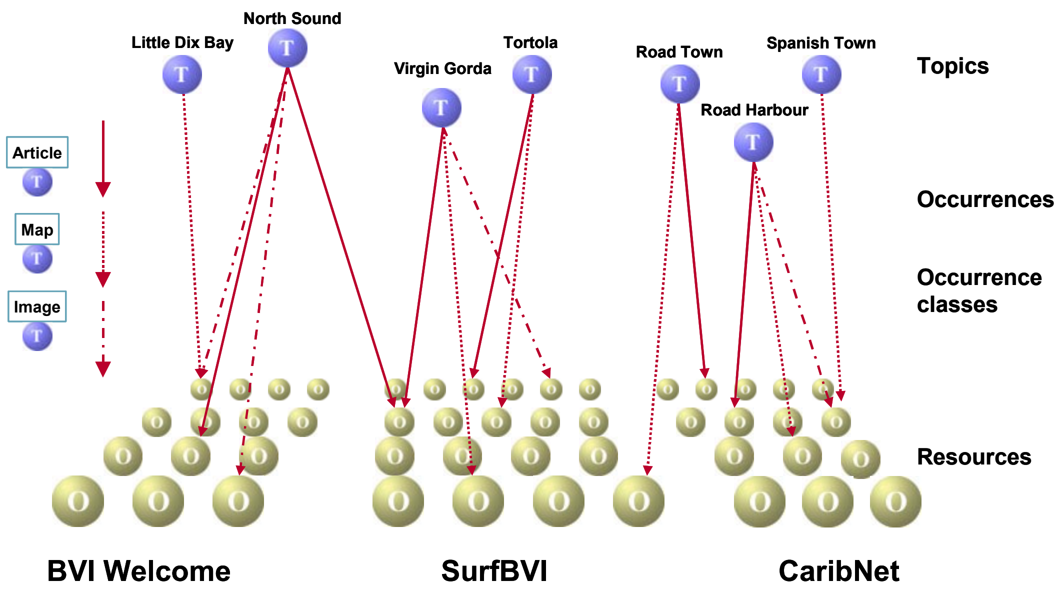
NOTES:

1. *The topic map standard does not pre-define any data types for resource data occurrences. The value is 'only' a character string. The interpretation of the string is up to the application.*
2. *Resource data occurrences model the facet concept of the SGML/HyTime notation for topic maps.*

Some occurrence examples for the topic "Tortola":

- "http://bviwelcome.com/intro.html#tortola" is an "article" about Tortola (resource reference occurrence).
- "14,000" is the number of "inhabitants" of Tortola (resource data occurrence).

**Figure 5** Occurrence examples.



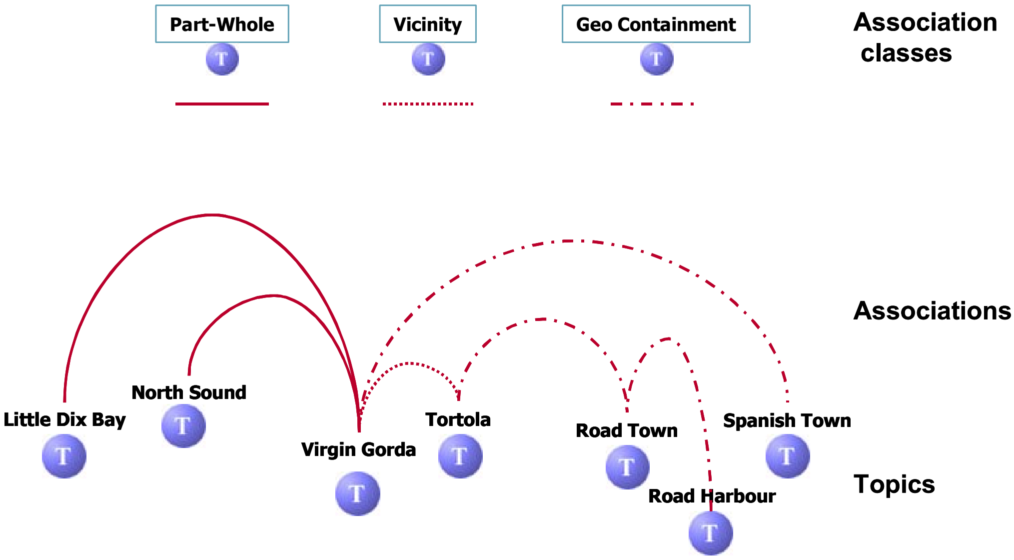



**Figure 6** Association examples (roles hidden for readability reasons).

Occurrence syntax:

```
<topic id="tortola">
  <baseName>
    <baseNameString>Tortola
    </baseNameString>
  </baseName>
  <occurrence>
    <instanceOf>
      <topicRef xlink:href="#article"/>
    </instanceOf>
    <resourceRef xlink:href=
"http://bviwelcome.com/intro.html#torto
la"/>
  </occurrence>
  <occurrence>
    <instanceOf>
      <topicRef

xlink:href="#inhabitants"/>
    </instanceOf>

<resourceData>14,000</resour
ceData>
  </occurrence>
</topic>
```

### 3.2.4 *Associations – as we may think*

Associations provide the context information necessary to better understand a topic – as the see also relationships in back-of-book indices do. Associations simulate the way humans think and as such are essential for knowledge modelling.

They establish relationships between topics – and only between topics. The number of topics related by one association is not limited – it could be only one topic (rather academic), two topics (so called binary associations, which are the most common ones), three topics, as many as the application requires.

An association can be an instance of one class, but it does not have to be. The class is a topic. Please note, that in contrast to topics, associations must not be instances of multiple classes. Figure 6 shows some associations and their classes.

Associations specified by the topic map paradigm do not imply a direction of the relationship. Associations are assertions, statements, which are valid independently from the direction you traverse them (an example is given below). You can look at the association from all involved topics – it always expresses the same assertion. But how do we know, what roles the topics play in the relationship? Figure 6 shows some binary associations. One relates "Little Dix Bay" and "Virgin Gorda" by a "Part-Whole" association. We cannot derive from the figure if "Little Dix Bay is-part-of Virgin Gorda" or vice versa, because we cannot obtain which topic is the "part" and which is the "whole" in the association.

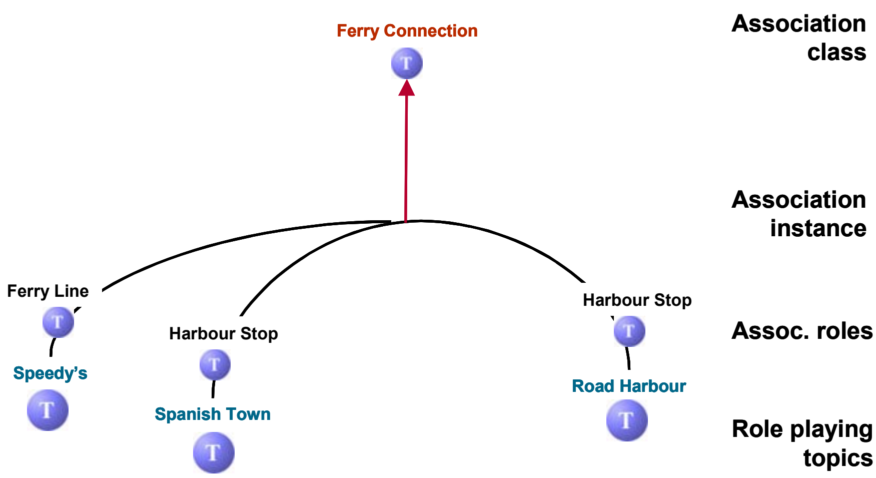



**Figure 7** Example showing a ternary association and its roles.

The concept of association roles provides the missing piece. Figure 7 illustrates that the association roles (“Ferry Line”, “Harbour Stop”) placed between the topics (“Speedy’s”, “Spanish Town”, “Road Harbour”) and the association (“Ferry Connection”) explain what the topics ‘are doing’ in the association. The topics play certain roles in the association – they are role playing topics or for short role players. That a topic plays a role in an association is one of the three topic characteristics. Association roles are topics, like class are as well.

Now we are able to derive the assertion that “Speedy’s is the ferry line providing a ferry connection between the harbour stops Spanish Town and Road Harbour” or “Spanish Town is a harbour stop in the ferry connection provided by Speedy’s to the harbour stop Road Harbour” or “Road Harbour is a harbour stop in the ferry connection provided by Speedy’s to the harbour stop Spanish Town”. All three sentences are verbal serialisations of the same association – the same logical assertion – illustrating that the association itself has no direction.

But direction becomes important when we want to present the association to a human reader because human language is a serialisation of words and the order has high impact on the meaning. Serialisation of binary associations is simple – at least in English and similar languages – but automatic serialisation of associations with three or more role players becomes difficult if not impossible. That is another reason why binary associations are so widely used.

Association syntax:

```
<topicMap ... >
  ...
  <association>
    <instanceOf>
      <topicRef xlink:href=
        "#ferry-connection"/>
    </instanceOf>
    <member>
      <roleSpec><topicRef xlink:href=
        "#ferry-line"/>
      </roleSpec>
      <topicRef xlink:href="#speedys"/>
    </member>
    <member>
      <roleSpec><topicRef xlink:href=
        "#harbour-stop"/>
      </roleSpec>
      <topicRef xlink:href=
        "#spanish-town"/>
    </member>
    <member>
      <roleSpec><topicRef xlink:href=
        "#harbour-stop"/>
      </roleSpec>
      <topicRef xlink:href=
        "#road-harbour"/>
    </member>
  </association>
  ...
</topicMap>
```

## 3.3 Classes and class hierarchies

Topic class, occurrence class, and association class were already introduced in the previous section. This section focuses on some background information about classes and how class hierarchies can be built in topic maps.

### 3.3.1 Classes – organising principles

A general definition of the term class:

*A group, set, or kind sharing common attributes.*

*Merriam-Webster’s Collegiate Dictionary*

We learned that topics, occurrences, and associations might be instances of classes. We also learned that all these classes are topics. Which might, again, be instances of other classes. This means that a topic could be class and instance at the same time.

The fact that all classes are regular topics also implies that a topic cannot be identified as class as long as there is not at least one instance of this class. What does this mean for topic maps? Think about a nearly empty topic map containing just one topic “Island”. As long as there is no topic e.g., “Tortola” defined as an instance of an “Island”, we do not know that “Island” is supposed to be a topic class.

Ideally, a topic map authoring system should present the author with a list of all available topic classes whenever a new topic is to be created. However, such a software could only present the classes, those topics that had already been used as classes – the system can only guess ‘sleeping’ classes. This and other requirements initiated the standards project Topic Map Constraint Language (see section 6.3.2 *ISO/IEC 19756 – Topic Map Constraint Language TMCL*).

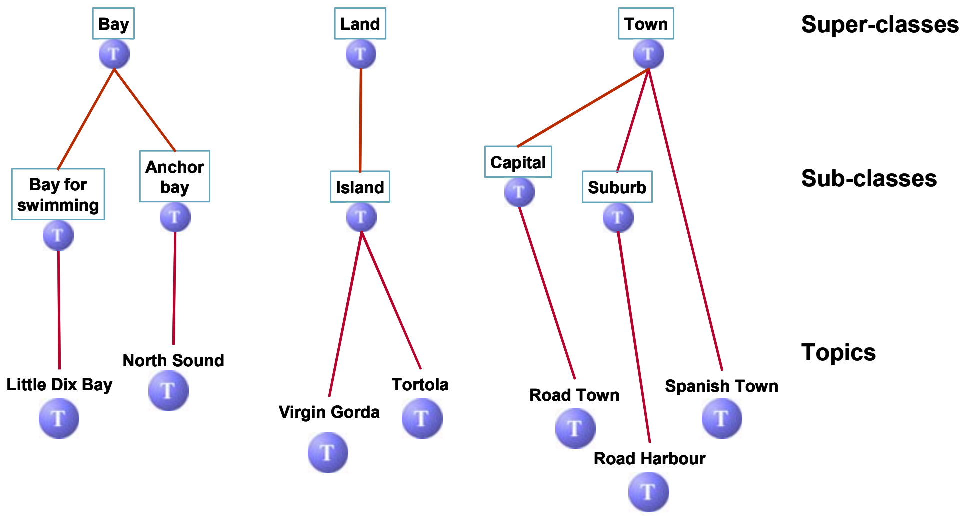



**Figure 8** Class hierarchy example.

*NOTES:*

1. *The ISO/IEC 13250:2000 standard uses the term type instead of class, which is used in the XTM specification. Type and class can be used synonymously. There is no difference between them in the topic map paradigm.*
2. *The topic map standard does not predefine any classes. It provides a framework to define application specific classes.*

### *3.3.2 Class hierarchies – superclasses / subclasses*

A class hierarchy consists of classes hierarchically organised by the superclass-subclass relationship. Figure 8 shows that “Bay” has subclasses “Bay for swimming”, “Anchor bay” and that “Island” has the superclass “Land”. A typical application of a class hierarchy is a taxonomy or subject classification schema.

The superclass-subclass relationship is defined in terms of an association with predefined class “superclass-subclass” and roles “superclass”, “subclass” (see Figure 9). The class “superclass-subclass” and the roles “superclass” / “subclass” are defined as standardised Published Subjects (see section 3.8 *Published Subjects* for further details), which are application independent.

*NOTE : In addition, the class-instance relationship is alternatively declared as association with predefined class (“class-instance”) and roles (“class”, “instance”), which are also standardised as Published Subjects. The <instanceOf> element can be seen as syntactical shortcut for an unconstrained “class-instance” association. Having a class-instance association opens up the scoping possibility, which is not available for the <instanceOf> element (see next section 3.4* Modelling viewpoints with scope*).*

Topics might be superclass and subclass at the same time (e.g., “restaurant” in the class hierarchy “Eating place – Restaurant – Gourmet temple” or “Meal” in the class hierarchy “Food – Meal – Five course menu”). Class hierarchies can be built for topic classes, association classes (e.g., “Eating place provides Food – Restaurant serves Meal – Gourmet temple celebrates Five course menu”), and occurrence classes (e.g., “Image – Photograph – Food photography”).

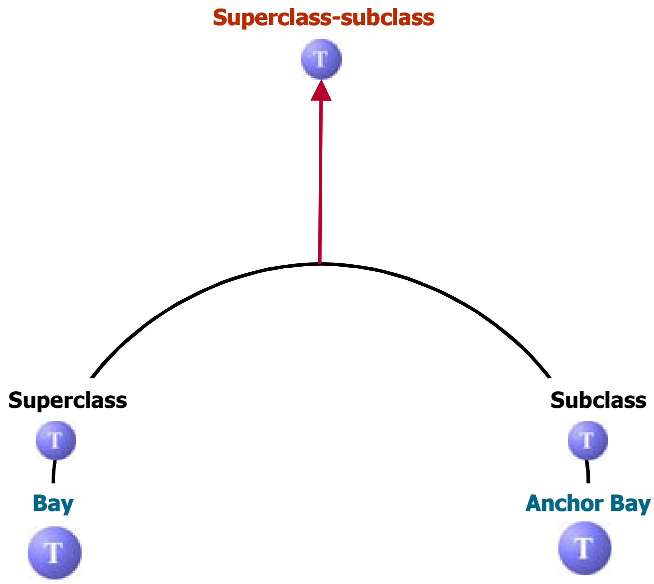



**Figure 9** Superclass-subclass association example.

Class hierarchy syntax:

```
<association>

  <instanceOf>
    <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.
xtm#superclass-subclass"/>
  </instanceOf>
```

```
  <member>
    <roleSpec>
      <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.
xtm#superclass"/>
    </roleSpec>
    <topicRef xlink:href="#bay"/>
  </member>

  <member>
    <roleSpec>
      <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.
xtm#subclass"/>
    </roleSpec>
    <topicRef
      xlink:href="#anchor-bay"/>
  </member>

</association>
```

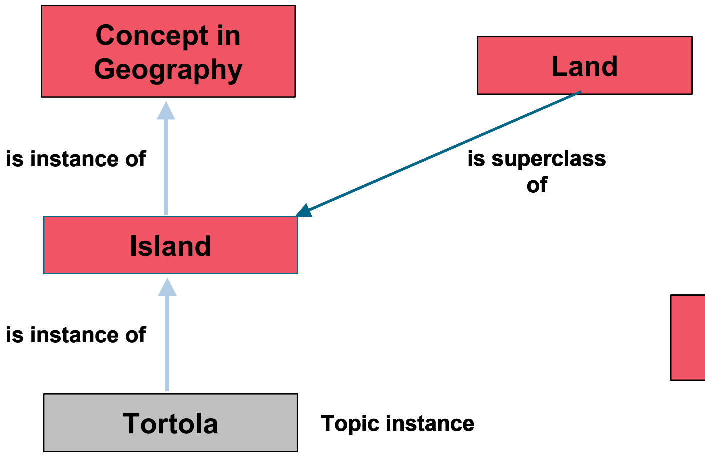



**Figure 10** Example of a topic being class, instance, and sub-class

#### 3.3.3 *Class-instance vs. superclass-subclass*

We have to distinguish carefully between the class-instance relationship and the superclass-subclass relationship.

Figure 10 shows the topic "Island". It is three different things at the same time. It is:

- instance of class "Concept in Geography",
- class for its instance "Tortola", and
- subclass of superclass "Land".

It is interesting to see the implications on "Tortola", the instance of "Island", illustrated in figure 11. It is perfectly right to say that "Tortola" is an instance of class "Land", but it is wrong to say that "Tortola" is an instance of class "Concept in Geography".

What can we derive from this example? The superclass-subclass relationship is transitive. It propagates properties of the superclass down to its subclasses and their sub-subclasses. Therefore all instances of a class are also instances of all the superclasses of its class. The class-instance relationship is not transitive.

It is a good test to check whether it is a class-instance or superclass-subclass relationship to assume that the instances of the lower class is also an instance of the highest class. When the assumption holds, it is the superclass-subclass relationship. If the assumption is nonsense, it is the class-instance relationship.

## 3.4 *Modelling viewpoints with scope*

The concept of *scope* was added to the topic map paradigm to deal with the fact that there is rarely one view of the 'world' – the application domain. Different people expect different assertions about different subjects. The reasons why different views are needed are very diverse.

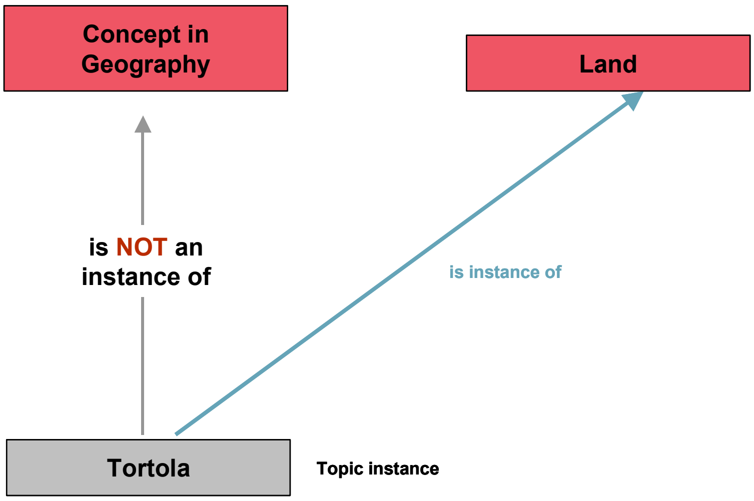



**Figure 11** Logical implications of the difference between class-instance and superclass-subclass

Typically, scopes are used to model:

- Languages: Names and occurrences are scoped by the language they are in. This could be real languages (e.g., "English", "German", "Japanese") or community dialects (e.g., "Sailor language", "Regular people language").
- Access rights: Occurrences and association provide access to further information, to resources or other topics. Scopes on such occurrences or associations can declare the level of

confidentiality or which user groups have the right to access this information (e.g., “Confidential”, “Public” or “Registered user”, “Unregistered user”).

- Views: Associations as well as occurrences provide context information about a given topic. Scopes can declare in which contexts this further information is meaningful. The context could be:
  - the skill levels of the user (e.g., “Expert”, “Intermediate”, “Beginner”) leading to different further information most appropriate for the know how level of the user;
  - the various interests of the users in the application domain (e.g., “Politics”, “Geography”, “Culture”, “Sports”);
  - the effectivity/validity of an assertion (e.g., immigration regulations depend on where the visitor is coming from, hotel rates depend on the season).

Scopes help to filter the ‘noise’ in large topic maps and allow us to concentrate on the interesting parts. They help to build semantic slices through the topic map. They can also help to resolve name conflicts (see section Figure 15 *The topic naming constraint (TNC)*).

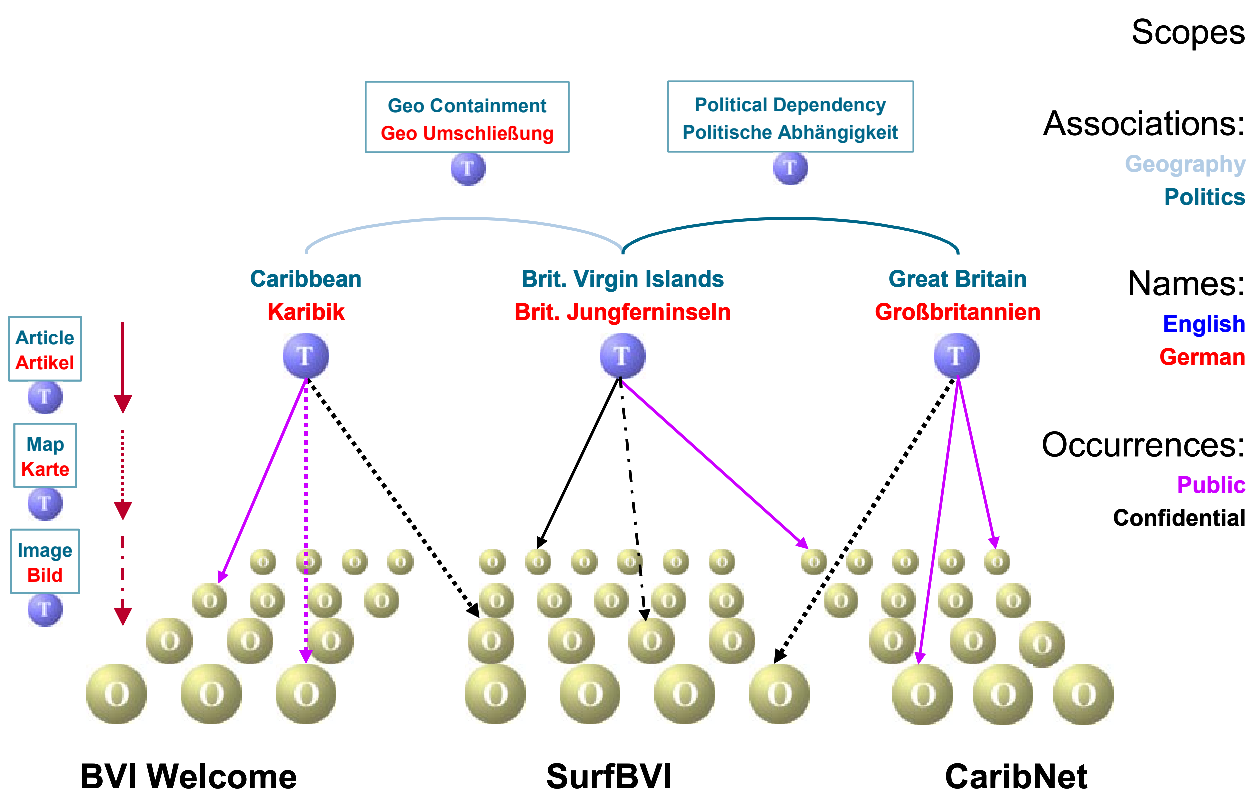
Technically, a scope specifies the extent of the validity of a topic characteristics (= base name, occurrence, association). A scope is a set (*scope set*) of topic references. A topic referenced in a scope set is called a scoping topic or a theme. If no explicit scope set is assigned to a topic characteristic the unconstrained scope is the automatic default stating that the characteristic is always valid.

The proper interpretation of the scope sets is up to the application software and its business logic. The standard neither predefines any scopes nor any interpretation of scope sets.

NOTE: *Only the topic characteristics can be scoped, not the topic. The scope attribute of the <topic> element in the HyTM syntax is just a syntactical shortcut for a topic with all its characteristics in the same scope.*

Figure 12 shows various scope sets assigned to the names, occurrences, and associations of an example topic map. Names are available in two languages (themes “English”, “German”), occurrences carry access right information (themes “Public”, “Confidential”), and associations declare which field of interest they

**Figure 12** Scope examples



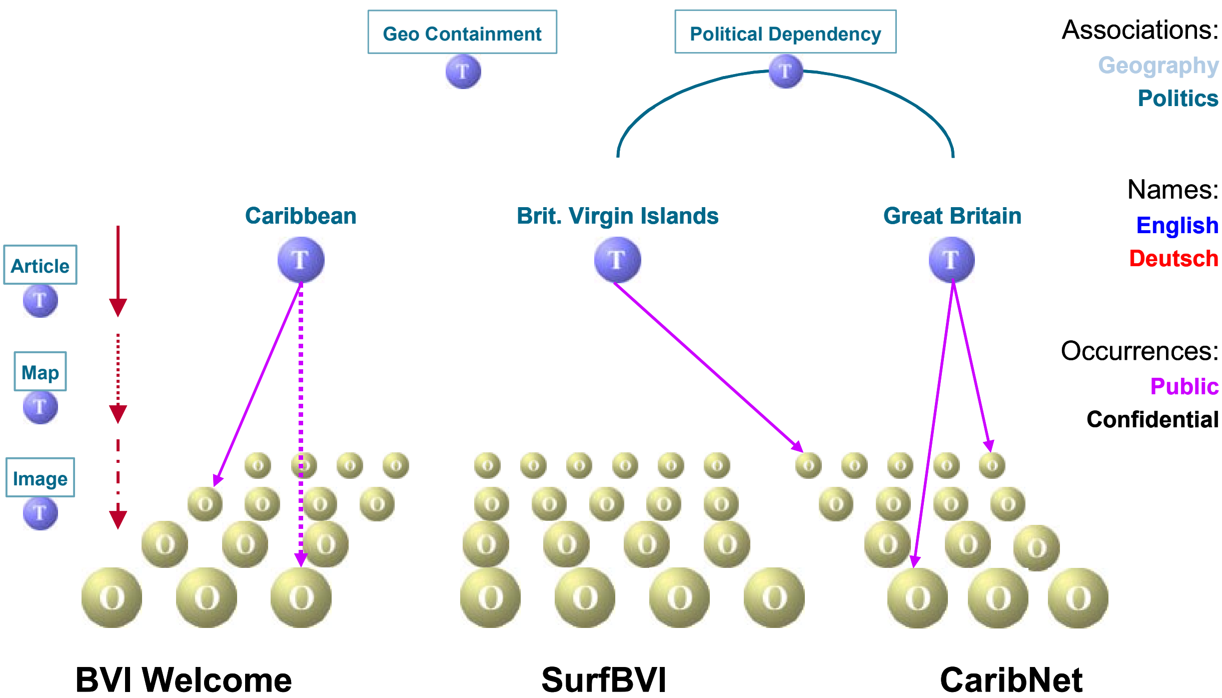



**Figure 13** Example for topic map with applied scope setting “English or Public or Politics”

cover (themes “Geography”, “Politics”). Figure 13 shows the same example topic map with the applied scope setting “English or Public or Politics”. All topic characteristics not in one of these scopes are hidden from the user. Nevertheless, they are still part of the topic map.

Scope syntax:

```
<topic id="brit-virgin-islands">

  <baseName>
    <scope>
      <topicRef xlink:href="#english"/>
    </scope>
    <baseNameString>
      Brit. Virgin Islands
    </baseNameString>
  </baseName>

  <baseName>
    <scope>
      <topicRef xlink:href="#german"/>
    </scope>
    <baseNameString>
      Brit. Jungferninseln
    </baseNameString>
  </baseName>

  <occurrence>
    <instanceOf>
      <topicRef xlink:href="#article"/>
    </instanceOf>
    <scope>
      <topicRef xlink:href="#public"/>
    </scope>
      <resourceRef xlink:href=
"http://bviwelcome.com/intro.html"/>
    </occurrence>

</topic>

<association>

  <instanceOf>
    <topicRef
      xlink:href="#geo-containment"/>
  </instanceOf>

  <scope>
    <topicRef xlink:href="#geography"/>
  </scope>

  <member>
    <roleSpec>
      <topicRef
        xlink:href="#container"/>
    </roleSpec>
    <topicRef xlink:href="#caribbean"/>
  </member>

  <member>
    <roleSpec>
      <topicRef
        xlink:href="#containee"/>
    </roleSpec>
    <topicRef xlink:href=
      "#brit-virgin-islands"/>
  </member>

</association>
```

## 3.5 Variant names

Variant names are alternate forms of base names for display, sorting of topics, or other – application dependent – purposes.

Some variant name examples:

- Display name for topic “Little Dix Bay”:
  - 1024x768: http://www.littledixbay.com/images/ldblogo.gif
  - WAP display: “LITTLE DIX”
  - print-disabled: http://www.littledixbay.com/audio/ldb.au
- Sort name for topic “Brit. Virgin Islands”:
  - “British virgin islands”

The purpose of a variant name is defined by its parameters. As a scope set defines the validity of topic characteristics, the parameters define the ‘validity’ of the variant name. A variant name can have sub-variants supporting cascading purposes (e.g., different display names for different output devices). The parameters of the ‘super’-variants are inherited down to the ‘sub’-variants.

A display name could be either a text string or a referenced resource. The referenced resource could contain any kind of data to be used to display the name of a topic (e.g., icon, audio file). A sort name is a text string defining the sort key of the topic. The sort key is used to calculate the – sort – position of the topic when listed together with other topics. Both display name and sort name are defined as standardised Published Subjects (see section 3.8 *Published Subjects* for further details), which are application independent.

*NOTES:*

1. *The base name is used for displaying and sorting purposes if no display name or sort name is provided.*
2. *Variant names are not governed by the Topic Naming Constraint (see section Figure 15* The topic naming constraint (TNC)*). Which means that there could be two or more topics with the same variant name.*
3. *There is no variant name without a base name. Base name could be empty, but it has to be there.*

Variant name syntax:

```
<topic id="little-dix-bay">
  <baseName>
    <baseNameString>Little Dix Bay
    </baseNameString>
    <variant>
      <parameters>
        <subjectIndicatorRef
          xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.
xtm#sort"/>
      </parameters>
      <variantName>
        <resourceData>
          british virgin islands
        </resourceData>
      </variantName>
    </variant>
  </baseName>
  ...
</topic>
<topic id="brit-virgin-islands">
  <baseName>
    <baseNameString>
      Brit. Virgin Islands
    </baseNameString>
    <variant>
      <parameters>
        <subjectIndicatorRef
          xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.
xtm#display"/>
      </parameters>
      <variant>
        <parameters>
          <topicRef xlink:href=
            "#s1024x768"/>
        </parameters>
        <!-- display + 1024x768 -->
        <variantName>
          <resourceRef xlink:href=
"http://www.littledixbay.com/images/ldb
logo.gif"/>
        </variantName>
      </variant>
      <variant>
        <parameters>
          <topicRef xlink:href="#wap"/>
        </parameters>
        <!-- display + wap -->
```

```
        <variantName>
          <resourceData>LITTLE DIX
          </resourceData>
        </variantName>
      </variant>
    </variant>
  </baseName>
  ...
</topic>
```

## 3.6 Addressable and non-addressable subjects

As mentioned earlier, topics represent subjects in the computer. So far, we have presented topic names as describing the subject represented by a topic. However, names are quite often ambiguous and different people might think about different subjects when seeing the same name. There is also the issue of automatic merging of multiple topic maps into one topic map – and automatic merging is finally about the question if two topics are representing the same subject and have as such to be merged into one topic (see section 3.10 *Merging topic maps*). These are good reasons to take a closer look at the nature of subjects.

### 3.6.1 Addressable subjects

If the subject itself is a resource in the computer – it has an URI –, it is an addressable subject. It can be electronically addressed.

Examples of addressable subjects:

- Web site “http://bviwelcome.com”
- XTM specification “http://www.topicmaps.org/xtm/”

The address uniquely identifies the subject. A topic can refer to its – addressable – subject using the address as resource reference.

*NOTE : An addressable subject is the addressed resource, it is not the subject described by the resource. E.g., the subject addressed by “http://www.topicmaps.org/xtm/” is the XTM specification itself, it is not XTM.*

## 3.7 Non-addressable subjects

If the subject exists outside the bounds of the computer system, it is a non-addressable subject. There is no way to address the subject itself electronically. The majority of subjects are non-addressable.

Examples of non-addressable subjects:

- British Virgin Islands
- The topic map paradigm

The identity of a non-addressable subject is not computable. Nevertheless, its identity can be established indirectly using a so-called subject identifier.

### 3.7.1 Subject identifier & subject indicator

A subject identifier is a machine readable URI representing a non-addressable subject. The subject identifier is a kind of proxy for the subject.

Examples of subject identifiers:

- URI “http://www.topicmaps.org/xtm/1.0/country.xtm#VG” representing the country “Virgin Islands (British)” as defined by ISO 3166 [9].
- URI “http://www.topicmaps.org/xtm/1.0/language.xtm#en” representing the language “English” as defined by ISO 639 [8].

If the subject identifier, the URI points to a resource – which is not required – the resource is called a subject indicator. It should contain a human readable documentation describing the non-addressable subject. The resource could be a document, audio, video, topic map, anything able to describe to a human what the subject is about. The resource indicates (specifies) the subject, which led to its name ‘subject indicator’.

An important marginal note: Topic maps claim to be a format for knowledge representation, a base technology for knowledge management. Whenever we talk about knowledge we talk about humans and their individual way of interpretation and understanding of facts, information, and knowledge. A subject indicator describes a certain subject with the purpose of being widely used by various topic map authors. Consequently, it is very important that the author writing the subject indicator clearly states what the subject is about and what not. But even if the author works as accurate as possible, different people will never interpret a subject indicator exactly 100% the same way – philosophically speaking, we are all biased by a our experiences and contexts and therefore understand things, maybe only slightly, different. However, the concept of subject indicators is worth using it, because there is finally no alternative to it.

## 3.8 Published subjects

Subject indicators provide together with their subject identifiers a technical solution to identify non-addressable subjects. If topic map authors use these identifiers depends on certain conditions:

- Marketing and visibility: The author has to be aware of the existence of subject identifier.
- Agreement: The author has to agree that the subject indicator describes really the same subject the author has in mind.
- Trust and stability: The author wants the guarantee that the subject indicator is accessible and unchanged at least as long as the created topic map will exist.

Marketing, visibility, stability, and trust of and in content (= subject indicators and identifiers) are typical tasks of a publisher. That is the reason why a subject and its subject indicator, which are placed and maintained at an advertised site, are called published subject (PS) or published subject indicator (PSI) respectively. Agreement is, of course, up to the humans defining and using published subjects.

Published subjects are necessary to standardise and re-use non-addressable subjects in various topic maps and to facilitate merging.

*NOTE : Several OASIS Technical Committees define sets of published subjects for vertical application domains. The OASIS Topic Maps Published Subjects TC develops a recommendation about how to define, document, publish and apply published subjects.*

### 3.8.1 Subject identity

The subject identity finally connects a topic with the subject the topic is representing. It depends on the kind of subject how the subject identity is established.

The subject identity is a reference to either

- a resource, which is the addressable subject,
- a subject identifier, which identifies a non-addressable subject but it is not the subject, or
- an other topic, which shares the same subject with the current topic.

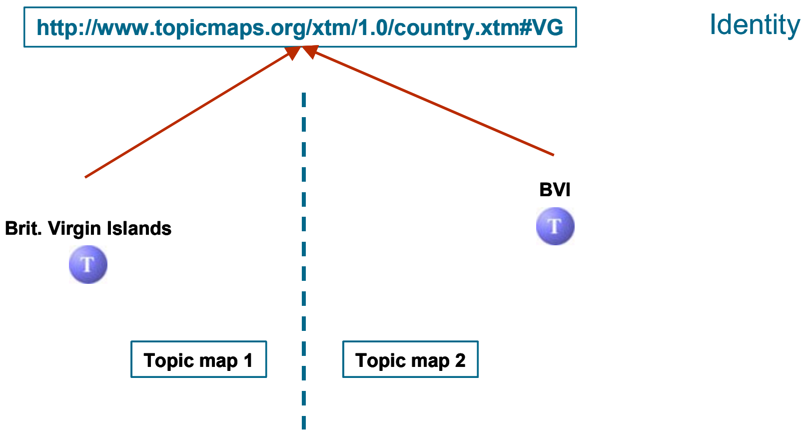



**Figure 14** Identity of a topic example

Subject identity syntax:

```
<topic id="bviwelcome.com">
  <subjectIdentity>
    <resourceRef xlink:href=
      "http://bviwelcome.com"/>
  </subjectIdentity>
</topic>
<topic id="brit-virgin-islands">
  <subjectIdentity>
    <subjectIndicatorRef
      xlink:href=
"http://www.topicmaps.org/xtm/1.0/count
ry.xtm#VG"/>
  </subjectIdentity>
</topic>
<topic id="bvi">
  <subjectIdentity>
    <topicRef xlink:href=
      "#brit-virgin-islands"/>
  </subjectIdentity>
</topic>
```

### 3.8.2 A summary on subjects

The importance of subjects was already emphasized in our back-of-book example. Subjects are the 'things' every topic map is finally about. They are the bridge between the 'real' world and 'modelled' world, the topic map. Every topic is a 'computerised' subject. A subject declares the topic's 'identity'. Furthermore, subjects control the merging of topic maps – as we will explain further down. The topic map standard underpins their importance by providing a complete set of concepts and rules dealing with subjects.

Subject can be either addressable (the subject itself has a URI) or non-addressable. A topic representing an addressable subject can use the subject's address, the URI to establish the topic's identity. A topic representing a non-addressable subject has to use a proxy URI, the subject identifier to establish the topic's identity. The proxy URI, the subject identifier might resolve to a human readable information (e.g., an HTML document) describing the subject. This human readable information, the subject identifier is pointing to, is called the subject indicator. Figure 15 illustrates the subject identity of topics and the relationship between topic, resource reference, subject identifier, and subject indicator.

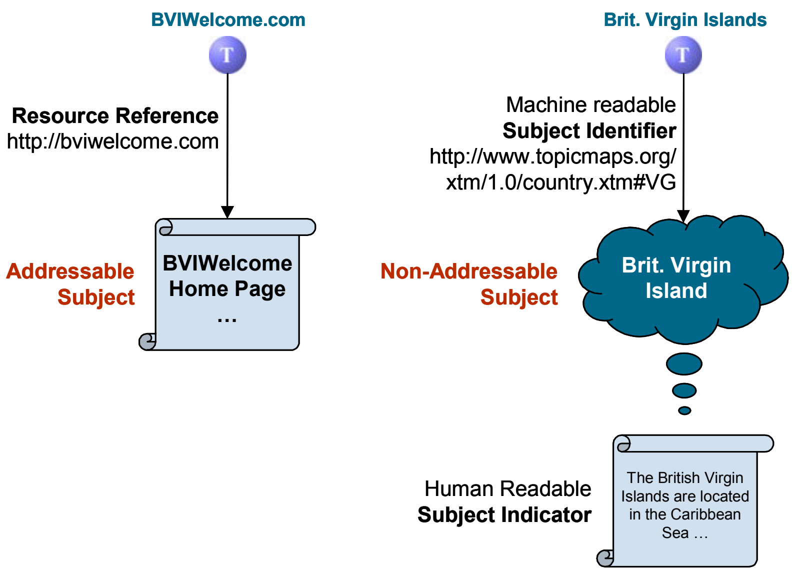



**Figure 15** Subject identity of topics

## 3.9 The topic naming constraint

The topic naming constraint (TNC) states that

*Topics having the same base name in the same scope implicitly refer to the same subject and therefore should be merged.*

*XTM 1.0 specification*

*NOTE : 'Having the same base name' means that the content of the <baseNameString> elements are compared byte-by-byte. Even empty <baseNameString> elements have to be considered equal.*

The TNC is controversial. On the one hand it reflects that humans – in general – identify subjects by giving them a name. On the other hand it bears the danger of merging topics even if they are not about the same subject – they just have accidentally the same name. Even worse, the TNC has precedence over the subject identity provided by addressable subjects. If two topics have the same base name in the same scope, they have to be merged – regardless if the topics are identified by different addressable subjects stating that they are representing different subjects. Moreover, after merging, the software has to report an error because topics must not be identified by two addressable subjects – only one is allowed. Even if topics are instances of different classes the TNC has to be applied.

There are several work-arounds to overcome the TNC:

- Scope every base name with its topic – only a theoretical solution, not of practical value because too many topics are misused as scoping topics.
- Scope every base name with the topic's class – more practical, but still leads to a lot of unnecessary scoping topics and can still not avoid unwanted merging.
- Use variant names instead of base names – especially display name provides appropriate semantic. But be aware, that there is no variant name without a base name.

empolis TRANSFORMING INFORMATION INTO VALUE

- Use resource data occurrences instead of base names – is not only an alternative to store names but also provides the ability to provide different name classes.

*NOTE : The ISO SC34 committee responsible for the topic map standard decided to make the TNC an optional feature. Base names will become instances of name classes. A name classes can represent a name spaces with unique names i.e. the TNC. It is therefore up to the topic map author to switch the TNC on or off by assigning a name space class to a base name (on), a non-name-space class (off), or no class at all (off).*

As long as the amendment to the ISO standard is not adopted topic map authors have to consider the TNC by either storing names as base name (= TNC is active) or by storing names as variant names or, even more TNC-safe, as resource data occurrences (= TNC is inactive).

When the author decides to activate the TNC two options are possible to deal with same base names:

- Qualify the base name by a scope consisting of:
  - topic class e.g., "Jost van Dyke" scoped by topic class "Pirate" or "Island";
  - associated topic e.g., "Jost van Dyke" scoped by the associated topic "The Netherlands" ("Birth" association) or "Brit. Virgin Islands" ("Part-whole" association).
- Qualify base name by adding a name extension following same principles:
  - e.g., "Jost van Dyke (pirate)", "Jost van Dyke (island)";
  - e.g., "Jost van Dyke (The Netherlands)", "Jost van Dyke (Brit. Virgin Islands)".

## 3.10 Merging topic maps

Merging is the process of joining two topic maps or joining two topics. It is built-in the topic map standard because merging of indices was the requirement initiating the development of the topic map paradigm.

The process of merging two topic maps is initiated by the <mergeMap> directive or because of application specific purposes. Merging of two (or more) topics has to be done by a topic map processor when a topic map coded in XTM is imported or when the imported topic map is changed e.g., by an interactive authoring tool or by an API function call.

Figures 16 and 17 show two topics map before merging and the merged topic map as the result of merging.

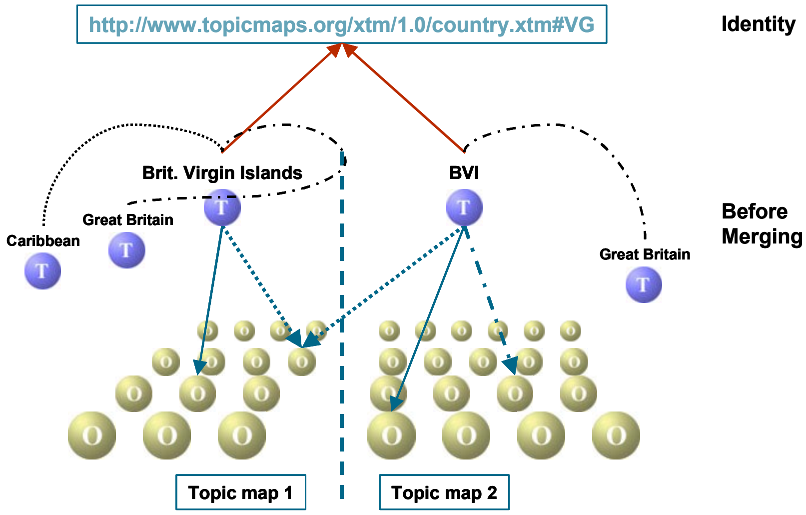



**Figure 16** Merging example: Two topic maps before merging

When two topic maps are merged all topics with the same subject identity or those complying to the topic naming constraint (TNC) are merged. When two topics are merged the characteristics of the resulting topic is the (set) union of the characteristics of the original topics, which implies that all double names, occurrences, and associations will be removed. Names are compared byte-by-byte, occurrences must be instances of the same class and point to the same resource (or have the same resource data), associations must be of the same class, have the same roles, and the same role playing topics.

*NOTE : The use of scope has a certain side effect on merging, because scope is part of the TNC rules. This implies that scope should be used carefully when the topic map might be merged with other maps.*

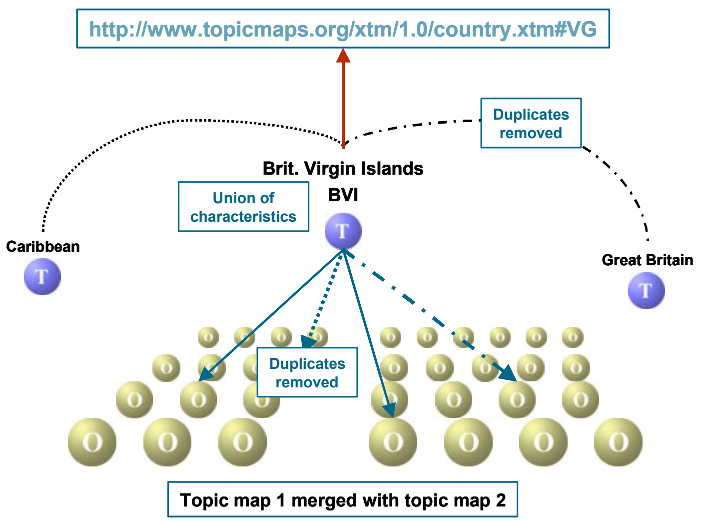



**Figure 17** Merging example: One topic map after merging

The <mergeMap> directive does not only merge a referenced topic map with the one containing the directive. The directive is also able to add resource references, subject identifier, and topics to the scope sets of all characteristics of the referenced topic map. Consequently, this helps avoiding unwanted TNC merging and also provide a mechanism to identify the 'merged-in' characteristics in the merged topic map.

The syntax example merges the topic map "http://mydomain.com/bvi.xtm" into the "Caribbean" topic map and assigns the resource reference "http://mydomain.com/bvi.xtm", the subject identifier "http://mydomain.com/bvi.xtm", and the topic "English" to all scope sets of the merged-in map.

Merging topic maps syntax:

**Identity**

**After Merging**

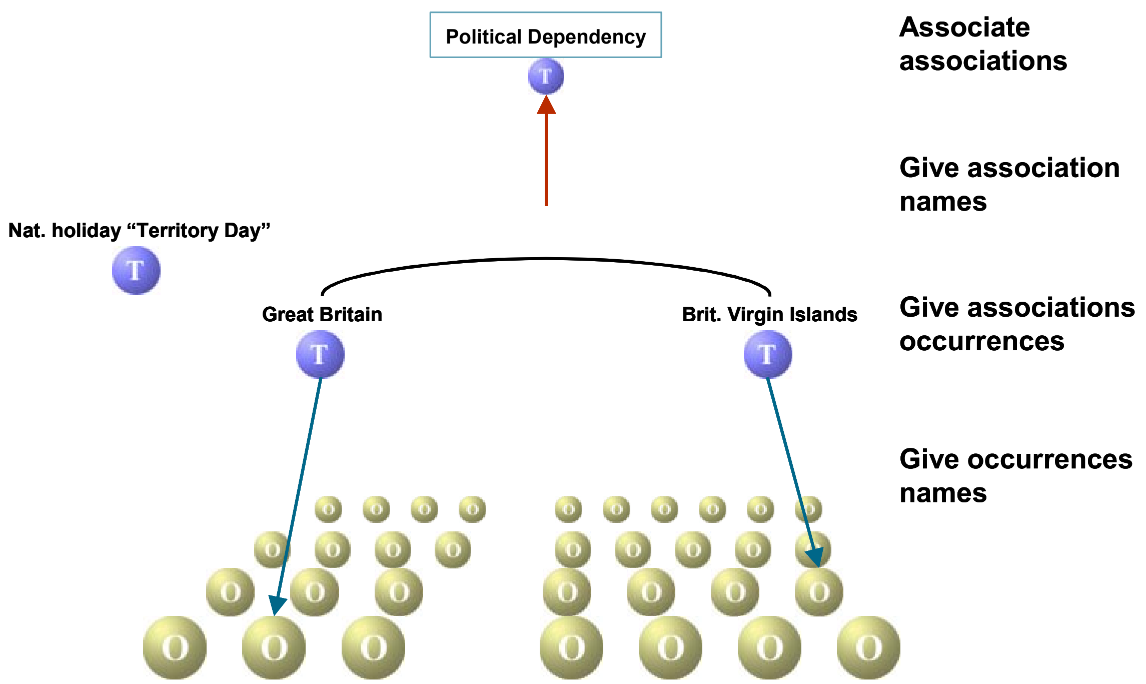
```
  <topicMap id="caribbean" ... >
    ...
    <mergeMap xlink:href=
  "http://mydomain.com/bvi.xtm">
      <resourceRef xlink:href=
  "http://mydomain.com/bvi.xtm"/>
      <subjectIndicatorRef
        xlink:href=
  "http://mydomain.com/bvi.xtm#politi
  cal"/>
      <topicRef
  xlink:href="#english"/>
    </mergeMap>
    ...
</topicMap>
```

## 3.11 *Reification: everything becomes a topic*

We learned that topics have the characteristics names, occurrences, and they are associated one another. But sometimes the topic map author wants to assign these characteristics to associations or occurrences.

Figure 18 shows some occurrence and association examples, which require additional descriptive information:

**Figure 18** Requirements leading to reification



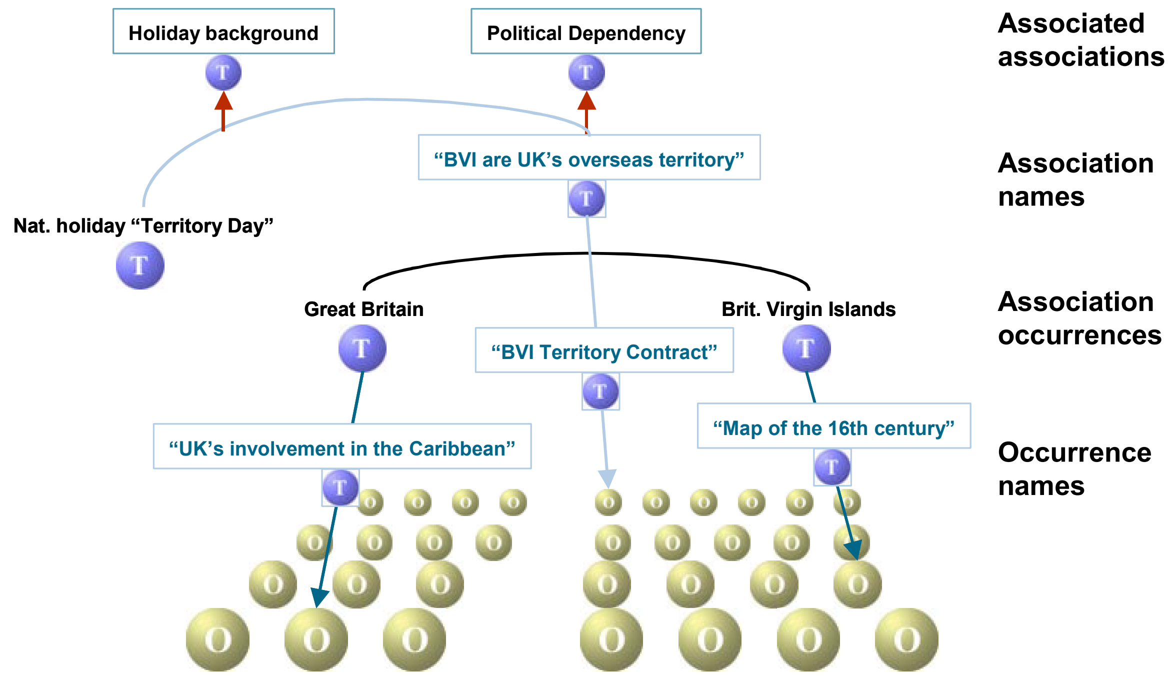



**Figure 19** Reification example

- Assigning names to the two occurrences ("UK's involvement in the Caribbean", "Map of the 16th century");
- Assigning a name to the association ("BVI are UK's overseas territory");
- Assigning an occurrence (the resource "BVI Territory Contract") to the association, which proves that the association is a true statement;
- Associating the association with the topic "Nat. Holiday 'Territory Day'" expressing that the fact that BVI are UK's overseas territory is the historical reason for this holiday.

We learned that associations and occurrences do not have characteristics. So, what to do? Remember that a topic represents, reifies a subject and that all characteristics assigned to the topic mean that the reified subject carries these characteristics. What hinders us from reifying an occurrence, an association, or an arbitrary addressable information resource as a topic? By definition, this topic would stand for the occurrence, association, or resource in the topic map.

The 'technique' of reification allows us to assign the topic characteristics (names, occurrences, associations) to all addressable subjects including objects of the topic map itself. Figure 19 shows topics reifying occurrences and associations and characteristics assigned to these topics.

Reification syntax (last topic assigns a property-value pair to a resource):

```
<association id="UK-territory-BVI"> ...
</association>

...

<topic id="UK">

  <occurrence id="UK-involvement"> ...
  </occurrence>

</topic>

...

<topic id="r-UK-territory-BVI">

  <subjectIdentity>
    <subjectIndicatorRef xlink:href=
      "#UK-territory-BVI"/>
  </subjectIdentity>

  ...

</topic>

...

<topic id="r-UK-involvement">

  <subjectIdentity>
    <subjectIndicatorRef xlink:href=
      "#UK-involvement"/>
  </subjectIdentity>

  ...

</topic>

...

<topic id="bviwelcome.com">
```

```
<subjectIdentity>
  <resourceRef xlink:href=
    "http://bviwelcome.com"/>
</subjectIdentity>
<occurrence>
  <instanceOf>
      <topicRef xlink:href=
        "#visitors-per-day"/>
    </instanceOf>
    <resourceData>842</resourceData>
  </occurrence>
</topic>
```

# 4 How to Create a Topic Map

The topic map paradigm standardised in ISO/IEC 13250:2000 [10] defines the concepts and two syntaxes – HyTM and XTM – to interchange topic maps between different topic map software systems. The standard does not define how a topic map is created and maintained. This task is completely left up to the user and the application.

There are three obvious possibilities for the creation of topic maps: manually, automatically, or a mixture of both. However, there are also two different layers in a topic map: the instances and the classes. Classes are created in the design phase, instances are created in the authoring phase, and both phases could make use of different creation techniques.

This chapter introduces a systematic approach to manual design and discusses options for automatic generation of topic maps.

## *4.1 A topic map cookbook*

The topic map cookbook is a guide to successful manual topic map design. The sections of the cookbook continues the “British Virgin Islands” example.

### *4.1.1 Objectives and definitions for the cookbook*

#### 4.1.1.1 Objectives

Topic map design and implementation is achieved through the following key tasks:

1. define the application domain – what will be covered by the topic map;
2. define the functional requirements – who will use the topic map and for which purposes;
3. define the schema – what kinds of subjects will be covered and how will they be related; what should a valid and consistent topic map look like;
4. select tools and implement application – which software and architecture to use to implement the application;
5. populate the topic map – generate instances automatically or manually; does the populated topic map satisfy the user requirements; does the schema require improvement;
6. maintain the topic map and its application – what has to be done to keep the application running and topic map up-to-date.

The cookbook focuses on the schema definition.

#### 4.1.1.2 Definitions

The cookbook will use some terms, which are not yet defined.

Ontology:

*An ontology is an explicit specification of a conceptualization.*

*T. Gruber, What is an Ontology? [5]*

*Ontology defines the kinds of things that exist in the application domain.*

*J.F. Sowa, Knowledge Representation [16]*

Class:

*A group, set, or kind sharing common attributes.*

*Merriam-Webster's Collegiate Dictionary*

Constraint:

*Rule(s) governing classes of objects in an ontology for the purpose of validation and/or guided topic map authoring.*

*ISO JTC1 SC34, N0226 [12]*

Template (topic, association, occurrence, scope):

*Declaration of a class and its constraints to which instances of the class must conform.*

*ISO JTC1 SC34, N0226 [12]*

Topic map schema:

*The collection of templates that together define a class of topic maps.*

*ISO JTC1 SC34, N0226 [12]*

Some comments on constraints: Constraints are a powerful mechanism to control the topic map authoring and to semantically validate the content of a topic map. As of today (December 2002), no topic map constraint language (TMCL) is standardised and only a few proprietary solutions exist. Nevertheless, the TMCL standard is under development and first requirements outline the possibilities of a constraint language.

The cookbook describes which constraints could be defined for the different topic map constructs. The definition itself has to be either done in the vendor-specific notations or in prose as an editorial guideline for the topic map authors.

Some examples of typical constraints:

- Constraints for topic class "island":
  - names – minimum one name in scope "English"
  - occurrences – "map" (exactly one), "article" (max. one)
  - role playing – participates in "political containment" association playing role "political containee"
- Constraints for association class "political containment":
  - roles – exactly one "political container", exactly one "political containee"
- Topic classes of role players for roles "political container" and "political containee":
  - "country" – "island"

Constraints can be defined from different 'angles' (e.g., topic class constraints the association role its instances have to play and association constraints the possible role player classes). As a consequence and caused by the absence of a standardised constraint language, the designer has to be careful not to define conflicting constraints.

### *4.1.2 Analysis phase*

The analysis phase ensures that the project results in a topic map solution, which really solves the initial problem – and not something else.

Roadmap of the analysis phase:

1. List the various requirements the topic map application has to fulfil and assign priorities. Consider user requirements, functional requirements, design requirements, system requirements, performance requirements as well as budget, time, and human resources.
2. List existing schemas, which could be re-used.
3. Brainstorm the subjects of the application domain and organise the brainstorming results.

#### 4.1.2.1 Requirements analysis

Requirements should influence all further design decisions. Especially the user requirements should be taken very seriously. The listed questions (see below) can be used as a guide through the analysis phase and form a 'checklist' for completeness. They also help to structure the requirements into several categories (e.g., design requirements, population requirements, user requirements, system requirements). All collected requirements should be listed in a document. A priority assigned to each requirement helps to distinguish important from unimportant ones.

Some categorized questions to find requirements:

**General design:**

a) Where are the boundaries of the application domain?

The question is critical because width and depth of the ‘knowledge’ captured in the topic map is ‘infinite’ but has to be limited to become manageable. The defined boundaries will govern the detail of subject (= depth) and the breadth of subject (= width) and will control topic map design and population. Consider in which direction the domain will change in the future.

b) Who are the users and what are their expectations? Are there important user groups with specific requirements?

The topic map has to fulfil the user requirements and expectations. They have to be considered at every design step. If the topic map has to serve different user groups take care about the most important groups.

c) What are the constraints for budget and/or time? What are the available (human) resources?

The topic map paradigm allows a low-cost start and later improvement. Avoid starting with too great detail – it is expensive and might not be necessary. Start that simple just that it fulfils the main user requirements.

d) Should the topology of the topic map be restricted?

A tree-like topic map is already very powerful (e.g., it models a taxonomy or a subject classification) but easier to understand and maintain than topic maps modelling multi-dimensional with lots of associations. You might want to restrict the number of tree hierarchies to avoid to detailed classification? Think carefully about what you want to model (e.g., a subject categorisation, subject classification, taxonomy, thesaurus, knowledge network, web site map, or arbitrary metadata) – your decision will effect the work of all topic map authors and users.

e) Do the users want to ‘see’ the topic map bits-and-bytes or higher level business objects?

Some topic map applications display all details of topic maps to the user – all their classes, topics, names, occurrences, associations, scopes. Other applications hide the topic map details and provide an abstraction of the topic map paradigm to the user by presenting only the business objects the user is familiar with. The first category of applications is about taxonomies, thesauri, knowledge models. The latter category is about topic maps as complex meta data in document management and content management hidden by the user interfaces of the software systems.

**Use of scope:**

a) Are different views required?

The scope concepts allows modelling different views on the topic map and provides the ability to filter, hide, parts of it. Views could model user interests, languages, access rights etc. They can be considered from the beginning or added later.

b) Will you need a multi-lingual topic map?

Multi-lingual names of topics is typically modelled with scope. The OASIS Technical Committee Published Subjects for Geography and Languages [33] provides a set of published subjects for all major languages, which can be used.

**Topic map population:**

a) What will the topic map population look like?

A topic map could be manually authored or automatically generated or both. The expected quality of the outcome should govern the decision. Usually, a handcrafted topic map is more accurate than a generated one.

b) Do you need to deal with legacy data?

Legacy data means already existing taxonomies, thesauri, ontologies, which could be migrated to or converted into a topic map. Quite often, this kind of legacy data exists in industries and organisations and could be an ideal starting point for the topic map development. Critical is to find them and to convert them.

c) What do the resources look like which should become topic occurrences?

Identify the resources, which will be candidates for occurrences in the design phase. It is a good idea to limit the resources to a manageable amount or to categorize them and only deal with the categories – it is not necessary to deal with all resources in the design.

d) How tight should the topic map be connected to resources?

Manual creation and maintenance of occurrence might be an expensive task. Consider also automatic occurrence creation/maintenance by an categorization/classification tool.

**Editorial issues:**

a) Who are the authors and what are their skills?

Ensure that the authors have to knowledge about the subject domain and the target audience, the users.

b) How many authors will work on the topic map?

Will the topic map be created by a single author or by many concurrently working authors? This questions has certain implications on the system environment (transaction handling, locking, etc.) and the organisation of the authoring work. You should also consider that the topic map schema shall not be changed at the same time when authors populate the topic map – it might lead to unexpected results. Change the schema when no author is populating the map.

c) Do you have to care about access rights and user management?

Different groups of authors might have different access rights. The allowed user actions (e.g., create, change, delete a topic, name, occurrence, association) might depend on the topic's class and the user group.

d) Do you have to distinguish between the topic map continuously authored and the topic map the users operate on?

A publishing step might separate these two topic maps. The authored topic map is 'copied' periodically and the copy becomes the user topic map. The 'copy' might be a full copy or only a copy of a part of the topic map. The part could be identified by a scope set or by a topic map query. Define the parameters defining how the selection looks?

e) How will the maintenance of the topic map be handled?

Ensure that outdated or invalid information will be deleted at regular intervals. Plan when and how you will run the automatic updates – this is an issue when you will auto populate your map.

f) Is or will merging be an issue?

Investigate if sets of published subjects are available or if they do have to be defined. Train your authors that they apply the published subjects to topics whenever possible. Think about how you want to deal with the topic naming constraint.

g) Do you have to take care of the ratio between topics, occurrences, and associations?

The readability of the topic map depends on these numbers. E.g., a topic associated with hundreds or thousands of other topics is hard to navigate. However, if the topic map is not navigated or displayed but used for other purposes large numbers might not harm that much.

#### 4.1.2.2 Do not reinvent the wheel

Humans have been ordering, classifying and grouping information for generations. Topic maps are quite often built over existing resources and repositories or are used to migrate from a legacy system to a topic map application. The structures of the resources, repositories, or legacy systems should be investigated to see if they can be re-used in the topic map schema. You might also check if the same or similar schemas have been developed before (in your community, in public).

Look at existing schemas for sources for topics, classes, or themes:

- ontology;
- subject categorisation, subject classification, taxonomy;
- relational schema, OO schema;
- metadata vocabulary;
- table of contents, index, glossary, thesaurus, data dictionary;
- document structures (DTD, XML Schema);
- link structures.

#### 4.1.2.3 Brainstorming

Because topics computerise the – relevant – subjects of the application domain, brainstorming of these subjects is the starting point. Just list anything regardless of what it might finally become a class, a topic, an association, an occurrence, or a theme. Think about the different angles, views from which users will look at the topic map. Be creative, but have the topic map user in mind (they are the ones you are developing the topic map for).

As brainstorming is – with intention – a creative but quite disorganised process, the identified list of subjects has to be organised before it becomes input to the design phase.

1. Ensure that all your listed subjects are really different subjects and not just synonyms. If you find synonyms just record them as alternative names for a subject.

2. Look at the listed subjects and identify the classes for topics, occurrences, and associations, as well as the association roles. You might also find instances of topics, occurrences, or associations in your list. Note them as examples for the appropriate classes.
3. List subjects, which might be used in scope sets defining views, context information, or access rights.

**Example "British Virgin Island":**

*Brainstormed subjects:* British Virgin Islands, country, map, CIA world fact book entry, Tortola, Virgin Gorda, island, island belongs to country, Cane Garden Bay, White Bay, bay, bay belongs to island, photograph, Little Dix Bay Resort, hotel, hotel is located at island / bay, web site, Brandywine Bay Restaurant, restaurant, restaurant is located in town / hotel / harbour / marina / bay, menu card, Road Harbour, harbour, harbour / marina belongs to town, harbour / marina is at bay, navigation map, Virgin Gorda Yacht Harbour, marina, fees, Road Town, town, city map, town is on island, Speedy's, ferry line, time table, ferry line connects harbours, Pusser's Company Store, souvenir shop, shop is located in town / harbour / marina, Sunsail, yacht charter company, yacht charter company is located in harbour / marina, Hertz, car rental, car rental is located in town, taxi company, contact information, taxi company operates on island, regular tourist, sailor, English, French, German, language.

Organised as classes:

- topic classes: country (e.g., British Virgin Islands), island (e.g., Tortola, Virgin Gorda), bay (e.g., Cane Garden Bay, White Bay), town (e.g., Road Town), harbour (e.g., Road Harbour), marina (e.g., Virgin Gorda Yacht Harbour), hotel (e.g., Little Dix Bay Resort), restaurant (e.g., Brandywine Bay Restaurant), souvenir shop (e.g., Pusser's Company Store), yacht charter company (e.g., Sunsail), ferry line (e.g., Speedy's), car rental (e.g., Hertz), taxi company, language.
- occurrence classes: map, city map, navigation map, CIA world fact book entry, photograph, web site, menu card, fees, time table, contact information.
- association classes: island belongs-to country, bay belongs-to island, hotel is-located-at island / bay, restaurant is-located-in town / hotel / harbour / marina / bay, harbour / marina belongs-to town, harbour / marina is-located-at bay, town is-on island, ferry line connects harbours, shop is-located-in town / harbour / marina, yacht charter company is-located-in harbour / marina, car rental is-located-in town / hotel, taxi company operates-on island.

Scoping topics: regular tourist, sailor, English, German, French.

### *4.1.3 Design phase*

The brainstorming session in the analysis phase provided a 'first cut' at a list of classes and association roles. The design phase now organises and defines the classes more thoroughly.

Every class should be defined as a template (including constraints). It is this collection of templates that forms the schema for our topic map. Even if TMCL is not standardised, templates and constraints help to achieve a clearer design and better documentation. To express constrains, straight text or some form of business rule notation could be used. The critical thing is that you clearly and precisely identify the constraints that must operate for this topic map.

#### 4.1.3.1 Topic classes

Distinguish between resource, topic instance, and topic class. Here is a rules-of-thumb: Only resources reified as topics can have characteristics, e.g., have a name, have an occurrence, or be associated with other topics. If a resource does not need to have names, occurrences, or be associated, the resource will probably be 'only' an occurrence, but not a topic.

The topic template constraints the properties of the instances:

- number of name(s) in scope(s);
- occurrence class(es) in scope(s) and their cardinality;
- association role(s) the instances play in association of certain class;
- distinguish between optional, must, and must not constraints.

**Example "British Virgin Island":**

Constraints of topic class "island":

- one name in each of the scopes "English", "French", and "German";
- exactly one "map" occurrence in scope "regular tourist"; zero or more "map" occurrences scoped by "sailor";
- has to be associated with a "country" by a "political containment" association and playing the role "containee".

#### 4.1.3.2 Occurrence classes

Check available resources and their semantics for the application domain to identify occurrences and their classes. Look at every topic and association class and figure out – from the user's perspective – meaningful occurrence classes.

The occurrence template constraints the use of the occurrence class instances:

- topic class(es) instances of current occurrence class are assigned to;
- scope(s);
- number of name(s) in scope(s) ***;
- distinguish between optional, must, and must not constraints.

*NOTE : *** = through reification*

**Example "British Virgin Island":**

Constraints of occurrence class "photograph":

- may be used at instances of classes "island", "bay", "town", "harbour", "marina", "hotel", "restaurant";
- may be scoped by "regular tourist", "Sailor";
- zero or one name in each of the scopes "English", "French", and "German".

#### 4.1.3.3 Association classes

Look at all topic classes and their possible combinations to figure out the meaningful association classes.

Detailed topic classes can sometimes be expressed as general topic class plus an association.

- "Pusser's Company Store" is instance of class "souvenir shop";
- "Pusser's Company Store" is instance of class "shop" & "Pusser's Company Store" is-selling "souvenir".

Keep this in mind when inventing topic classes and association classes.

Naming of association class:

- should be a noun expressing the relationship;
- e.g., "company location".

Naming of association role:

- should be a noun, which might be a subclass or superclass of that topic class the role players are instances of;
- e.g., "company" (superclass of "hotel", "restaurant", "souvenir shop", "yacht charter company", "car rental") / "location" (superclass of "island", "bay", "town", "hotel", "harbour", "marina").

Naming of arc (possible for binary associations only):

- should be a verb expressing the meaning when getting from one role playing topic to the other one;
- e.g., (company) "is located in" (location) / (location) "is location of" (company).

The association template constraints the use of the association class instances:

- scope(s);
- association role(s) and their cardinality;
- combination of topic class(es) allowed to play the role(s);
- arc labels;
- number of name(s) in scope(s) ***;
- occurrence class(es) in scope(s) and their cardinality ***;
- association role(s) the instances of current association class play in association of a certain class (= associated association) ***;

- distinguish between optional, must, and must not constraints.

*NOTE : *** = through reification*

**Example "British Virgin Island":**

Constraints of association class "company location":

- may be scoped by "regular tourist", "sailor";
- must have role one "company" and one role "location";
- allowed topic classes of the role players of roles "company" and "location" are:
  - "hotel" – "island", "bay";
  - "restaurant" – "town", "hotel", "harbour", "marina", "bay";
  - "shop" – "town", "harbour", "marina";
  - "yacht charter company" – "harbour", "marina";
  - "car rental" – "town", "hotel", "harbour", "marina";
- arc labels are: (company) "is located in/at" (location) / (location) "is location of" (company);
- may have zero or more "map" occurrences, which may be scoped by "regular tourist", "sailor", "English", "French", "German".

#### 4.1.3.4 Scope sets

Scope sets express the validity, applicability of topic characteristics names, occurrences, and associations. Scope is used to define various views on the topic map. Additionally, scope sets have an impact on name-based merging of topic maps by providing name spaces. Analyse the needed views and merging name spaces to identify the scoping topics.

Scoping topics should be instances of classes (for grouping) and not just 'classless' topics.

Class examples:

- properties: e.g., languages
- access control: e.g., security level, subscription level
- views: e.g., effectivity, skill level

Logical interpretation of scope combinations is up to the software application. Remember: Scope sets are flat lists.

The scoping topic template constraints the use of the scoping topic:

- topic class(es) to which instance characteristics the scoping topic might be assigned to;
- occurrence class(es) to which instances the scoping topic might be assigned to;
- association class(es) to which instances the scoping topic might be assigned to;
- distinguish between optional, must, and must not constraints.

**Example "British Virgin Island":**

Constraints of scoping topics "English", "French", "German":

- must be assigned to names of instances of topic class "country", "island", "bay";
- may be assigned to occurrences of class "map", "city map", "navigation map", "web site", "menu card", "fees", "time table", "contact information";
- must not be assigned to any association.

#### 4.1.3.5 Class hierarchies

Class hierarchies are an essential part of an ontology. They model a taxonomy and are the foundation for compact topic maps, inferencing, powerful searching.

Distinguish between "instance-of" (is-a) and "is-subclass-of" (is-a-kind-of) – see also section Figure 10 *Class-instance vs. superclass-subclass.* You might need your own association class to model the class hierarchy if your 'classes' are not 'real' classes as defined by the topic map paradigm and the built-in superclass-subclass association seemed not applicable.

**Example "British Virgin Island":**

Various class hierarchies:

- "location"

- "country", "island", "bay", "town", "hotel" (its ground)
- "harbour"
  - "marina"
- "company"
  - "hotel", "restaurant"
  - "shop"
    - "souvenir shop"
  - "transportation company"
    - "ferry line", "taxi company"
  - "rental company"
    - "yacht charter company", "car rental"
- "map"
  - "city map", "navigation map"
- "part-whole"
  - "geographical containment", "political containment"

#### 4.1.3.6 Subject identity

If you want to prepare your topic map for proper merging with other maps you should establish the subject identity of the topics. You could either refer to addressable subjects of your application domain or you have to use subject identifier for non-addressable subjects.

It might be the case that published subjects are available for your application domain and their subject identifier could be used directly. When you have to define your own subject identifier and – as recommended – subject indicators, you should consider existing vocabularies (e.g., from ISO) as a basis.

**Example "British Virgin Island":**

Available addressable subject:

- web site bviwelcome.com: http://www.bviwelcome.com/

Available published subjects identifiers for non-addressable subjects:

- country British Virgin Islands: http://www.topicmaps.org/xtm/1.0/country.xtm#VG
- language English: http://www.topicmaps.org/xtm/1.0/language.xtm#en

Application specific subject identifiers for non-addressable subjects (note: these PSIs are invented for the purpose of the example, they are not existing in reality):

- hotel Little Dix Bay: http://psi.rosewoodhotels.com/hotel.xhtml#little-dix-bay
- souvenir shop Pusser's Company Store: http://psi.bvi.gov/store.xhtml#pussers

#### 4.1.3.7 Properties

Topics might carry properties-value pairs, which are modelled as resource data occurrences. The occurrence class is the property and the resource data string is the value.

**Example "British Virgin Island":**

Inhabitants property assigned to topics of class "country":

- "inhabitants" (property) = "20,812" (value)

### *4.1.4 Creating the designed map*

So far, the design was done without creating a real topic map – it was just 'paper work'. This section explains how to transform the design into a real topic map.

Three techniques are possible:

- Writing XTM (or HyTM) code in an ASCII editor or XML/SGML editor.
- Writing a topic map pseudo code, which is transformed to XTM automatically.
- Using topic map software, which provides an authoring user interface.

Writing native XTM is difficult and error-prone when lots of topics have to be created and interlinked with manually assigned ids. Writing pseudo code instead of XTM and transforming it automatically into XTM does not overcome the id problem, but the syntax of the pseudo code might be optimised for manual topic map authoring at ASCII level – e.g., minimising the amount of characters to be typed. An authoring interface of topic map software hides the complexity of the syntax, the id handling, and the proper interlinking of the topics from the user. As of today (December 2002), some topic map authoring interfaces exist as free or commercial products. One of the commercial products is the empolis knowledge manager (formerly known as empolis k42). It comes with a web-based and a Java-based authoring interface. Contact info@empolis.com for a free evaluation license.

#### *4.1.5 Testing*

Testing is a key part of successful topic map design and should be taken very seriously.

The designer should perform first tests. Create the obvious topics, associations, occurrences, and assign the scope sets to the topic characteristics. Navigate the topic map and go through this checklist:

- Does everything 'make sense'?
- Is something missing? Is something in the map, which should not be in there?
- Is the granularity okay?
- Are the constraints okay? Are they too restrictive or too weak?
- Are names consistent? Do you find only singular or only plural names?
- Check for classes with only one subclass or more than twelve subclasses. Both might lead to modelling problems.

Now, model a 'deep' and a 'broad' example and ask the same questions again.

After the designer has finished the checks and improved the topic map according to the results, the user should be involved in testing. Are users confused or happy? Is every expected traversal path realised? Do they find what they are looking for?

Besides user testing – which is essential as it has direct impact on the acceptance of the map – various other test methods are possible:

- Consistency test: Are all constraints fulfilled?
- Statistical test: What are the number of topics, associations, and occurrences in each or specific classes? How many and which roles do topics play in associations? How long are association chains on average, maximum, within certain classes, or class combinations, or any classes?
- Functional testing: Are the functional requirements fulfilled?

These test methods should be applied to a topic map from time to time ensuring that it keeps its quality even when it size grows. But keep in mind: A topic map is a 'living thing', which evolves together with the information / knowledge contained in the application domain.

#### *4.1.6 Documentation*

Documentation of a topic map has – as software documentation usually does – different kinds of reader. The topic map designer maintaining the map needs different documentation as the author or the user. A naïve documentation approach would consist of manuals for the different audiences, each is written and maintained separately from the topic map. However, a sophisticated documentation approach would make use of the topic map paradigm itself and integrate the documentation – or at least parts of it – into the topic map the documentation is about. The documentation would be part of the topic map and the map becomes self-documenting.

How is this done? The approach is to assign the necessary pieces of documentation as occurrences to the classes and instances, needing to be documented. The occurrence classes can model the kinds of documentation (e.g., "description", "editorial guideline", "help text"). Scopes assigned to the occurrences can express the target audience (e.g., "designer", "author", "user") and their language (e.g., "English", "French" "German", "Japanese"). Finally, it is the task of the topic map software to display the documentation to the right user in the right language.

## *4.2 Manual population of topic maps*

Manual population of a topic map is a task similar to creating a back-of-book index. The designed topic map schema is populated manually with instances of topics, occurrences, associations, topic characteristics are scoped, and subject identities are established. It is recommended that the topic map is populated using the authoring user interface of a topic map software.

In contrast to a schema authoring interface the population interface can consider information from the schema. This results in an authoring interface providing better user comfort to the author. Instead of many troublesome steps (e.g., creating a topic, assigning names, creating occurrences, finding other topics and assigning them as class or scope, finding an association class and its roles, finding the role players and linking all together as association) the population interface offers a single command (e.g., “Create instance of topic class XYZ”) hiding all the details going on in the background. After activating the command automatically those input and selections fields for the topic characteristics appear, which are defined in the schema. More advanced population interface could consider user access rights as well, allowing certain users (user groups) the editing of certain parts of the topic map only (e.g., group “geography author” is allowed to edit topics of classes “country”, “island”, “bay”, and “town”, but no other classes).

A completely different population interface might be necessary, when the topic map is modelling a specific business logic. As the user interface of a relational database application does not show any tables, rows, columns, keys of the Entity-Relationship schema, the interface on top of the topic map (database) does not show any topics, occurrences, associations but the business objects and their relations. These kinds of interfaces are very specific to the business logic being mapped and cannot be provided out-of-the-box by topic map software. Nevertheless, the software can provide a powerful API or framework facilities to easily build such interfaces.

## 4.3 Auto-population of topic maps

Manual topic map population may require lots of resources – time, money, humans. Therefore the serious question of auto-population of topic maps arises. This section addresses the question and provides some answers. In the context of this handbook answers will have to be quite general, because auto-population highly depends on the available sources and the complexity of the target topic map.

*NOTE : Whenever the term topic name in used this section it means that the name of the topic could be either expressed as base name with all the implications of the topic naming constraint or as an occurrence of class “name”.*

### 4.3.1 Objectives

We learned already, that topic map design should consider existing schemas provided by other systems. Consequently, this section covers the automatic population of topic maps. It implies that we are talking about auto-generation of topic instances, occurrence instances, association instances, scope and their assignments, as well as subject identity of topics.

### 4.3.2 Available sources

The same sources, from which the topic map design is derived, could also act as a source of input to auto-population – unstructured documents could also be used. As we are talking about auto-population, naturally all sources would have to be electronically available and accessible.

The list of available sources:

- ontology (e.g., RDF [26], RDF-Schema [25], DAML+OIL [1], OWL [24]);
- subject categorisation, subject classification, taxonomy, table of contents;
- relational database, object-oriented database;
- metadata about resources – internal in resources (e.g., MS Office properties, e-mail headers, XML mark-up) and external to resources (e.g., repository directories, filenames, file properties, Dublin Core [3]);
- index, glossary, thesaurus, data dictionary;
- document structures (XML DTD, XML Schema [30]);
- link structures (e.g., HTML links, XLink [28]);
- unstructured documents (e.g., MS-Word, PDF, e-mail, Lotus Notes).

### 4.3.3 *Possible methodologies*

#### 4.3.3.1 Ontologies

A transformation of an ontology into a topic map is straightforward, because ontologies also use the concepts of classes (=> topic classes), instances (=> topics), slots (=> associations, occurrences), class hierarchies (=> superclass-subclass), resources (=> occurrences, subject identity). The transformation would be developed for a particular ontology. The transformation could be also developed on the level of the modelling language of the ontology (e.g., RDF, DAML+OIL, OWL). In the former case the development will be quite specific for the concrete ontology and probably not re-useable, but still quite easy to realise. In the latter case development of the transformation is more difficult, but it will transform all ontologies modelled in the supported modelling language.

*NOTE : There are several proposals published as to how to map RDF and DAML+OIL ontologies to XTM topic maps on the model level.*

#### 4.3.3.2 Classification schemas

Classification schemas consist of classes, class codes, class name, class description, superclass-subclass relationships, and sometimes cross-relationships between classes in different sub-hierarchies. Resources classified according to a classification schema contain the class code as metadata (see also section 4.3.3.4 *Metadata*) or might be directly linked into the schema.

Table 1 shows a transformation approach trying to use as much of the built-in topic map concepts as possible.

**Table 1. — Transformation of a classification into a topic map**

| Classification schema | Topic map |
|---|---|
| **class** | topic of class “classification class” |
| **class code** | subject identifier of topic |
| **class name** | topic name |
| **class description** | occurrence of class “description” |
| **superclass-subclass relationship** | superclass-subclass association |
| **cross-relationship** | association of class “see also” (or more meaningful class name if possible) |
| **classified resource** | occurrence of class “classified resource” |

If it is important for the resource to be part of the topic map, it has to be represented as topic of class “classified resource”. The name could be extracted from some metadata of the resource (see below) and be assigned as a topic name. The topic’s subject identity might point to the resource if it is electronically addressable or use a resource code as subject identifier if it is not. The resource becomes an occurrence of class “resource” – or a more meaningful class name if possible. The resource’s topic and its class topic are connected by an association of class “Classification” – the use of the built-in class-instance relationship is not appropriate because the resource is quite often not a real instance of the class.

#### 4.3.3.3 Database schemas

A relational database consists of tables, columns, rows, keys, and foreign keys. The transformation will probably not map all tables into the topic map, but selected ones. The obvious mapping is shown in table 2.

**Table 2 — Transformation of relational database into a topic map**

| Relational database | Topic map |
|---|---|
| **table** | topic class |
| **row** | topic instance of corresponding class |
| **column** | name or occurrence |
| **key** | topic id |
| **foreign key** | association |

*NOTE : The association class and roles can be derived from the combination of table and column names.*

An object-oriented database consists of classes, class methods, class properties (values or pointers), and objects (= instances of the classes). The obvious mapping is shown in table 3.

**Table 3 — Transformation of object-oriented database into a topic map**

| Object-oriented database | Topic map |
|---|---|
| **class** | topic class |
| **object** | topic instance of corresponding class |
| **object address (id)** | topic id |
| **value property** | name or occurrence |
| **pointer property** | association |

*NOTE : As pointers are not necessary bi-directional in the OO paradigm, check if the generated association, which is bi-directional, makes really sense.*

#### 4.3.3.4 Metadata

This sections covers metadata about resources. The metadata can be part of the resource (internal metadata) or it can be stored independently from the resources (external metadata).

Typical internal metadata fields are title, author, category, keyword, version, creation date, publication date and that like. You find them in MS Office and PDF documents, e-mail headers, HTML <meta> tags. Also XML documents may carry internal metadata using specific mark-up. Accessing these metadata fields and values requires access to the resources e.g., via the Visual Basic API of MS Office, by scanning PDF and e-mails, or by parsing HTML and XML.

The resource becomes a topic. The title metadata field becomes the topic name. The resource file is assigned as an occurrence and might be also assigned as a subject identity. Other metadata fields might become either occurrences (e.g., version, publication data) or topics of corresponding class (e.g., author, category, keyword), which are associated with the resource's topic. Some of the metadata fields might refer to a classification schema (e.g., category) – correlating resource transformation and classification schema transformation.

External metadata is managed by the repository. The simplest repository is the file system of the computer's hard disk. Directories map to topic classes, sub-directories to sub-classes. Files in a directory might be topic instances of the corresponding class. File names are topic names. Directory path concatenated with file name is occurrence's address and might be subject identity. File extension is occurrence class or topic class. Conventions in file naming could also give some hints about the topic class.

More sophisticated repositories, like document management system and content management systems, provide a both file system metadata and as well as properties about the resources. The kinds of properties are very similar to internal metadata – they store the same kind of information.

The metadata fields might comply with a standardised metadata vocabulary e.g., Dublin Core. This opens up the possibility that the transformation can be developed once and applied to all metadata fields complying with the standard.

#### 4.3.3.5 Index, glossary, data dictionary, thesaurus

A back-of-book index, glossary, thesaurus, or data dictionary contain mainly lists of topics. An index might provide information about the topic classes, multiple topic names, and class-less associations. A glossary contains class-less associations as well. A data dictionary is mainly a complete set of the terms of the application domain with all their synonyms, translations, and a definition. A thesaurus provides the content of a data dictionary plus a classification of the terms (type of speech, narrower and broader term) and maybe corpus information, usage examples of the terms.

#### 4.3.3.6 Hierarchical document structures

XML is the most prominent hierarchical document structure. But the concept could be applied to other hierarchical formats as well. An XML resource consists of elements, sub-elements, element content, attributes, and attribute values.

Elements and attributes define classes for topics, occurrences, or associations. Element content and attribute values are topics or occurrences of the corresponding classes. Sub-elements express either a superclass-subclass relationships, associations, or occurrences.

XML example:

```
<?xml version="1.0"?>
<TravelFile>
  <Client id="1234">
    <Name>Miller, John</Name>
    <Contact>
      <Phone>+42.4321.9876</Phone>
      <Email>jm@somecompany.com</Email>
    </Contact>
    <Travel id="5678">
      <Schedule>
        <Departure>01-06-2002
        </Departure>
        <Arrival>23-06-2002</Arrival>
      </Schedule>
      <Destination>
        <Country from="01-06-2002"
          until="22-06-2002">
          British Virgin Islands
        </Country>
        <Hotel from="01-06-2002"
          until="08-06-2002">
          Little Dix Bay
        </Hotel>
        <Charter from="08-06-2002"
          until="22-06-2002">
          Sunsail
        </Charter>
      </Destination>
      <Flights>...</Flights>
    </Travel>
  </Client>
</TravelFile>
```

Extracted classes – which should have been already identified in the design – are:

- topic classes: “person”, “travel”, “country”, “hotel”, “charter company”;
- occurrences classes: “phone number”, “email”, “departure”, “arrival”;
- association classes: “client’s travel”, “travel destination”, “travel booking”.

Extraction rules for instances are:

- for every <Client> element create topic of class “person”:
  - assign value of id attribute as topic id;
  - assign content of <Client>/<Name> as topic name;
  - assign content of <Client>/<Contact>/<Phone> as resource data occurrence of class “phone number”;
  - w assign content of <Client>/<Contact>/<Email> as resource data occurrence of class “email”;
- for every <Client>/<Travel> element create topic of class “travel”:
  - assign value of id attribute as topic id;
  - assign content of <Client>/<Travel>/<Departure> as resource data occurrence of class “departure”;
  - assign content of <Client>/<Travel>/<Arrival> as resource data occurrence of class “arrival”;
  - relate this topic with current “person” topic by association of class “client’s travel”, this topic plays role “travel” and current “person” topic plays role “client”;
- for every <Travel>/<Destination>/<Country> element create topic of class “country”:
  - generate topic id;
  - assign content of <Country> as topic name;
  - relate this topic with current “travel” topic by association of class “travel destination”, this topic plays role “destination” and current “travel” topic plays role “travel”;



empolis TRANSFORMING INFORMATION INTO VALUE

- assign value of from attribute as resource data occurrence of class “arrival” to association;

- assign value of until attribute as resource data occurrence of class “departure” to association;

- for every <Travel>/<Destination>/<Hotel> element create topic of class “hotel”:
  - generate topic id;
  - assign content of <Hotel> as topic name;
  - relate this topic with current “travel” topic by association of class “travel booking”, this topic plays role “booked arrangement” and current “travel” topic plays role “travel”;
    - assign value of from attribute as resource data occurrence of class “arrival” to association;
    - assign value of until attribute as resource data occurrence of class “departure” to association;

- for every <Travel>/<Destination>/<Charter> element create topic of class “charter company”:
  - generate topic id;
  - assign content of <Charter> as topic name;
  - relate this topic with current “travel” topic by association of class “travel booking”, this topic plays role “booked arrangement” and current “travel” topic plays role “travel”;
    - assign value of from attribute as resource data occurrence of class “arrival” to association;
    - assign value of until attribute as resource data occurrence of class “departure” to association;

#### 4.3.3.7 Link structures

Links between resources express relationships, which could be modelled as associations. The link context (e.g., names of anchor elements or the classes of the related resources) provides information about the association class and its roles – as richer the link semantic as easier the extraction of the necessary information. But not all links will become meaningful associations – either because their semantics do not fit into the topic map design or the assertions expressed by the links are about pieces of information for which no topics will be created. The creation of an occurrence instead of an association might be considered in such a case.

#### 4.3.3.8 Unstructured documents

Unstructured documents are the most challenging source category for the auto-population. Typical unstructured documents are MS-Word, PDF, e-mail, or Lotus Notes files. The only available source is raw text and maybe little structure (e.g., Word paragraph formats, paragraphs in e-mails, PDF, Lotus Notes) – how to deal with internal metadata was already described in section 4.3.3.4 *Metadata*.

Analysing text is the domain of linguistics and natural language processing (NLP). But already simple text patterns modelled as regular expressions could produce useful results. Extracting topics is easier than detecting associations. Algorithms based on statistical approaches (e.g., finding often occurring combinations of terms – so called co-occurrences) could be helpful to find associations. But you have to be careful as the results of statistical approaches might contain surprising and not necessary wanted associations. There is no way to avoid this behaviour of the algorithms. Only a final manual check can detect the wrong associations.

### *4.3.4 Proper scheduling*

The previous sections explained how to auto-generate topic maps from various sources. This section explains when and how often to execute the auto-population and under what conditions.

One problem remains and is independent of the selected auto-population strategy: information redundancy. Redundant information resides in both the sources and the topic map after the topic map information is extracted from the sources. This raises the issue of how, when, and where to maintain the information and how to synchronise sources and topic map. Three approaches are possible:

- Maintain only the sources and always auto-generate the map: It has the disadvantage that all corrections done in the topic map will be lost and have to be done again and again – as long as the sources are changing. It could work out quite nice when the auto-population works that so perfectly no manual corrections are necessary at all.

- Maintain only the topic map and never auto-generate it again: This is certainly possibly when the sources are real ‘legacy’ data and will be deleted after their transformation into a topic map or when the source is stable and does not change – unfortunately a quite rare situation. If the source changes you either have to mirror every relevant change manually in the topic map or you have to take the risk that topic map and sources will drift apart.

- Maintain both sources as well as topic map: This is the most challenging approach, as you have to merge the maintained topic map with the latest auto-generated version. Scope settings could identify those parts of the map, which were changed since the last synchronisation. These parts could be extracted and merged with the latest auto-generated version. Precondition is proper assignment of subject identities and careful use of base names.

The three approaches have a direct impact on when and how often to auto-generate the topic map. Again, we have several options:

- Only one batch run: The auto-population batch process is performed only once. This is suitable when either both sources and topic map are static or only the topic map will be maintained afterwards.

- Repeated batch runs: The auto-population batch process is performed either periodically (e.g., every night, every weekend, once a month, etc.) or on demand (e.g., new version of sources was released). This reflects that the sources change quite often and that the changed have to be mirrored in the topic map.

- Hot sync: The system(s) containing the sources and the topic map application are integrated in a way that every relevant change in the sources is directly propagated to the topic map application, which updates the map immediately. The integration can be that tight that the topic map software does not even store topic map data, but extracts it on-the-fly from the systems containing the sources. Such an application provides a unified topic map view on all integrated systems and their sources – even if those have never ‘intended’ being part of a topic map. As interesting as the approach looks like, runtime and performance problems might easily become an issue in such a ‘very’ hot sync solution.

A topic map generated from an ontology or classification schema contains many topics and associations but might not contain any occurrences – because ontologies and classification schemas do not care about links to resources. Creating all the occurrences manually would be a painful and expensive task. What is needed is an approach to find corresponding resources automatically and to generate the occurrence links either for part of the map or to generate them on-the-fly when needed. The latter has the advantage that the sources are allowed to change continuously – a quite typical scenario.

An intelligent search engine could be used for this task. It takes the topic names and finds with the support of linguistics algorithms (e.g., stemming) all occurrences of the names in the sources. The sources could be arbitrary text in any document format the search engine ‘understands’. An improved variant of the ‘on-the-fly’ approach considers also names of related topic by following the class-instance, superclass-subclass, and other selected associations. It broadens the ‘query’ for the names and has therefore to be connected with proper ranking of the search results. The ranking calculation is based on the ‘distance’ of the found name to the current topic.

# 5 Application Scenarios

So far, we showed why topic map are useful, explained their concepts, and described how to create them. But we have not talked a lot about possible topic map application scenarios.

## 5.1 *Subject classification*

The most obvious application and already mentioned in Chapter 4 *How to create a topic map* are subject classifications or classification schemas. Classifications are a major approach to organise resources and to simplify access to them. They are a key feature of knowledge management.

The topic map fulfils two functions at the same time: it represents the classification schema with its classes, class hierarchy, class codes, and cross relationships and it assigns the resources into the schema. As these occurrences can point into various repositories, a topic map based classification can easily span multiple systems providing one classified view on all resources.

A scoped topic map or multiple topic maps can model different classifications of the same resources showing the different aspects of the resource by such a multi-dimensional classification.

Users can navigate the classification in a topic map navigator to find resources. But they can also browser the resources and see the assigned classifiers as well as related classes.

## 5.2 *Knowledge representation and ontologies*

The British Virgin Islands examples represented the knowledge about this specific application domain. It is a simple ontology, an explicit model of the domain knowledge.

Such an ontology providing explicit access to explicit knowledge structures to:

- navigate the knowledge structures;
- visualise the knowledge structures as graphs;
- query the knowledge structures;
- derive new knowledge structures through inferencing;
- connect resources from various repositories to knowledge structures;
- analyse the knowledge structures with e.g., with statistics;
- publish or sell the knowledge structures.

A typical application of knowledge representation is the corporate memory used in enterprise knowledge management. It models the knowledge about products, projects, people, policies, processes, and practices and provides it to employees. Again a topic map offers an ideal means of presenting and accessing the 'corporate memory'.

An other application is B2C web sites (online shops). The represented knowledge about sold goods and services are the base technology behind an intelligent virtual sales assistant. Such a virtual sales assistant helps the customer to explore the stock, navigate in it, ask queries and get proper answers, and to communicate with the B2C system via feedback dialogs.

It is a good idea to distinguish between light-weight ontologies and heavy-weight ontologies. Topic maps are seen as light-weight ontologies, because they are able to model knowledge in a very 'shallow' way – e.g., just topics, their classes, occurrences, and associations, but no class hierarchies, constraints, or inference rules. Even 'shallow' topic maps are already very useful without having put large investments in their creation. Heavy-weight ontologies, by contrast, contain class hierarchies, constraints, and inference rules. It takes a long time and many resources to develop and maintain them and if the benefit gained from this extra effort will ever pay of is uncertain.

## 5.3 Publishing knowledge networks

Publishing knowledge networks is about selling added-value – the core business of a publisher. Publishers – commercial and corporate publishers – gather, verify, assemble, and distribute resources as publications. The new information age forces them to change their publishing paradigm from product-centric to information-centric. The information is what's focused on and must be continuously updated. The various publications can be generated out if the central information pool through various channels on different media.

A publisher can make use of topic maps in two ways:

- Supporting the editorial work as well as the information selection process and
- enriching the information published online.

Editorial work and the selection process benefit from subject classification and similar techniques. Enrichment of online information (= resources) is mainly about optimising access to the resources. This covers clever navigation paths, queries, and hyperlinks.

Topic maps are key to interactive access. They provide different views on the same content (resources) e.g., personalised views based on user profiles or even user defined publications. However, they are also able to provide one view on many different resources from various repositories. Their feature separating resources from the topic map data enables business models in which content (= resources) and added-value (= topic map) are created, packaged, syndicated, and sold independently.

## 5.4 Search engines

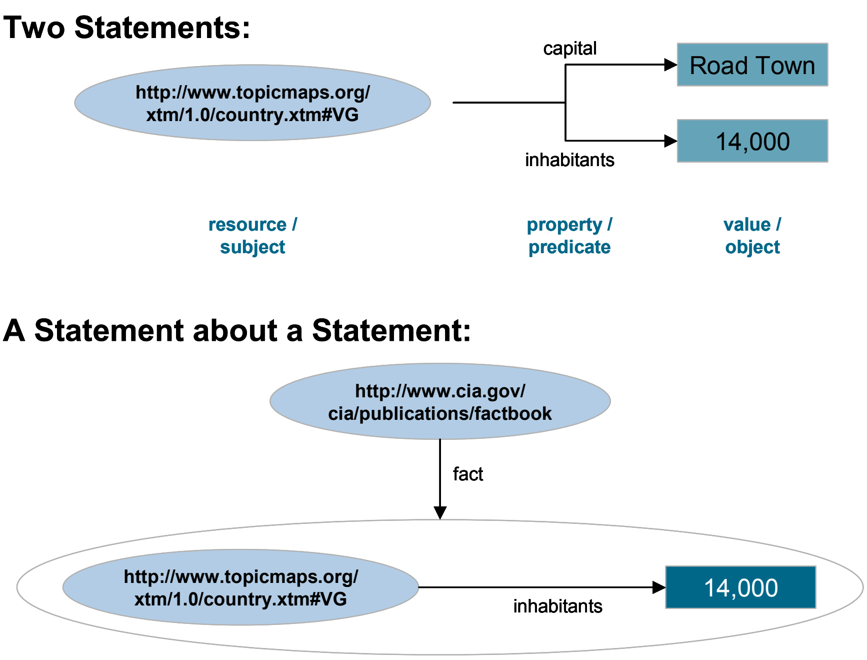
'Searching powered by topic maps' could be the slogan of search engines already using the paradigm to improve their query results. Intelligent 'find' technologies are the result. But not every search technology is prepared to benefit from topic maps. Only those, which make use of an explicit knowledge model, can easily migrate to topic maps. Others based on e.g., statistical algorithms just have no knowledge model and cannot consequently apply the paradigm.

Typical knowledge models of intelligent search engines are based on concept hierarchies with synonyms and maybe weighted similarities between concepts. E.g., Case-Based Reasoning – a result of Artificial Intelligence research – is based on such models. With topic maps, the knowledge models could be represented in a standardised notation instead in a proprietary format.

## 5.5 Application integration, semantic web

Knowledge models and ontologies not only help us to find the requested information, they can also help machines – computer programs –find the necessary data. They can even enable automated electronic data interchange. Which is an important issue in application integration. Nowadays, data interchange and communication between software is based on syntax and its semantic, that both parties have agreed on. But as more computers are connected e.g., via the Internet, the more challenging it is to connect them, let them communicate, and do useful things.

**Figure 20** RDF examples



It is the vision of the Semantic Web, an initiative of the World-Wide Web Consortium W3C, to enable ontology-driven communication between software agents without human intervention. Every agent can fulfil a certain task. They can connect on-the-fly with other agents to cooperatively solve more complex tasks. The agents find each other and interchange and interpret data through ontologies.

Currently, the W3C builds it semantic web architecture on RDF (Resource Description Framework [26], a W3C Recommendation) and OWL (Ontology Web Language [24], a W3C project based on DAML+OIL [1]). RDF has several similarities with topic maps: Both are able to model complex metadata and ontologies as multi-dimensional graphs using XML as interchange format.

The basic concept in RDF is a statement of the form resource-property-value also named subject-predicate-object (see Figure 20). The RDF resource is mainly the same as a topic's subject identity, the property can model a topic name, occurrence, association, or a scope, and the value can model a name string, occurrence reference, occurrence data, or a topic. The illustrated example has two statements about the British Virgin Islands: "British Virgin Islands have the capital Roadtown" and "British Virgin Islands have 14,000 inhabitants". RDF supports statements about statements to express more complex relationships (e.g., "The CIA World Factbook states the fact that the British Virgin Islands have 14,000 inhabitants"). RDF is based on a simple mathematical model, which allows complex logic operations (Description Logics) and inferencing to deduce 'new' knowledge.

However, RDF is a very general paradigm with nearly no predefined semantics. RDF is therefore extremely flexible and powerful, but requires a lot of conventions when certain abstractions have to be expressed and processed. This becomes especially true when the ontology should be visualized for a human being.

A user will be extremely confused when he sees all the arcs connecting nodes; some of these arcs are important for the application domain while others are just there for "bookkeeping" purposes. What is needed is a convention – a schema – which helps to distinguish between application-relevant and application-irrelevant RDF statements. This is where topic maps come in. Topic maps could be seen as an application on top of RDF defined by one of RDF's schema languages – preferably OWL.

The pre-defined topic map concepts name, occurrence, scope, and association make topic maps more suitable for display to users. To summarize the RDF vs. topic maps comparison: Topic maps are for humans, RDF is for the machines.

## 5.6 Who is using topic maps?

Topic maps are quite new phenomenon, but several industries already apply them or will soon make use of them. Their flexibility and expressiveness as well as the fact that topic maps are an ISO standard makes them very attractive.

Commercial publishers are very interested, because topic maps give them a standard at hand to add value to their content. Encyclopaedia publishers, legal publishers as well as publishers from the e-Learning, media, and news domains are early adopters of XTM.

Web portal provider use topic maps to organise their web site and to provide clear and consistent navigation patterns.

The industry applies topic maps in call centres supporting the call centre agent or directly the customer. So-called knowledge gateways provide answers to many typical questions the customer has about the products and services. A corporate memory is a further application in the industrial sector. And very innovative companies use topic maps already as next generation content management paradigm or are at least investigating the approach.

# 6 The family of topic map standards

## 6.1 Brief history of topic map standardisation

The development on topic maps started in 1991. The Davenport Group – a consortium of UNIX system vendors and others, including the publisher O'Reilly & Associates – was established to develop a framework for the electronic management of software manuals. One subgroup worked on the task to define an SGML DTD to mark-up the content of the manual – the outcome of this subgroup's work is the well-known DocBook DTD [2]. The second subgroup had a task to develop an electronic 'master index' for X-Windows (the user interface you still use on UNIX and LINUX machines) manuals of the vendors and an X-Windows book from O'Reilly. The master index would become the neutral and consistent source for all derived vendor specific indices. It would be created by merging existing indices. As it seemed like a trivial task – compared to the difficult task of the other subgroup agreeing on a DTD between different vendors – the group called the to-be-developed model humorously SOFABED (Standard Open Formal Architecture for Browsable Electronic Documents). But it took nine years until ISO published the topic map standard ISO/IEC 13250 in January 2000 manifesting the concepts originally desired by the SOFABED model.

What happened in the meantime? In 1993, SOFABED was passed over to an-other group called 'Conventions for the Application of HyTime (CApH)' knowing that the – now recognised – complexity of describing an index in electronic form would be a perfect application of the sophisticated hypertext facilities of the ISO/IEC 10744 HyTime standard. The CApH group saw the stunning possibilities far beyond just modelling indices, elaborated SOFABED, and renamed it Topic Maps. In 1995, ISO accepted the model as 'New Work Item' and in December 1999 the national bodies of the ISO committee JTC1 SC34 as International Standard ISO/IEC 13250 approved it. The standard specifies the topic maps concepts and defines the interchange format allowing different applications to exchange topic map information in a standardised manner.

But this is not the end of the story. The ISO topic map standard published 2000 was based on SGML (ISO 8879) and HyTime and was as such not really Web enabled – a 'must' for an IT standard with the aim of organising information. The independent organisation TopicMaps.Org was founded in beginning of 2000 with the goal of specifying topic maps based on the W3C recommendations XML and XLink. The group worked very efficiently and was able to publish a core version of the XTM (XML Topic Maps) specification in December 2000 and the official XTM 1.0 specification in March 2001. The XTM 1.0 DTD was passed over to ISO, which approved a Technical Corrigenda of ISO/IEC 13250 [11] in October 2001 making the XML/XLink based XTM notation part of the ISO standard.

## 6.2 Vertical applications at OASIS

After finishing XTM 1.0, TopicMaps.Org decided to focus on vertical domain applications and marketing of the topic map paradigm – topic map standards work was completely left to ISO SC34. In August 2001, three OASIS Technical Committees were formed:

- Topic Maps Published Subjects [34]: Develops a recommendation how to define, document, publish and apply published subjects.
- Topic Maps Published Subjects for Geography and Languages [33]: Defines sets of published subjects for languages and geographical objects based on existing publications (e.g., ISO, UN).
- Vocabulary for XML Standards and Technologies [35]: Defines a vocabulary (set of published subjects) for the domain of XML standards and technologies providing a reference set of topics, topic types, and association types.

OASIS is open to establish further Technical Committees defining vocabularies for other vertical domains.

A further OASIS Technical Committee will be formed beginning of 2003. It will develop an XTM conformance test suite for the purpose of testing the standards conformance of the various available topic map engines.

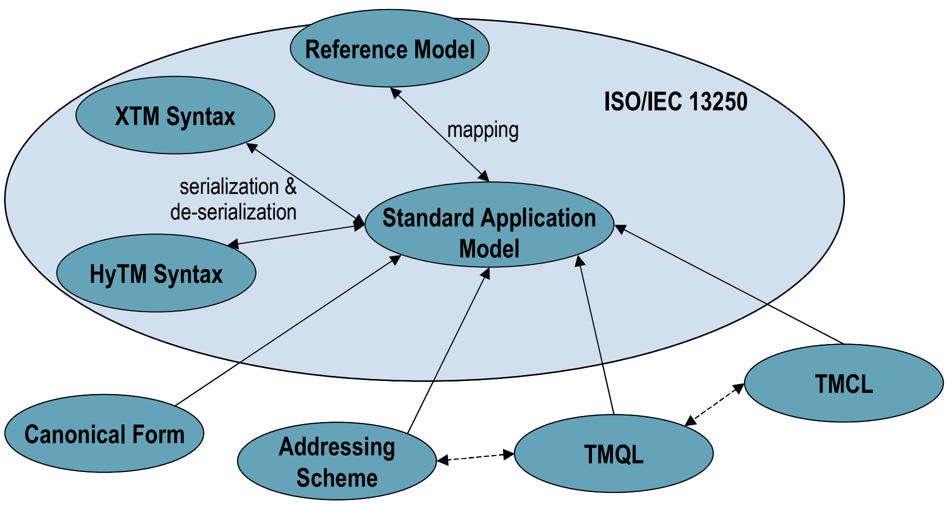
## 6.3 Latest standards work at SC34

The ISO committee JTC1 SC34 started two further standard initiatives in 2001: Topic Map Query Language (TMQL, ISO/IEC 18048) and Topic Map Constraint Language (TMCL, ISO/IEC 19756). Both are still under development and first drafts will probably not be published before 2003.

The committee also decides to define two data models for the topic map paradigm: the Reference Model (RM) and the Standard Application Model (SAM). The groups felt that already, right after publishing the interchange syntax, that a concise data model is necessary to develop further standards on top of the core standard. Moreover, a precise data model also helps software developers to better understand the topic map paradigm and to avoid misinterpretation of the standard's text.

Parsing of the two interchange formats HyTM (SGML/HyTime notation) and XTM (XML/XLink notation) and the generation of these formats is defined in terms of the Standard Application Model. Both TMQL as well as TMCL are also defined in terms of the Standard Application Model. A canonical form and an addressing schema are on the committee's agenda. Figure 21 shows how all these standards will work together.

**Figure 21** Overview about the family of topic map standards




### 6.3.1 *ISO/IEC 18048 – Topic Map Query Language TMQL*

TMQL aims to get the same status for topic map management systems as SQL has for RDBMS – a standardised interface to query, create, and update a topic map. TMQL will become a two-part standard: Part 1 will define how to query topic maps and Part 2 how to create and update topic maps. The standard will cover particularly distributed topic map applications in a global network (e.g., the internet). Figure 22 illustrates how TMQL could be applied – together with SQL – in a 3-tier system architecture.

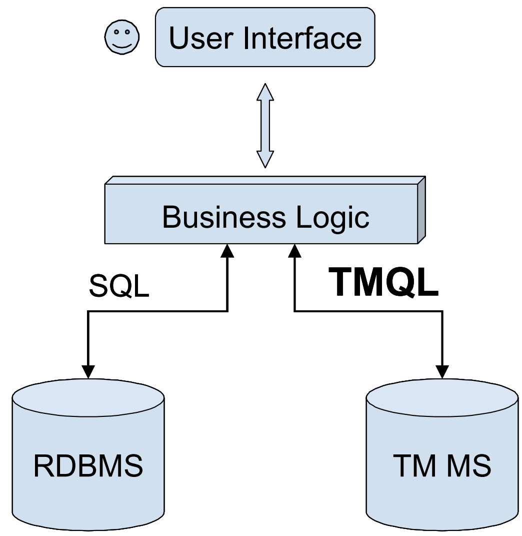



**Figure 22** TMQL in a 3-tier architecture

### 6.3.2 *ISO/IEC 19756 – Topic Map Constraint Language TMCL*

TMCL will define a framework for the definition of topic map schemas for vertical or domain specific applications. It will enable semantic validation and guided editing of topic maps. The TMCL requirements document explains the goal of the standards project.

*TMCL shall permit the definition of classes of topic maps in order to:*

*enable the documentation of the structure and semantics of a class of topic maps;*

*provide a foundation for defining vertical or domain specific applications of topic maps;*

*provide means of validation to ensure consistency within a topic map or across a class of topic maps;*

*enable applications to provide easier and more intuitive user interfaces for creating and maintaining topic maps;*

*enable the separation of the tasks of modelling and populating topic maps.*

*ISO JTC1 SC34, N0226 [12]*

### 6.3.3 Topic map data models

The Reference Model (RM) defines the underlying foundation for general assertion structures. It provides the base setting for the topic map paradigm and is independent from any particular syntax – for storage, interchange or other purposes.

*The Reference Model provides a robust and predictable basis for collating ('aligning' or 'merging') knowledge about subjects, regardless of the diversity of the ontologies that govern the understanding of such knowledge.*

*In the draft Reference Model, a topic map is seen as a set of 'assertions', no more and no less. Each assertion asserts the existence of a strongly typed relationship between some specific set of subjects of conversation.*

*ISO JTC1 SC34, N0298 Rev. 1 [13]*

The Standard Application Model (SAM) defines the topic map data model in terms of an info set. The Standard Application Model is based on the Reference Model, and is as such closer to the topic map concepts and its implementation in a software application. To be concrete, the Reference Model defines the structure of arbitrary assertions and the Standard Application Model defines those specific assertion types, which are part of topic map paradigm (e.g., name, class-instance, occurrence, association, scope).

*The Standard Application defines virtually all of the familiar features of the Topic Maps paradigm, including topic names, topic occurrences, and scopes.*

*ISO JTC1 SC34, N0298 Rev. 1 [13]*

# 7 Conclusions and Outlook

Topic maps are the 'GPS of the Information Universe'. Coming from back-of-book indices, the paradigm defines the necessary concepts to model explicit knowledge structures over resources. Topic maps are simple, but not too simple. They are light-weight, but have the potential to grow together with the demands of the information age. They are an international ISO standard ensuring stability, reliability, and openness – all important for a secure IT investment. And ISO is working on a family of topic map standards to complete the existing standard with data model, query language, and schema language.

Subjects are in the focus of the topic map paradigm. Subjects can be any thing whatsoever – addressable or non-addressable. The subject's identity is very well defined and establishes the foundation for reliable merging. Topics represent subjects, or better, computerise subjects. Topics have characteristics: names, occurrences, and roles they play in associations. Characteristics can be scoped, which can limit, 'slice and dice' parts of the topic map through the application of validity rules and by this mechanism, model different views of an application domain. A scope is described as a set of topics. Topic names have to comply with the topic naming constraint, which forces merging of topics with the same name in the same scope. Occurrences assign resources to topics. Associations relate topics. Every topic plays a certain role in the association. Roles are topics. Topics, occurrences, and associations can be instances of classes. Classes are topics. Classes can build class hierarchies.

The above paragraph explains – very briefly – the topic map paradigm. But the paradigm is a technology, not a solution. The design, creation and maintenance of topic maps, their implementation in topic map software, and its integration with other software components are all necessary tasks to building a complete solution. The topic map cookbook proposed a step-by-step analysis, design, testing, and documentation of a topic map schema – consisting of classes and constraints. The population of the topic map with instances of the classes can be done manually using topic map authoring software. Or it can be the result of an auto-generation process, which transforms information from various systems into topic maps.

Possible topic map applications range from a simple electronic index or thesaurus over subject classification and web site organisation up to knowledge representation and ontologies. Topic maps can be the 'brains' of intelligent search engines and bring application integration to the next level – the vision of the semantic web. Corporate memory and other key aspects of a knowledge management solution can also benefit from topic maps. Although a quite new standard, topic maps have left the 'ivory-tower' of theoretic thinking and become a 'real world' phenomenon – with real products, real projects, real solutions, and a growing list of user experience.

What's next? How will the topic map future evolve? Let's take the look in our crystal ball.

The obvious things first. ISO will complete the data models (Reference Model and Standard Application Model) in 2003. The work on TMQL and TMCL will continue with the availability of the Standard Application Model, as both have to be defined in terms of that model – publications of committee drafts can be expected in 2003/2004, approved standards probably not before 2004/2005.

The OASIS TCs Published Subjects, GeoLang, and XMLvoc will probably finish their work in 2003. A new TC will be formed (beginning of 2003) to define an XTM conformance test suite. More OASIS TCs will be established to define further vocabularies for vertical domains. Public organisations (e.g., of the United Nations) might start to register concepts they use in their classification schemas. Important industry communities relying on proper information interchange will follow (2003/2004).

Now, the difficult predictions about tools and customers. Several companies and public organisations will start projects – or at least pilot projects – using topic maps. Classification will be the main purpose of such applications. Especially commercial publishers will utilise topic maps to organise editorial work and to enrich online publications. More web sites will use the paradigm to organise their pages and to have a flexible solution with shorter time-to-market benefit. Industry will apply topic maps in two major domains: call centres and corporate knowledge portals.

Tools dedicated to the development, maintenance, and access to ontologies will more increasingly support topic maps – at least as import / export format together with other formats like RDF. That both import and export comply with the full standard is an important requirement – export is critical as it secures the investments made in the ontology development. Software developed merely for the support of topic maps will further improve and support other ontology formats as well. Performance and scalability of topic map engines will increase as well as their support for query and schema languages. But the fasted growing topic maps market will be in product categories where topic maps are supporting the key features, but are not the key feature of the product. These categories are search engines, integrated knowledge management systems, and content management systems.

How quickly we achieve this 'manifest destiny' depends on how quickly we resolve the issue of competing formats. Without doubt, RDF, DAML+OIL, and OWL are extremely high profile in the Web context. As W3C developments, they are pushed by the members of the W3C and academic research groups. They will become the foundation of the semantic web. But topic maps have certain advantages over RDF & Co. and the communities are in close contact to converge the formats – either by integrating or by layering them. The latter is more likely. When this is achieved, topic maps will be an essential part of the next generation Web and many knowledge-based applications.

# 8 Glossary

| | |
|---|---|
| addressable subject | A subject↑, which is a resource↑ directly addressable by a URI↑.<br>See also *non-addressable subject*. |
| association | A characteristic↑ of a topic↑. An asserted relationship between an arbitrary number of subjects↑, each represented by a topic↑ playing certain roles↑ in the association. |
| association class | A topic↑ of which a particular association↑ is an instance of. The topic's subject↑ is the class↑ of the association. |
| association role | The role played by the topic↑ (= role player↑) being associated by the association↑. |
| association type | See *association class*. |
| base name | A characteristic↑ of a topic↑. Assigns a name (= string of characters) to a topic. Has to conform to the topic naming constraint↑. |
| characteristic | See *topic characteristic*. |
| class | A group, set, or kind sharing common attributes (Merriam-Webster's Collegiate Dictionary).<br>Role↑ in an class-instance association↑ played by the topic↑ being the class.<br>See also *instance*. |
| class-instance association | Assertion stating that one topic↑ playing the class↑ role and another topic playing the instance↑ role establish a class-instance relationship. It is normally expressed by the XTM element `<instanceOf>` but could be expressed by an `<association>` element as well utilising the pre-defined PSIs. |
| constraint | Rule(s) governing classes↑ of objects in an ontology↑ for the purpose of validation and/or guided topic map↑ authoring. |
| display name | Variant name↑ with the purpose of device-specific name display. Can be a string of characters or a reference to an external resource (e.g., an image).<br>See also *base name* and *sort name*. |
| instance | An individual illustrative of a class↑.<br>Role↑ in an class-instance association↑ played by the topic↑ being the instance.<br>See also *class*. |
| HyTM | SGML/HyTime-based interchange syntax for topic maps.<br>See also *XTM*. |
| merging | Process of joining two topics↑ or joining two topic maps↑. Two topics are merged if they comply to the topic naming constraint↑ or if they have the same subject identity↑. Merging of topic maps is triggered by the XTM element `<mergeMap>`. |
| merge | See *merging*. |

| | |
|---|---|
| name | See *base name*. |
| non-addressable subject | A subject↑, which is not an addressable resource↑.<br>See also *addressable subject*. |
| occurrence | A characteristic↑ of a topic↑. Assigns a resource or a string of characters to the topic – both should be somehow relevant to the topic's subject. |
| occurrence class | A topic↑ of which a particular occurrence↑ is an instance of. The topic's subject↑ is the class↑ of the occurrence. |
| occurrence type | See *occurrence class*. |
| ontology | An explicit specification of a shared conceptualisation (T. Gruber). |
| parameter | Defines the purpose of a variant name↑. |
| PSI | See *Published Subject Identifier* and *Published Subject Indicator*. |
| Published Subject Identifier | Published URI↑ representing a non-addressable subject↑. |
| Published Subject Indicator | Human readable resource a Published Subject Identifier↑ resolves to. Resource describes, indicates a non-addressable subject↑. |
| Reference Model | Reference Model. Meta-model for the topic maps paradigm. |
| reification | Process of creating a topic↑ for a subject↑.<br>Process of creating topics↑ for occurrences↑ and associations↑ to assign topic characteristics↑ to them. |
| reify | See *reification*. |
| resource | Can be represented as a sequence of bytes, and thus could potentially be retrieved over a computer network. Typically, it is addressed by a URI↑. |
| RM | See *Reference Model*. |
| role | See *association role*. |
| role player | One of the topics↑ being associated by an association↑. |
| SAM | See *Standard Application Model*. |
| scope | Specifies the context within which a topic characteristic↑ is valid. It is a set of topic references (= scoping topic↑ or theme↑) that together define the context. See also *unconstrained scope*. |
| scope set | See also *scope*. |
| scoping topic | See also *theme*. |
| sort name | Variant name↑ with the purpose of providing a sort key for a topic. See also *base name* and *display name*. |
| Standard Application Model | Defines the topic map data model in terms of an infoset. Is based on the Reference Model↑. |
| subject | A 'subject' is any thing whatsoever, regardless of whether it exists or has any other specific characteristics, about which anything whatsoever may be asserted by any means whatsoever. See also topic↑. |
| subject identity | The identity of a subject↑ ensuring to distinguish it from all other subjects. A topic↑ uses the subject identity to uniquely address its subject. The identity of a subject can be established by a resource↑ (addressable subject↑) or by a subject identifier↑ (non-addressable subject↑). |

| | |
|---|---|
| subject identifier | URI↑ representing a non-addressable subject↑. |
| subject indicator | Human readable resource↑ a subject identifier↑ resolves to. Resource describes, indicates a non-addressable subject↑. |
| subclass | A class↑ that is derived from a particular class, perhaps with one or more classes in between..<br>Role↑ in an superclass-subclass association↑ played by the topic↑ being the subclass.<br>See also *superclass*. |
| superclass | A class↑ from which a particular class is derived, perhaps with one or more classes in between.<br>Role↑ in an superclass-subclass association↑ played by the topic↑ being the superclass.<br>See also *subclass*. |
| superclass-subclass association | Assertion stating that one topic↑ playing the superclass↑ role and an-other topic playing the subclass↑ role establish a superclass-subclass relationship. It is expressed by an `<association>` element utilising the pre-defined PSIs. |
| template | Declaration of a class↑ and its constraints↑ to which instances↑ of the class must conform. |
| theme | A topic↑ that is a member of a scope set↑. |
| TMCL | See *Topic Maps Constraint Language*. |
| TMQL | See *Topic Maps Query Language*. |
| TNC | See *topic naming constraint*. |
| topic | The computer representation of a subject↑. A topic is the reification↑ of a subject. |
| topic characteristic | The names↑, occurrences↑ of a topic↑ and the roles↑ the topic plays in associations↑. |
| topic class | A topic↑ *C* of which a particular topic↑ *T* is an instance of. The topic *C*'s subject↑ is the class↑ of the topic *T*. |
| topic map | A collection of topics and associations between them. It exists in interchangeable syntax form (HyTM↑, XTM↑) or in a processed data model form (e.g., SAM↑) in a topic map processor. |
| topic map schema | The collection of templates↑ that together define a class↑ of topic maps↑. |
| Topic Maps Constraint Language | A framework for the definition of topic map schemas↑ for vertical or domain specific applications to be published in ISO/IEC 19756. |
| Topic Maps Query Language | A standardised interface to query, create, and update a topic map↑ to be published in ISO/IEC 18048. |
| topic naming constraint | Topics↑ having the same base name↑ in the same scope↑ implicitly refer to the same subject↑ and therefore should be merged↑.<br>Note: The topic naming constraint will become optional in the next edition of the topic map standard. |
| topic type | See *topic class*. |
| type | See *class*. |
| unconstrained scope | The scope set↑ of a characteristic↑ is empty and the characteristic is considered to have unlimited validity. |
| Uniform Resource Identifier | A compact string of characters for identifying an abstract or physical resource (IETF RFC 2396). |
| URI | See *Uniform Resource Identifier*. |

**variant name** — An alternate form of base names↑ for display↑, sorting↑ of topics, or other – application dependent – purposes.

**XTM** — See *XML Topic Maps*.

**XML Topic Maps** — XML/XLink-based interchange syntax for topic maps.

See also *HyTM*.

# 9 Resources

The handbook's text contains many references to literature for further reading or consortia/committees to get in contact with.

Some comments on the resources:

- The only available book about topic maps is (as of January 2003) *XML Topic Maps*, published by Addison Wesley [15]. It contains a collection of topic maps articles about history, concepts, underlying data models, their visualisation, how to create a topic map, topic maps in life science, e-learning, web site management, knowledge representation, their relationship to RDF and the Semantic Web.
- A quite active online discussion group about topic maps is the Topic Map Mailing [19]. Beginners and experts around the world discuss various issues about topic maps.
- If you want to be involved in topic map standardisation become a member of ISO/IEC JTC1 SC34 [31]. You have to be part of a national body delegation (e.g., of Standards Australia) or a member of one of the SC34 liaisons. Contact Dr. James D. Mason <masonjd@y12.doe.gov> for further details.
- If you want to be involved in standardisation of topic map applications and PSI sets contact one of the chairs of the OASIS Technical Committees [33][34][35].

## 9.1 Literature


[1] *DARPA Agent Markup Language + Ontology Inference Layer (DAML+OIL)*, http://www.daml.org/ and http://www.ontoknowledge.org/oil/

[2] *DocBook*, http://www.docbook.org/

[3] *Dublin Core*, http://dublincore.org/

[4] Gartner Group: *Topic Maps: Emerging Knowledge Management Technology*, Research Report, June 2000.

[5] Gruber, T.: *What is an Ontology?*, http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

[6] Internet Engineering Task Force (IETF): *Uniform Resource Identifiers (URI)*, http://www.ietf.org/rfc/rfc2396.txt

[7] International Organization for Standardization (ISO): *International Classification for Standards (ICS)*, http://www.iso.org/iso/en/prods-services/otherpubs/Informationexchange.html#ics

[8] International Organization for Standardization (ISO): *ISO 639:1998 Code for the representation of names of languages*, 1998.

[9] International Organization for Standardization (ISO): *ISO 3166:1997 Codes for the representation of names of countries and their subdivisions, part 1–3*, 1997.

[10] International Organization for Standardization (ISO): *ISO/IEC 13250:2002 Topic Maps (2nd edition)*, http://www.y12.doe.gov/sgml/sc34/document/0322_files/iso13250-2nd-ed-v2.pdf

[11] ISO/IEC JTC1 SC34: *N0220, Defect Report on Topic Maps (ISO/IEC 13250:2000)*, http://www.y12.doe.gov/sgml/sc34/document/0220.htm

[12] ISO/IEC JTC1 SC34: *N0226, Draft requirements, examples, and a "low bar" proposal for Topic Maps Constraint Language (TMCL)*, http://www.y12.doe.gov/sgml/sc34/document/0226.htm

[13] ISO/IEC JTC1 SC34: *N0298 Rev. 1, A High-level Description of a Draft Reference Model for ISO 13250 Topic Maps*, http://www.y12.doe.gov/sgml/sc34/document/0298R1.htm

[14] META Group: *Topic Maps: An Emerging Content Navigation and Metadata Metaphor*, Research Report, January 2001.

[15] Park, J; Hunting, S. (eds): *XML Topic Maps*, Addison Wesley, ISBN 0-201-74960-2, 2003.

[16] Sowa, J.F.: *Knowledge Representation*, Brooks/Cole, ISBN 0534-94965-7, 2000.

[17] Sowa, J.F.: *Semantic Networks*, http://www.jfsowa.com/pubs/semnet.htm

[18] Standards Australia International Ltd: *Knowledge Management: A Framework for succeeding the knowledge era*, HB 275–2001, ISBN 0-7337-3903-2, 2001.

[19] Topic Map Mailing List (hosted by Infoloom): Subscribe at http://www.infoloom.com/mailman/listinfo/topicmapmail

[20] TopicMaps.Org: *XML Topic Maps (XTM) 1.0 Document Type Declaration*, http://www.topicmaps.org/xtm/1.0/xtm1.dtd

[21] TopicMaps.Org: *XML Topic Maps (XTM) 1.0 Specification*, http://www.topicmaps.org/xtm/

[22] World-Wide Web Consortium (W3C): *eXtensible Markup Language (XML)*, http://www.w3.org/XML/

[23] World-Wide Web Consortium (W3C): *Hypertext Transfer Protocol (HTTP)*, http://www.w3.org/Protocols/

[24] World-Wide Web Consortium (W3C): *Ontology Web Language (OWL)*, http://www.w3.org/TR/owl-ref/

[25] World-Wide Web Consortium (W3C): *RDF Vocabulary Description Language 1.0: RDF Schema*, http://www.w3.org/TR/rdf-schema/

[26] World-Wide Web Consortium (W3C): *Resource Description Framework (RDF)*, http://www.w3.org/RDF/

[27] World-Wide Web Consortium (W3C): *Semantic Web*, http://www.w3.org/2001/sw/

[28] World-Wide Web Consortium (W3C): *XLink*, http://www.w3.org/TR/xlink/

[29] World-Wide Web Consortium (W3C): *XPointer*, http://www.w3.org/TR/xptr/

[30] World-Wide Web Consortium (W3C): *XML Schema*, http://www.w3.org/TR/xmlschema-0/


## 9.2 Consortia, committees


[31] ISO/IEC JTC1 SC34, contact chair Dr. James D. Mason <masonjd@y12.doe.gov>.

[32] ISO/IEC JTC1 SC34 WG3, http://www.isotopicmaps.org/

[33] *OASIS TC Published Subjects for Geography and Languages (GeoLang)*, http://www.oasis-open.org/committees/geolang/

[34] *OASIS TC Topic Maps Published Subjects (PubSubj)*, http://www.oasis-open.org/committees/tm-pubsubj/

[35] *OASIS TC Vocabulary for XML Standards and Technologies (XMLvoc)*, http://www.oasis-open.org/committees/xmlvoc/

[36] *World-Wide Web Consortium (W3C)*, http://www.w3.org/

# About empolis

Information is the basis for decisions, a prized commodity, a company value and the basis for our communication. Quick and easy access to the required information, its structures, its links and distribution networks is often more essential than the information itself. This applies to both internal organizational processes and to the dialog with the customer.

empolis is the leading supplier of software solutions with extensive industry experience and corresponding services for enterprise content and knowledge management. With the empolis product lines, e:kms, e:catalox, and e:solutions, an organization's information is transformed into indispensable and profitable capital that achieves a fast return on investment.

empolis has influenced the development of innovative industry standards such as XML, Java and Topic Maps and, in turn, these standards have decisively influenced the development of empolis products for the benefit of our customers.

empolis is part of arvato – a Bertelsmann company – and employs 300 people in Germany, Hungary, Scandinavia, Poland, UK and the US.

Contact: info@empolis.com and www.empolis.com