Representation and Navigation Techniques for Semi-Structured Knowledge in Collaborating Communities

Zsolt Kemény[1], **Gábor Erdős**[1], **József Váncza**[1, 2]

[1]Computer and Automation Institute, Hungarian Academy of Sciences, H-1111 Budapest, Kende u. 13–17, Hungary
[2]Budapest University of Technology and Economics, H-1111 Budapest, Műegyetem rkp. 3–9, Hungary
{kemeny,erdos,vancza}@sztaki.hu

Abstract The paper addresses problems inherent to gathering, managing and browsing knowledge relevant for maintaining and serving a collaborating group with common interests, i. e., a *knowledge community*. First, current solutions for information management will be examined, highlighting the need of more flexible means of information storage and retrieval. Hereafter, two of the most common – currently available – paradigms, i. e., semantic web technologies and topic maps, will be presented in an overview, and finally, the most suitable of these will be explained in a practical example.

Keywords: Knowledge community; Semantic web; Topic map

1 Introduction

1.1 Motivation: Towards a Knowledge Community

In this paper we discuss the problem of how to make information, and later on, knowledge resources of a research community, usable and accessible to the members of the group in concern. Our work is specifically motivated by the objectives of the *Virtual Research Laboratory for a Knowledge Community in Production* (VRL KCiP). Research entities – both individual and organizational – associated in this network are aimed at creating a common culture for information, and eventually, for knowledge sharing [21]. VRL KCiP is going to realize a new kind of academy-industry cooperation as well, as it is meant to provide intellectual support to industrial partners in the development, design, manufacturing and marketing of high-tech products.

In its formation stage, members of the VRL KCiP community constituted a loosely connected network of various organizations and people, each having

A. Bernard, S. Tichkiewitch (eds.),
Methods and Tools for Effective Knowledge Life-Cycle-Management,
DOI: 10.1007/978-3-540-78431-9_11, © Springer 2008

competencies and disposing over some information resources in distinct – mostly design and production related – domains. The network is, however, large and distributed; competencies of people are partly implicit and most of their information resources are hidden.

In general, we assume a *community of practice* whose members – both individuals and organizations, forming geographically distributed groups – have some partially overlapping areas of interest and expertise. Each community member disposes over various *information resources* such as research papers, reports, technical documents, presentations, web-based and multimedia materials; these distinct information resources being heterogeneous as far as their format, content, amount, quality, stability and relevance is concerned. Members of the community are willing to make well-defined sets of their information resources public and accessible, thus *i)* contributing to a common information repository, and *ii)* giving an expression of their specific expertise and domain knowledge as well. This expression of knowledge is implicit, fragmentary, and far from being coherent across the community.

Our main motivation is *finding an appropriate match* between community members, the information resources they possess and/or contribute to in any way, and their domain knowledge. Our hypothesis is that this mapping is a necessary – albeit not sufficient – condition to a transition *from information to knowledge management* in the community. As a vehicle for this transition process, we suggest to establish a *competence map* of the network which facilitates navigating through the VRL KCiP, finding the right persons and organizations, locating the right information resources (documents), and using them in solving particular problems.

In what follows we give main requirements of a system which could assist in transforming information management into knowledge management. These requirements point towards the application of the *map* metaphor. We discuss alternative representation technologies (semantic web and topic maps) and argue in favour of the latter. Finally, by making use of the topic map technology, we demonstrate a novel way for organizing and presenting the domain knowledge of a classical design problem, the *synthesis of mechanisms*.

1.2 Competence Map

A distributed and potentially incoherent collection of information can be turned into a kind of knowledge repository if we can attach a *semantic annotation* to the information resources which is accepted, developed, shared and used by members of the community.

The primary method for structuring information is the development of subject hierarchies, directories, classification schemes or taxonomies. All these are simple yet efficient ways of organizing large volumes of information, especially when coupled with a search function, yet an increasing demand for flexibility quickly sheds light on some severe limitations of any one-dimensional classification:

- It provides only a single point of view for keeping similar objects together, which may lead to difficulties in putting an item into the right class if there are several possibilities. In fact, *the* proper place does not even exist in such a case, which makes either duplication or omission of entries necessary, both cases leading to a truncated or difficult-to-handle model of reality.
- Relationships between objects on the same level cannot be captured, even though these could very well make up the better part of (implicit) knowledge about the given area.
- Only rudimentary visualization, if any, can be supported, even though this would substantially assist numerous cases of manual search.
- Navigation becomes increasingly difficult as the tree structure grows – often, users have the impression that they must already know the tree before they can start searching an item in it.

For representing both the global and local competencies of a community we suggest to employ the metaphor of a *map*. In broad terms, a map is such a representation of a space which *i)* provides an overview and *ii)* highlights relations between components (objects, regions) of that space, as well as *iii)* captures some of its local properties. As far as human users are concerned, a map can be attractive because:

- Relationships between components and various groups of components offer users multiple paths to the same content and *stimulate alternative content exploration*;
- Visualization is supposed to give the user an overall conceptual model, and give a feeling of being in a "relational space";
- While a map is organized following general principles adopted by all of its creators, and accordingly, has an approved structure, it can be modified, augmented and deepened locally.

Whether the map can be used for information storage, indexing and retrieval, or for knowledge management, depends on the way users think of and work with it. We foresee a maturation process that starts at information management and ends up in collaborative knowledge management. In the ideal case, this map can be a vehicle for knowledge diffusion that helps to turn the community of practice – both via its creation and usage – into a so-called collaborative network [19], or knowledge network [15]. To cover the whole spectrum from information to knowledge, we call it *competence map*: a representation that *i)* reveals the underlying relationships of various information and/or knowledge sources using the map metaphor, and, at the same time, *ii)* provides links to these resources. The main requirements towards the *competence map* are as follows:

- Support should be given for constructing individual and organizational profiles with facets dedicated both for internal and external use.
- There is a definite demand of the community for a well-structured organization with controlled content quality.

- At the same time, asynchronous, decentralized profile building and content updating is to be supported.
- Navigation services for human users should be provided: both search and browsing functions are expected.
- Access should be granted to heterogeneous information resources – including research papers, technical documents, presentations and multimedia materials – made public by the network partners.
- The development of a common understanding, and if possible, of a common language among different research groups should be promoted.
- There should be an option to support joint, collaborative work of network partners.

While the first five requirements are essential for making accessible and exchanging information among network partners of the Virtual Research Laboratory, the last two issues are pre-requisites of establishing a knowledge management system for the VRL.

1.3 Related Work: the Map Metaphor in Information and Knowledge Management

The application of the map metaphor in revealing abstract relationships for objects of interest in an information space is by no means a novelty. Below we present some approaches which are closely related to our notion of the competence map.

Concept maps. Concept mapping has a history of use in several disciplines, both as a formal, or more frequently, rather as a semi-formal organization, representation and visualization technique [16]. Structurally, *concept maps* are typed hypergraphs – i. e., generalized graphs where nodes can enclose other nodes (thus edges can connect sets of nodes). Each node has a type, a label and content, which may also be structured itself. Labelling is also extended to links. Visualization attributes are attached to specific node types, allowing an attractive and consistent appearance which can transcend the "flatland" of 2D display [25]. This general structure encompasses a wide range of diagrammatic knowledge representation techniques [4].

The concept map is flexible enough to associate alternative, evolving meaning to complex, even contradictory, sets of information. Recently, it has been used as a computational vehicle for supporting interpretation and argumentation over the corpus of technical documentation stored in distributed digital libraries [26].

The idea of concept maps underlies also the recently developed system *Waypoint* which provides services for information search and retrieval in large engineering document collections [14]. The system implements a *faceted classification* in which different classification categories are assigned to individual concepts, allowing the interconnections of concepts to be traced in several directions. The system has been used in knowledge management applications in engineering companies, as well as in healthcare and archaeology [7].

In contrast, so-called *mind maps* have a simple tree-like structure where nodes are linked by parent-child relations. Mind maps are organized around a focal topic represented by the root of the tree.

Technology and patent maps. It is a common practice of technological intelligence to draw so-called *technology maps* conveying topics of interest, main players and patterns in the development of a particular target technology. In [28], over the same set of information resources – abstract and patent databases – several maps are generated. A principal component map represents the relationships among main concepts, while a keyword map represents the relationships among frequently occurring subject index terms. An affiliation map is used for capturing the relationships of research topics on the level of organizations, based on terms they use in their documents, while there are similar maps for authors, countries and publication forums (such as journals, proceedings etc.). The family of maps allows the user to get an intuitive feeling of research and development activities in a specific domain. Navigation may take various kinds of routes: one can *i)* follow links in professional network, *ii)* have a view of the dispersion of activities on a specific field (e. g., medical applications of nano-technology R&D), or *iii)* identify latent relationships.

Similarly, patent information professionals apply patent mapping methods [3], [2] and set up so-called *patent networks* [27]. The methods look for the co-occurrence of specific terms or keywords; if they crop up together in the documents more frequently than expected, their relationship is represented in the map. Recently, the search for meaningful relationships has been supported by domain-specific ontologies [20].

Technology roadmaps. *Disruptive technologies*, emerging from the interaction of apparently diverse technological advances, change entirely the *status quo* on a market through the introduction of products and services which are dramatically cheaper, better, and more convenient. *Technology roadmaps* are drawn with the specific goal of identifying, developing and implementing such disruptive technologies.

In [10], an integrated roadmap is presented, consisting of research, development, capability, and requirement levels. Interlinked nodes in the research and development levels represent existing or proposed R&D programs, while nodes in the capability level represent necessary capabilities for executing them. Actual goals, objectives and requirements are represented in a separate layer.

As a technique for drawing roadmaps, *database tomography* (DT) has been suggested [9]. DT extracts multi-word phrase frequencies from textual databases, performs phrase proximity analysis, and finally, relying on domain experts, transforms disorganized data into an ordered representation. The roadmap is credible only if it covers all techniques necessary for meeting the requirements, and represents completely the relevant R&D efforts in the given community of practice.

A roadmap consists of multi-attribute nodes and links covering many dimensions; hence, any visualization requires the capability to traverse these dimensions rapidly and easily. Dimensionality is added to the intrinsically 2D visualization by the application of colours, shading and, last but not least, by the use of hyperlink techniques.

Self-organizing document maps. For creating *document maps*, the method of *self-organization* has been applied recently [12]. This method groups various kinds of information resources according to their contents, and maps them to a two-dimensional array of cells. Documents that are considered similar are mapped to the same or neighbouring cells, with links pointing to the corresponding records in the document database. The document map allows a search with soft matches: beyond locating documents matching a given search expression, further relevant matches can be found along the links in the nearby cells, even if they did not meet the search criterion exactly.

For instance, departing from sources available on the Internet, both one-dimensional hierarchical lists and 2D self-organizing maps have been created over news documents related to finance and health [17]. The resulting hierarchical *knowledge map* – so-called *NewsMap* – could be used for browsing business intelligence and medical knowledge hidden in news articles. Comparative empirical studies have shown that the map representation and its visual cues increased the performance of the users considerably.

2 Representation Techniques

As it was already pointed out in the previous section, a knowledge community where experts and distributed knowledge may have to join in new ways for meeting new challenges, often calls for a repository of resources which transcends today's widespread single-classification hierarchies and provides more expressivity with respect to common terms and more flexibility for browsing and directed search. It should be also clear that this capability is a matter of the right *representation* (i. e., the mapping of content elements and their relations onto an image with sufficient expressive power), rather than *presentation* (which determines how the user perceives the image hidden behind a "front end"). If the capabilities of the repository itself are sparse (especially with respect to structure), no presentation can offer the possibility of escaping a dead-end in a search through cross-links, or making oneself a general picture of a domain by browsing. To reformulate these ideas already proposed in the previous section, a practically usable representation of a common domain and its experts, information pool etc. should exhibit the following properties:

- The possibility to depart from the relatively rigid structure of one or more hierarchical trees, i. e., the possibility of *cross-links*, such as "similar to", "sounds like", "see also", "recommended reading" etc.
- Usable definition of objects and relations between them, as well as possible context and rules or statements, which should either facilitate easy human un-

derstanding and browsability or automated inference, whichever is needed in the given case. Note that as for now, browsing by humans is likely to receive more attention than automatic inference.
- The capability of the representation to deal with temporary incompleteness or inconsistency (this deserves strong emphasis especially in domains which are quickly evolving and should go hand in hand with methods for detecting and resolving these flaws of the representation).

The first two requirements may sound familiar in the context of computer science, since these two are named most commonly as key properties of an *ontology*. For the past 15–20 years, ontologies and ontological engineering have been subject to intense research, and some notable practical results have already emerged from these efforts. In the early generation of ontologies, much emphasis was laid on complete machine readability – and not least on consistency and completeness – to allow automatic inference. Also, an ontology was regarded as a more or less *persistent* achievement of *mutual consensus* in understanding terms, rules etc.

The 1990s and the beginning of the 2000s witnessed the emergence of another group of knowledge representation techniques where formal inference was not as much in the focus of interest. In turn therefor, *information exchange, human operation and distributed development* (associated with *merging* and *mediation* between knowledge representations) gained more attention, such concepts as Cyc [13] doing the first steps with managing *microtheories*. This moderation of goals was, most probably, due to the fact that the limits of present-day machine inference still do not allow inference-based applications to permeate everyday technology to a significant degree (and humans have to perform some of the tasks envisaged to be automated), as well as the growing demand for structured information interchange and searchable or human-readable images of semantic content. Also, recent advances appear to pay more attention to the problems of distributed and possibly uncoordinated development as it commonly occurs in loose communities of users and contributors.

Next, two of the most widespread answers to these demands will be described in more detail: the *semantic web* and *topic maps*.

2.1 The Semantic Web and Associated Languages

The term *Semantic Web* refers to distributed resources which are, much like the World-Wide Web, accessible online and linked to each other. What sets the Semantic Web apart is the additional machine-readable *semantic information*, i. e., not only mere data can be accessed in the documents but their *meaning* as well. This is the fundament of such applications as semantic search (as opposed to finding mere keywords or regular expressions in documents or analyzing them statistically at best), automated mediation between different forms of data representation, or, if syntax and semantics allow, even inference by machines. Although the need of the inclusion of semantic information with online resources was already addressed in 1994 at the first World-Wide Web Conference, most of the actual development of Semantic Web technologies and languages has taken place in recent years [1]. Intense research and development has been going on in numerous working groups and communities, resulting in a multitude of possible (competing) solutions, description languages and suggestions for standards or recommendations. Although refinement of the technological background of the Semantic Web still goes on, it became more or less clear by now that one given stack of markup languages is most likely to gain dominant acceptance with the users worldwide; therefore, this set will be now described in detail.

XML. The abbreviation stands for *eXtensible Markup Language*. XML belongs, syntactically, to the SGML (short for *Standard Generalized Markup Language*) family which includes, for example, the *HyperText Markup Language* HTML as well. XML has been created to be primarily used for data format description (although even definition of simple event sequences in robotics or automation are known as application examples), and is now commonly used as such, not only for online resources but also for structuring of documents, embedding exposure information in digital images etc. It should be noted that the URI (*Uniform Resource Identifier*) notation used in XML is defined flexibly enough to address resources outside the World-Wide Web, so that it is possible to use XML to store, e. g., pointers to physical entities (books, institutions, people etc.) and handle them uniformly together with URLs (*Universal Resource Locators*) which do realize online addresses. The underlying ideas of XML are:

- Format description is done by labelling data, i. e., placing a pair of appropriate *opening and closing tags* around the given piece of data, referred to as *element* (optionally, opening tags can be given *attributes* as well, and tagged elements can be embedded into further pairs of tags, too).
- Tags can be *freely defined* (hence the extensibility) upon mutual consensus, so that resources marked with the agreed tags can be interpreted and processed by an XML interpreter which also recognizes the given tags (such as a script embedded in an HTML file which then visualizes the XML-tagged data received);
- A set of tag definitions is grouped into a *namespace*, so that the same name of a tag can have an independent interpretation in all namespaces concerned (specifying the namespace unambiguously identifies how the given tag should be interpreted).
- It is possible to define a fixed *structure* for an XML document either with a DTD (*Document Type Definition*), or an XML schema.

RDF(S). While XML is largely used for data format description only, RDF (short for *Resource Definition Framework*, now accepted by the W3C as a recommendation) and RDFS (*RDF Schema*) venture a further step towards semantic contents. RDF is mainly meant to enrich Web resources with metadata providing semantic information according to the *semantic network formalism* of *resources*, *properties* and *statements*, while RDFS describes relationships between properties and resources [5]. Although there are several suggestions for realizing a possible RDF syntax, XML-based RDF (also known as XML/RDF) is now most widely accepted as the RDF interchange format of Semantic Web resources. RDF can describe the following categories:

- Concepts (classes);
- Individuals (instances) belonging to a given class;
- Properties of classes;
- Specific values of the properties.

RDFS extends these possibilities with:

- Binary relations between properties and resources;
- Domain and range specification for relations.

With these means, it is already possible to build simple taxonomies, and *reification* (representation of relations, statements etc. as instances of a class) allows us to define non-binary relations, as well as assertions with instances. In Fig. 1 and the corresponding piece of fictional RDF code, we can see an application example for RDF. Here, a company named "ACME Inc." is registered in an online directory which also defines a class for companies. Further properties are also listed for the given instance, some of which are online resources (such as the e-mail address of the company or an online entry describing the product supplied by the company), others are just character strings which, in fact, may be pointers to entities outside the World-Wide Web.

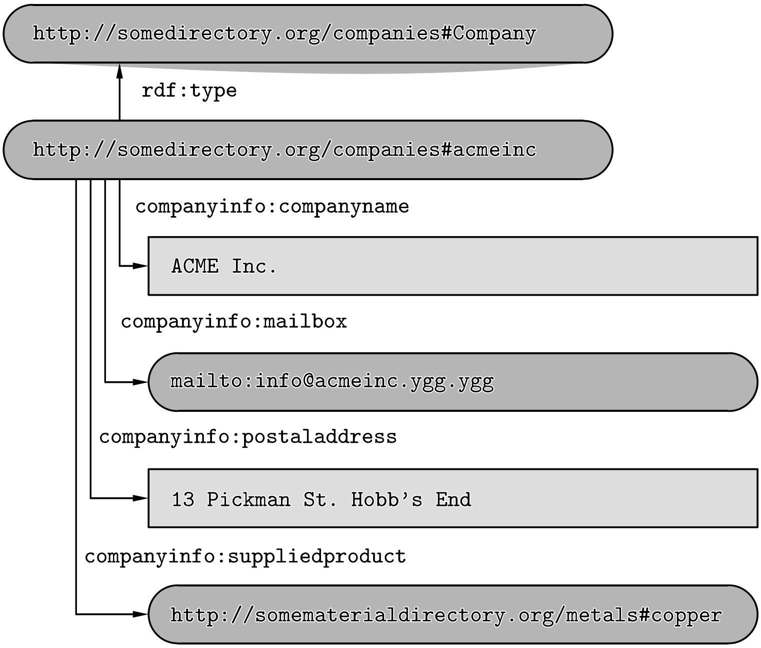


Fig. 1 Graphical representation of an RDF description of an instance and properties

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf=http://www.w3.org/.../rdf-syntax-ns#
  xmlns:companyinfo="http://somedirectory.org/companies#">
  <companyinfo:Company
  rdf:about="http://somedirectory.org/companies#acmeinc">
  <companyinfo:companyname>ACME Inc.</companyinfo:companyname>
  <companyinfo:mailbox rdf:resource="mailto:info@acmeinc.ygg.ygg"/>
  <companyinfo:postaladdress>
      13 Pickman St. Hobb's End
  </companyinfo:postaladdress>
  <companyinfo: suppliedproduct
    rdf:resource="http://somematerialdirectory.org/meta
    ls#copper"/>
  </companyinfo:Company>
</rdf:RDF>
```

As the example shows us, there are several ways of expressing the same content defined in RDF – this makes it possible to implement human-readable interfaces as well as retain machine-readability of the code and thus allow further processing such as sophisticated queries. However, RDF(S) still lacks some features which may be useful for a desired degree of expressivity:

- Decomposition description (disjoint, exhaustive, partitioning etc.) of subclasses;
- Cardinality constraints, default values and value restrictions for attributes;
- Functions;
- Axioms.

OWL. The *Web Ontology Language* OWL was accepted as a recommendation by the W3C in 2004 and has its roots in the fusion of two ontology languages built on top of RDF(S): DAML (*DARPA Agent Markup Language*) and OIL (*Ontology Inference Layer* or *Ontology Interchange Language*). OWL carries on the efforts of RDF(S) and adds further possibilities up to the point where the trade-off between the feasibility of computable and decidable inference (i. e., first-order logic) and power of rich description has to gain attention. OWL is thus subdivided into three layers:

- *OWL Lite* provides basic description means at the lowest possible formal complexity and is thus meant to be used when rudimentary classification hierarchies and simple constraints are to be expressed or efficient processing is required.
- *OWL DL* implements the maximum in description abilities to remain within the limits of first-order logic (DL stands for *Description Logic*). In some cases, this is only guaranteed by adhering to certain rules in the application of language constructs, therefore, care should be taken when resources are transformed into OWL documents.

- *OWL Full* grants maximal expressivity (e. g., instances and classes can be randomly combined in a hierarchy, pre-defined meanings can be further extended etc.), however, at the cost of transcending the limits of first-order logic. Therefore, this layer is meant for cases where a sufficiently rich description in needed rather than automatic reasoning over the semantic information.

In comparison with RDF(S), more elements of set theory can be used for subclass description in OWL (intersection, union, complement, disjoint class etc.), although such cases as exhaustive decomposition and partitioning can only be implemented with cumbersome workarounds. Restrictions can be enforced on properties, such as existential and number constraints, value restrictions, role fillers and inverse roles [5]. Properties can be placed into a hierarchy of subproperties and property equivalencies, symmetry or transitivity can be defined. Also, equivalence or difference between instances can be expressed in OWL.

Aside from the additions to the possibilities of RDF(S), however, OWL still lacks some features which other ontology languages do very well exhibit, such as description of procedures, rules and formal axioms. Whether these will be implemented in further layers – especially in view of the targeted purposes of the Semantic Web – is left to future development.

Support, tools, fields of use. The idea of the Semantic Web can be traced back to the needs of enriching "plain" data with semantic information in a form which can be read and interpreted by machines as well. As a consequence, much emphasis has been put on machine readability and reasoning while developing description languages for the Semantic Web. This, however, does not altogether hamper the possibility of manually navigating through the Semantic Web if a suitable user interface is given.

The XML – RDF(S) – OWL stack of languages has become quite popular with researchers and developers in recent years, and as a consequence, many tools and environments are available, a lot of them on an open-source basis. Aside from syntax highlighting and specific macros provided by numerous text editors for XML, and in some cases for RDF, there are dedicated editors available for constructing sources in XML, RDF(S) and OWL. Also, ontology editors may be able to create OWL ontologies or export existing ones to OWL. One of the most widely used of these editors is *Protégé*, whose development at Stanford University is ongoing. The figures below show screenshots of *Protégé 3.2*, showing parts of an example ontology in various views.

An important step of creating Semantic Web resources is the annotation of already existing web documents with semantic information. A leading example for an editor specifically created for this task is *SMORE* (short for *Semantic Markup, Ontology and RDF Editor*), developed at the University of Maryland's *MINDSWAP* working group (see Fig. 3). This is also a fitting example to represent the distributed construction philosophy of the Semantic Web: documents and semantic information are not necessarily stored separately, i. e., semantic information can reside within the document/resource itself as well, aside from generic ontologies accessible online.

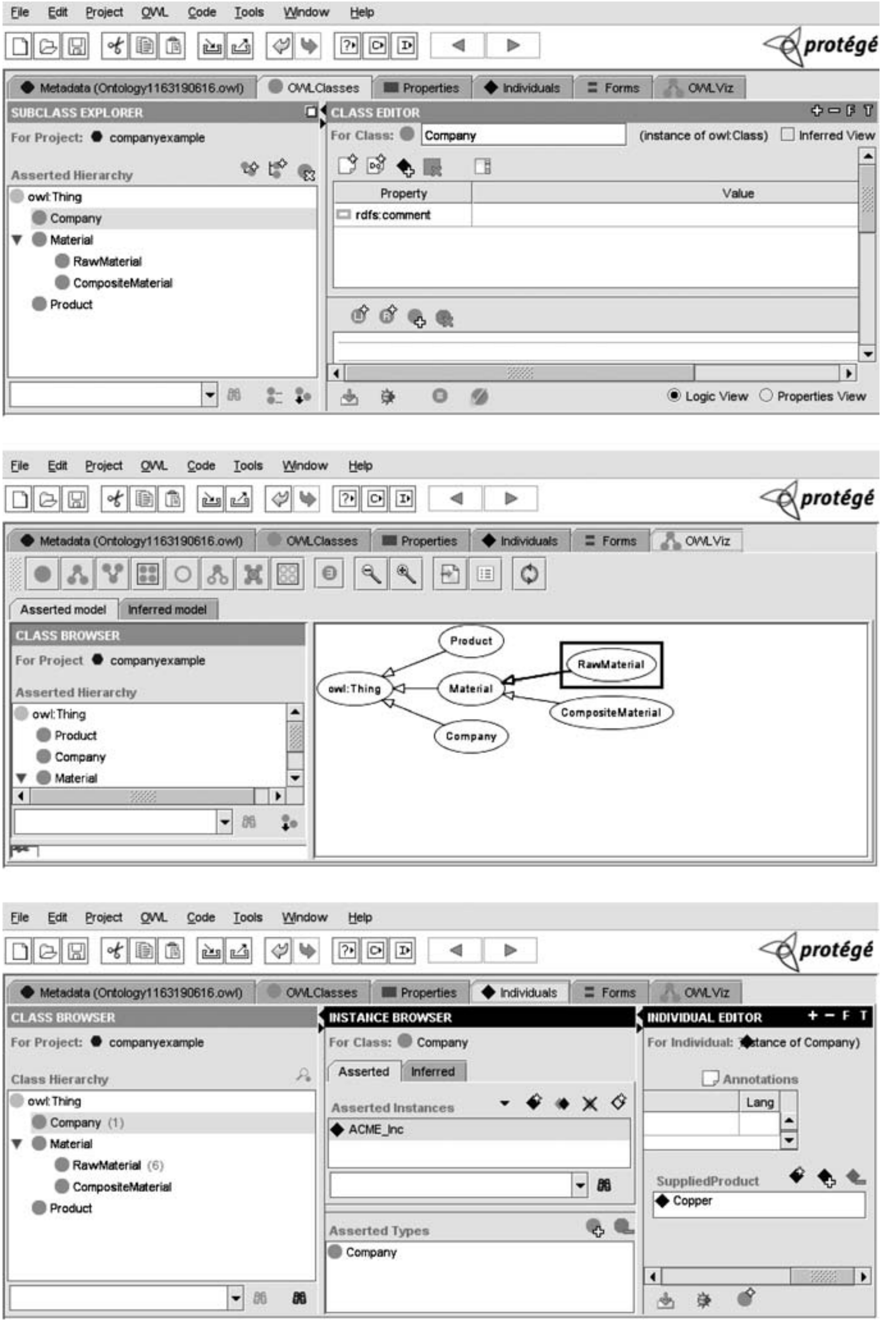
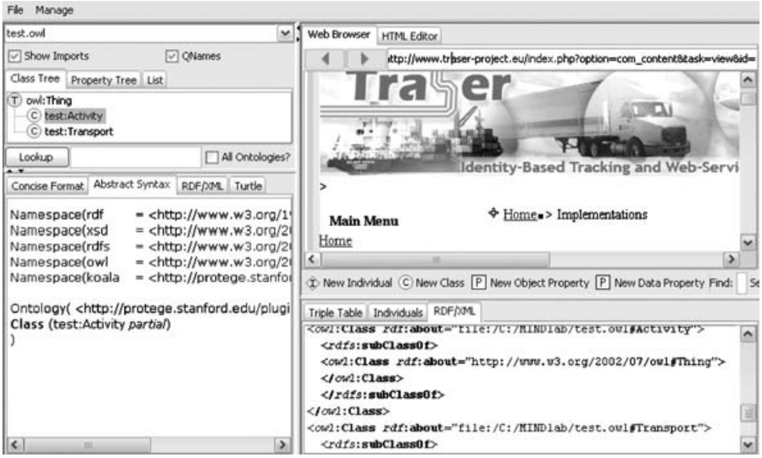


Fig. 2 Screenshots of *Protégé 3.2* – various views of the same ontology



```
Namespace(rdf     = <http://www.w3.org/1
Namespace(xsd     = <http://www.w3.org/2
Namespace(rdfs    = <http://www.w3.org/2
Namespace(owl     = <http://www.w3.org/2
Namespace(koala   = <http://protege.stanfo

Ontology( <http://protege.stanford.edu/plugi
Class (test:Activity partial)
)
```

```
<owl:Class rdf:about="file:/C:/MINDlab/test.owl#Activity">
  <rdfs:subClassOf>
  <owl:Class rdf:about="http://www.w3.org/2002/07/owl#Thing">
  </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="file:/C:/MINDlab/test.owl#Transport">
  <rdfs:subClassOf>
```

Fig. 3 Screenshot of the *SMORE* annotation editor

Last, but not least, reasoning over Semantic Web resources is also possible with a wide variety of inference engines. Just to name a few (and most widespread) examples, the *JENA* and *RACER* engines can already look back on a long history of development and gradual improvement, and reasoning capabilities can also be included into the most commonly used *Protégé* editor, e. g., via the *Jess* engine and the *SWRL* rule language for OWL ontologies.

2.2 Topic Maps

The development of *topic maps* was less layered (as opposed to Semantic Web languages and technologies) and led much earlier to a more or less finalized consensus which culminated in the topic map specification being accepted as an ISO norm in 2000. The topic map paradigm is often mistaken for a "competitor" of the Semantic Web. Contrary to these views, topic maps can rather be considered a "complementary idea" to the Semantic Web, the more so as these two technologies, though being similar in some respect, were created for two separate fields of use and differ in their conceptual structure as well. While the primary goal in creating Semantic Web components was machine readability and processability including reasoning, topic maps are rather thought of as an intelligent support for human browsing, not unlike a flexible index or map which can take on different shapes for different users and can offer services which were technically not possible with printed or "conventional" directories.

Topic maps can be considered comprehensive images mapping to a given field of knowledge – just as a table of contents, an index or a map can be a brief imprint of the contents of a book or a geographical area etc. – but they are not especially designed to reside together with the field they are depicting. Topic maps them-

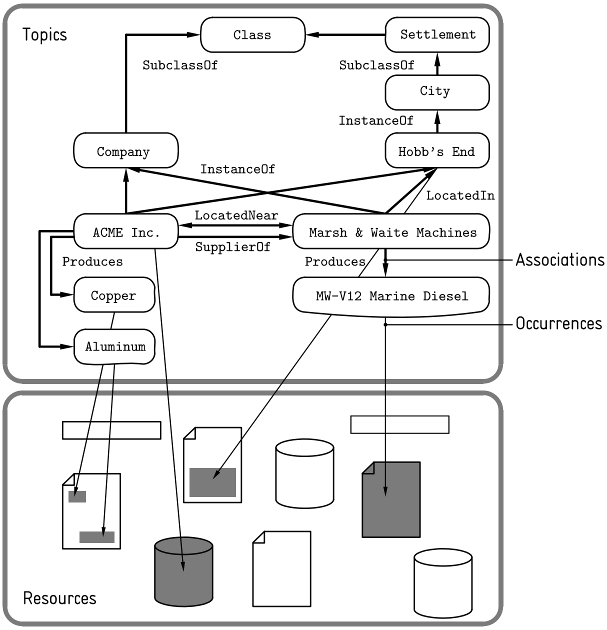


Fig. 4 Overview of a simplified topic map structure

selves revolve around three concepts – often referred to as *the TAO of topic maps* – namely, *topics*, *associations* and *occurrences* (see also Fig. 4).

Topics. One of the usual key components of semantic modelling techniques is the representation of "subjects" or "things" in the domain to be depicted. In topic maps, this is done through topics which represent "things" in general, regardless of where they reside (i. e., resources directly addressable by the computer or physical entities which can only be referred to by a description but cannot be reached directly by the computer). Each topic exhibits the following characteristics:

- an *identity* (usually given by a unique identifier which is used by the topic map management software only, and is thus not always mentioned in works dealing with topic maps);
- one or more *names*, one of which can be appointed a *base name*, while the rest then become *variant names*, analogous to *synonyms* in a natural language;
- *occurrences*, i. e., references that link a topic to resources of the modelled domain which are of relevance to the given topic;
- specific *roles* played in associations with other topics.

Occurrences. As already outlined previously, resources relevant to a topic may be linked to it as *occurrences*. Since a domain represented by a topic map can have elements in machine-addressable form (such as online documents) as well as physical entities existing outside the computer system, topic maps should be able to establish references to both kinds.

- If a subject is accessible within the computer system, it is called an *addressable subject*. A link to such a subject can be established, e. g., using the *XLink/ XPointer URI* notation.
- Entities not reachable by the computer can be referred to in a textual form, such as through a *subject identifier* URI pointing to a *subject indicator*, i. e., the description itself.
- Also, specific values may be directly assigned to a topic in an occurrence-like fashion which is then called a *resource data occurrence*. An example may be direct entries for the density, boiling temperature etc. of a material represented by a topic (even though it is likely that a structured document will describe these parameters in a real application case). In the topic map standard, all resource data occurrences are, in fact, character strings, yet it is possible to specify various ways of their interpretation as an analogy to data types.

Associations. Various kinds of relations can exist between topics which are represented by *associations*. The topic map specification leaves a wide range of possibilities for associations, as it does not restrict the number of topics involved (i. e., not only binary relations are possible) and it does not constrain the direction of the relations. Should this, however, be needed, *roles* can be defined for the topics taking part in the association, so that their place therein is unambiguously specified.

Abstraction: classes and instances. Means for abstraction and group-wise handling of topics, associations and occurrences are provided by instantiation and subclass-superclass relations. Their closer examination outlines the following (see also Fig. 5):

- All three kinds of components – i. e., topics, associations and occurrences – may be instances of various classes. A class itself is also a topic and may be, while functioning as a class, still be an instance of another class, or may be in a subclass-superclass relation with other classes. *This also implies that the topic map standard itself does not guarantee first order logic for classes and instances.* Should this be needed, e. g., for machine reasoning over the topic map, additional constraints must be enforced.
- Topics can be quite freely arranged as instances, classes and superclasses. Aside from the possibility of a topic being instance and class at the same time, it is also possible to place a given topic into several – overlapping – classes at a time. An example for such a case is the so-called *faceted representation* where the same set of items is arranged along several independent abstraction hierarchies.
- The assignment of associations and occurrences to various classes is more restricted, since they can be instances of only one class at a time. Also, in this context, the instance-class relationship is not considered transitive anymore, i. e., the instance of a class is no more held for the instance of its superclass. These constraints are necessary to ensure efficient filtering of the topic map's contents with *scopes*.

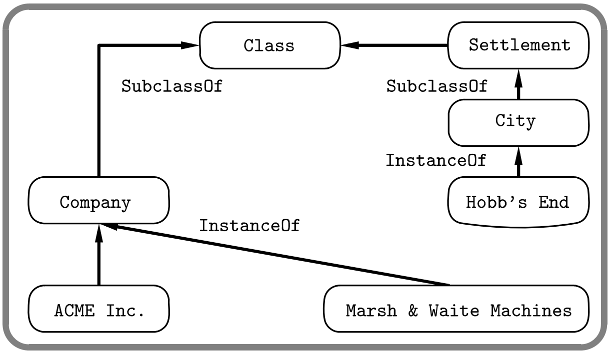


Fig. 5 Instance and subclass hierarchies in the simplified topic map example

Representing viewpoints with scopes. Topic maps provide the possibility of displaying information according to various viewpoints by "filtering" what becomes visible of the characteristics of various topics, as well as associations and occurrences. Referred to as *scopes* (Figs. 6, 7), several views can be exploited for such purposes as:

- Representing the same contents in various languages, with only the names, properties etc. of (one or more) selected languages being shown.
- Filtering associations and occurrences to suit the spectator's field(s) of interest (e. g., if one is only interested in a given range of products of various companies, these could be filtered out if the associations of these products is typified so that scopes can select those of relevance).
- Realizing access control (multi-level if needed), with, e. g., confidential occurrences shown for authorised viewers only.

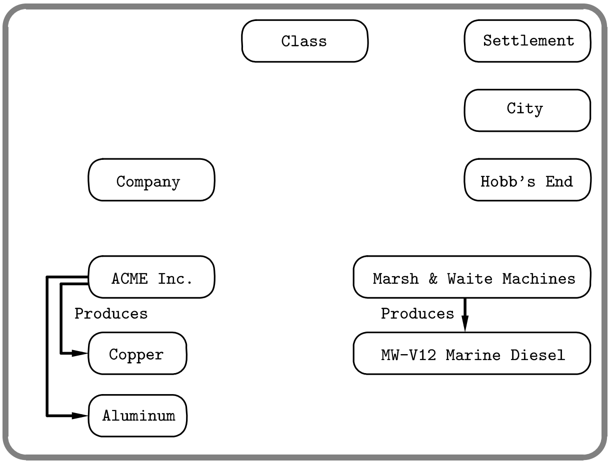


Fig. 6 Filtering associations with scopes

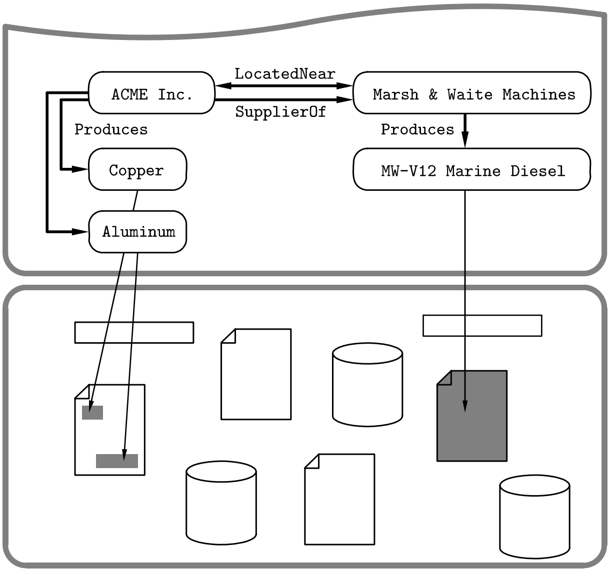


Fig. 7 Filtering occurrences with scopes

Methodology of building topic maps. Topic maps incorporate comprehensive implicite knowledge (especially in the structure of associations), and a well-designed and thoroughly constructed topic map is intended for:

- Longer use and gradual enhancement by large user groups (it is enough to think of community information portals whose "backbone" of knowledge may be provided by a topic map).
- The possibility of *merging* several topic maps while resolving inconsistencies brought about by the merger.
- Providing *portable topic maps* which can index several information pools, having separate sets of occurrences for each.

For this reason, sufficient care should be taken when a topic map is constructed, and to answer this demand, recent research has proposed various methodologies for manual or semi-automatic generation of topic maps. A good example is presented by Kásler et al. [8] (see also Fig. 8) which may also assist the reader in drawing the proper conclusions concerning the nature of Semantic-Web-like resources and topic maps. The process consists of the following phases:

- *Data organization phase.* This is, in fact, the pre-processing of raw data. Depending on the kind of resources (web pages, documents etc.), a variety of methods may be used to extract semantically relevant information from the data. In the example of [8], a large corpus of semi-structured text was processed with chiefly statistical and heuristic methods.

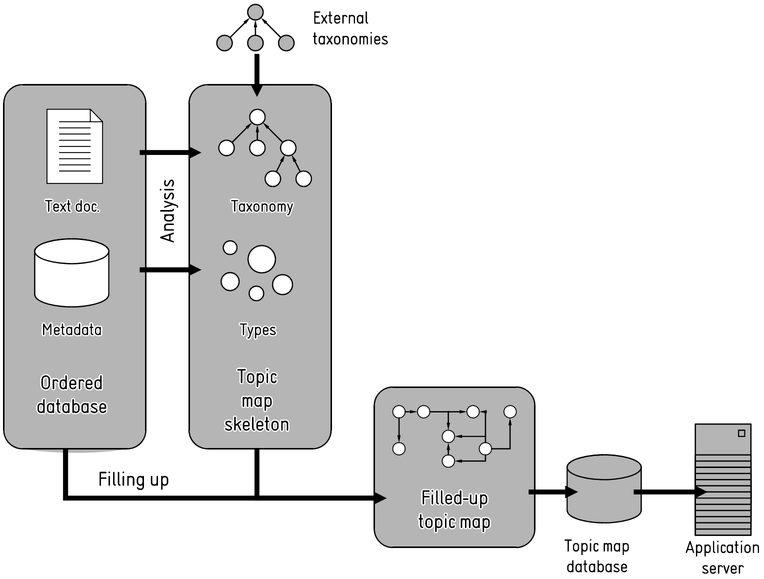


Fig. 8 Semi-automatic assembly of a topic map, as proposed by Kásler et al. [8]

- *Analysis phase.* This step results in a topic map skeleton which contains topics representing larger groups of instances according to one or more simple ontologies taken from external resources. *Note that these external ontologies were not created by this process; they are already existing resources and are used as guides for assembling the topic map skeleton.*
- *Topic map population phase.* Here, the topic map skeleton is populated with actual instance topics gathered in the data organization phase, where after links to occurrences within the resource pool are selected. In the specific case presented in [8], statistics and heuristic decisions are employed to complete the population of the map.

As it can be seen in this example, assembling a topic map is somewhat similar to creating e. g., an index for a book; however, the process requires more work and more intelligent decisions as topic maps may represent much more complicated relations of catalogized resources than a book index usually does.

Support, tools, fields of use. Although the topic map technology is not as commonly known as the Semantic Web, there still is plenty of tools and development environments for handling topic maps. Among the open-source environments, *TM4J* (Topic Map For Java) [23] is a noteworthy example, as well as the rather minimalistic *tinyTIM* topic map engine [22], the latter already implementing the *TMAPI* interface which is regarded by many as a standard topic map API of the future [24]. Also, numerous companies provide commercial topic map solutions, such as Infoloom or Ontopia [18] whose entire product spectrum revolves around topic maps.

2.3 *Comparing Semantic Web and Topic Map Technologies*

As already mentioned, semantic web and topic map technologies are held for competing paradigms by many. This view is, however, far from correct as they are rather complements with respect to their intended field of use and the possibilities they provide. Next, a few points of view will be examined which will reveal the complementary nature of the two approaches.

Location of meta-information. The semantic web technology primarily intends to enrich online resources with local semantic information which either resides within or closely linked to the annotated document, or serves as a central repository of more or less subject-independent semantic background information.

In contrast to this, topic maps are intended to serve as directories or maps which are disjoint from the pool of resources itself, linking its topics to the specific resource instances through occurrence pointers.

Support of machine reasoning. The languages employed to build the Semantic Web pay much attention to preserving first-order logic which ensures the possibility of machine inference. Numerous inference engines and tools are provided to exploit this, and it seems that machine reasoning will indeed contribute much to the future use of the Semantic Web.

Contrary to this, the topic map standard does not envisage to maintain first-order logic (it is enough to think of topics being class and instance at the same time), and most activities in recent research and development are rather concerned about compiling topic maps from resources of various degrees of organisation, as well as maintaining and visualizing them through front-ends.

Use or processing. The Semantic Web is an effort to present distributed online resources in machine-readable form, so that various automated agents or algorithms could process the semantic data and use machine reasoning to obtain further implicit information or draw additional conclusions. These could serve many purposes, but one of the original goals was the development of intelligent search services which are aware of the semantics of the resources rather than gather mere keywords.

Topic maps, on the other hand, are concerned about good readability by humans; all of its features – indexing and filtering of associations and occurrences for easier overview, “human-friendly” naming recommendations etc. – point towards this end. Here, either humans would draw conclusions about the information depicted, or the topic map already implements contents inferred earlier, mainly during map construction.

Concluding remark. As it can be concluded from the above comparison, the Semantic Web is rather a terrain of resources well-prepared for semi-automatic or automatic search, while a topic map is rather the result of a complex search-like (and possibly ongoing) process, most akin to a very versatile index which is already a compressed abstraction of a larger set of resources. Not only does this mean that both paradigms have their own specific field of use, they can also very well complement each other: resources in the Semantic Web can form a pool prepared for extracting semantic information which could then be condensed in a topic map built for the domain of concern.

As for the goals focused on in this paper, topic maps seem more suitable, since the initial phase of design and the search for new partners and solutions is certain to remain a manually steered activity in the next future.

3 Application Example

As a working example for demonstrating the capabilities of a topic map based representation, we selected the *mechanism synthesis* problem. Note that in general, *engineering design* is of particular relevance for us, due to the following reasons:

- Engineers have to look for, access to and work with many kinds of information resources, from technical drawings to textual documents. Normally, both search and browse access to these documents are needed.
- The documents are usually distributed in a number of heterogeneous repositories.
- The design process involves several designers who exchange information throughout a collaborative process.
- Product design covers more and more the complete life-cycle of products; consequently, designers have to take an increasing range of disciplines into account, from manufacturing to production, maintenance and recycling.
- There is a need to attach design rationale to the decisions, as well as appropriate documentation to the result of the design process.
- Last but not least, much of the main competencies of the VRL KCiP community relate also to design.

3.1 Mechanism Design and Classification

Mechanism design is a classical problem of mechanical engineering. The crux of this design problem is *kinematic synthesis*, because it involves the creation of new hardware to meet particular specifications concerning motion: displacement, velocity, and/or acceleration. Hartenberg and Denavit [6] divided the overall problem of kinematic synthesis into three phases:

- *Type synthesis*: Departing from the design specification, determining the structure or type of the mechanism.
- *Number synthesis*: Determining the number of links and the nature of the connections needed to permit the required mobility.
- *Dimensional synthesis*: Calculating the dimensions (lengths and angles) of the links necessary to accomplish the specified motion transformation.

Type synthesis is a hard, ill-structured problem because, on one hand, motion specifications can be combined, while, on the other hand, the designer has to take into consideration numerous factors beyond pure geometry, such as material properties, manufacturing processes etc. Hence, there is no unambiguous scheme for assigning mechanism types to desired motion specifications.

Number synthesis traditionally deals with the mobility of the mechanism which depends on the number of links and the nature of joints. Fortunately, this task can be well algorithmized; there are numerous methods available in the literature which provide specific criteria – e. g., the so-called *Grübler criterion* – concerning the mobility of mechanisms.

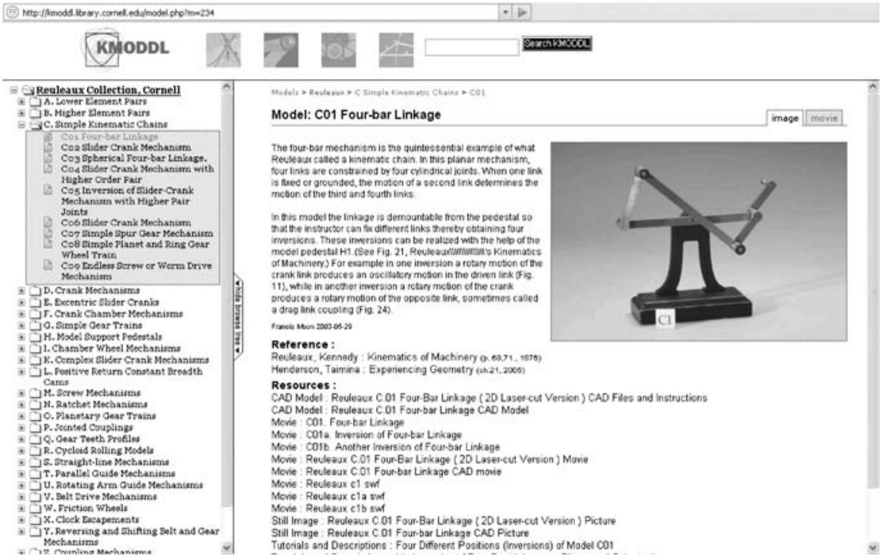
In the phase of dimensional synthesis, the geometric dimensions (mostly length and angle) of links are determined, which is necessary to create a mechanism to effect a desired motion transformation. This subproblem can be systematically solved, just as the number synthesis, and plenty of tools and techniques are available for supporting the solution of dimensional synthesis problems.

As far as type synthesis is concerned, traditional classification schemes of mechanisms, and in particular, the so-called *Reuleaux classification groups*, are used for a systematic consideration of various possibilities. This grouping guides the mind of the designer towards mechanisms best fit for the actual design specifications. Reuleaux's system merges two aspects into a *single classification tree*:

- The *functional* view considers the complete mechanism needed to transform a given motion into another one. Accordingly, groups of mechanisms are formed on the basis of the type of motion they take as input and produce as output.
- The *structural* view deals with the nature of the links and the kinematic pairs. It considers how the motion is transmitted between input and output members or between the kinematic pairs.

A nice example of this merged functional and structural classification can be found in the *KMODDL Kinematic Models for Design Library* [11]. This digital library presents different collections of mechanism models found at different universities (see Fig. 9). The core of *KMODDL* is the Reuleaux Collection of Mechanisms and Machines, a classical collection of 19th century machine elements held by Cornell University's Sibley School of Mechanical and Aerospace Engineering.

Note that the classification scheme is modern in the sense that it is backed by full-scale multimedia presentations (videos, photos, articles, technical documentations). However, it is still based on the traditional Reuleaux's classification groups. Hence, it presents a long list of groups characterized by various functional or structural properties of the mechanisms and puts the individual mechanism into these groups.



Reuleaux Collection, Cornell
- A. Lower Element Pairs
- B. Higher Element Pairs
- C. Simple Kinematic Chains
  - C01 Four-bar Linkage
  - C02 Slider Crank Mechanism
  - C03 Spherical Four-bar Linkage.
  - C04 Slider Crank Mechanism with Higher Order Pair
  - C05 Inversion of Slider-Crank Mechanism with Higher Pair Joints
  - C06 Slider Crank Mechanism
  - C07 Simple Spur Gear Mechanism
  - C08 Simple Planet and Ring Gear Wheel Train
  - C09 Endless Screw or Worm Drive Mechanism
- D. Crank Mechanisms
- E. Excentric Slider Cranks
- F. Crank Chamber Mechanisms
- G. Simple Gear Trains
- H. Model Support Pedestals
- I. Chamber Wheel Mechanisms
- K. Complex Slider Crank Mechanisms
- L. Positive Return Constant Breadth Cams
- M. Screw Mechanisms
- N. Ratchet Mechanisms
- O. Planetary Gear Trains
- P. Jointed Couplings
- Q. Gear Teeth Profiles
- R. Cycloid Rolling Models
- S. Straight-line Mechanisms
- T. Parallel Guide Mechanisms
- U. Rotating Arm Guide Mechanisms
- V. Belt Drive Mechanisms
- W. Friction Wheels
- X. Clock Escapements
- Y. Reversing and Shifting Belt and Gear Mechanisms
- Z. Coupling Mechanisms

Models > Reuleaux > C Simple Kinematic Chains > C01

Model: C01 Four-bar Linkage

The four-bar mechanism is the quintessential example of what Reuleaux called a kinematic chain. In this planar mechanism, four links are constrained by four cylindrical joints. When one link is fixed or grounded, the motion of a second link determines the motion of the third and fourth links.

In this model the linkage is demountable from the pedestal so that the instructor can fix different links thereby obtaining four inversions. These inversions can be realized with the help of the model pedestal H1. (See Fig. 21, ReuleauxâÂÂs Kinematics of Machinery.) For example in one inversion a rotary motion of the crank link produces an oscillatory motion in the driven link (Fig. 11), while in another inversion a rotary motion of the crank produces a rotary motion of the opposite link, sometimes called a drag link coupling (Fig. 24).

Francis Moon 2003-06-29

Reference :
Reuleaux, Kennedy : Kinematics of Machinery (p. 68,71., 1876)
Henderson, Taimina : Experiencing Geometry (ch.21, 2005)

Resources :
CAD Model : Reuleaux C.01 Four-Bar Linkage (2D Laser-cut Version) CAD Files and Instructions
CAD Model : Reuleaux C.01 Four-bar Linkage CAD Model
Movie : C01. Four-bar Linkage
Movie : C01a. Inversion of Four-bar Linkage
Movie : C01b. Another Inversion of Four-bar Linkage
Movie : Reuleaux C.01 Four-Bar Linkage (2D Laser-cut Version) Movie
Movie : Reuleaux C.01 Four-bar Linkage CAD movie
Movie : Reuleaux c1 swf
Movie : Reuleaux c1a swf
Movie : Reuleaux c1b swf
Still Image : Reuleaux C.01 Four-Bar Linkage (2D Laser-cut Version) Picture
Still Image : Reuleaux C.01 Four-bar Linkage CAD Picture
Tutorials and Descriptions : Four Different Positions (Inversions) of Model C01

Fig. 9 KMODDL: Kinematic Models for Design

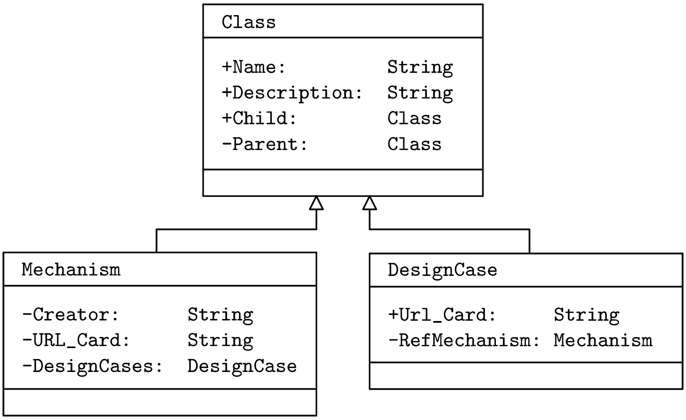
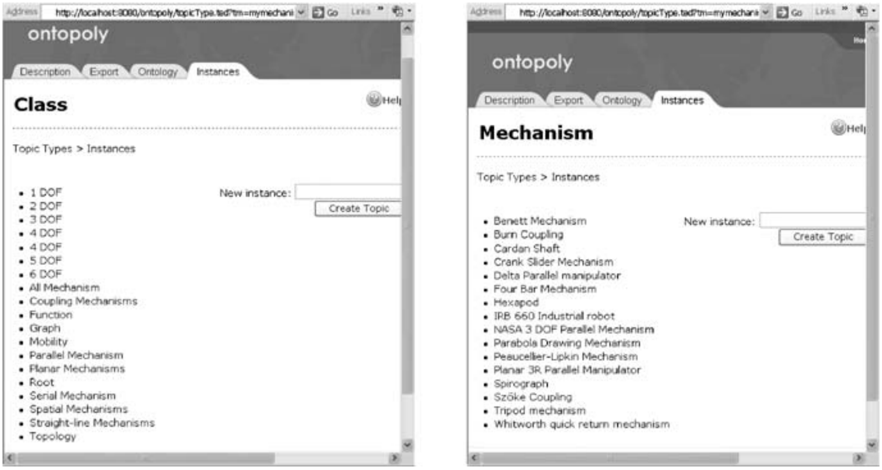


Fig. 10 Topic types of the mechanism topic map



http://localhost:8080/ontopoly/topicType.ted?tm=mymechani

Class

Topic Types > Instances

- 1 DOF
- 2 DOF
- 3 DOF
- 4 DOF
- 4 DOF
- 5 DOF
- 6 DOF
- All Mechanism
- Coupling Mechanisms
- Function
- Graph
- Mobility
- Parallel Mechanism
- Planar Mechanisms
- Root
- Serial Mechanism
- Spatial Mechanisms
- Straight-line Mechanisms
- Topology

http://localhost:8080/ontopoly/topicType.ted?tm=mymechani

Mechanism

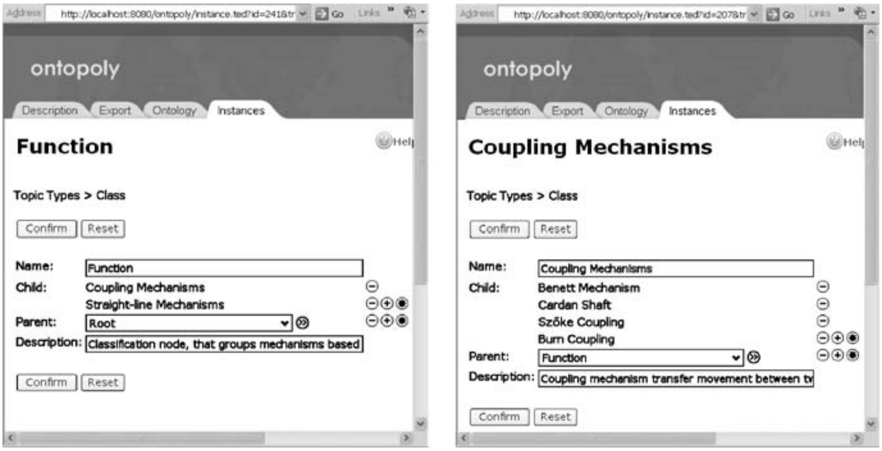
Topic Types > Instances

- Benett Mechanism
- Burn Coupling
- Cardan Shaft
- Crank Slider Mechanism
- Delta Parallel manipulator
- Four Bar Mechanism
- Hexapod
- IRB 660 Industrial robot
- NASA 3 DOF Parallel Mechanism
- Parabola Drawing Mechanism
- Peaucellier-Lipkin Mechanism
- Planar 3R Parallel Manipulator
- Spirograph
- Szőke Coupling
- Tripod mechanism
- Whitworth quick return mechanism

Fig. 11 `Class` and `Mechanism` instances of the topic map

3.2 *Mechanism Classification With a Topic Map*

The traditional mechanism classification scheme has some limitations, especially in supporting browsing. In order to find a special mechanism which might fit the design requirements – i. e., to solve the *type synthesis* problem – one has to browse more or less through the whole tree. This structure does not quite facilitate *associative jumps* of the designer, since there are no – or very limited – cross-references between the classification groups.



http://localhost:8080/ontopoly/instance.ted?id=2418tr

Function

Topic Types > Class

Name: Function
Child: Coupling Mechanisms
Straight-line Mechanisms
Parent: Root
Description: Classification node, that groups mechanisms based

http://localhost:8080/ontopoly/instance.ted?id=2078tr

Coupling Mechanisms

Topic Types > Class

Name: Coupling Mechanisms
Child: Benett Mechanism
Cardan Shaft
Szőke Coupling
Burn Coupling
Parent: Function
Description: Coupling mechanism transfer movement between tw

Fig. 12 Instance of the `Function` and `Coupling Mechanism` topics

Topic maps, however, provide an appropriate representation for these associative bridges. Hence, in the course of an experiment we have implemented a prototype topic map for representing and navigating over the set of classical mechanisms. The topic map has been developed using the Ontopia system [18]. Represented competencies were related to:

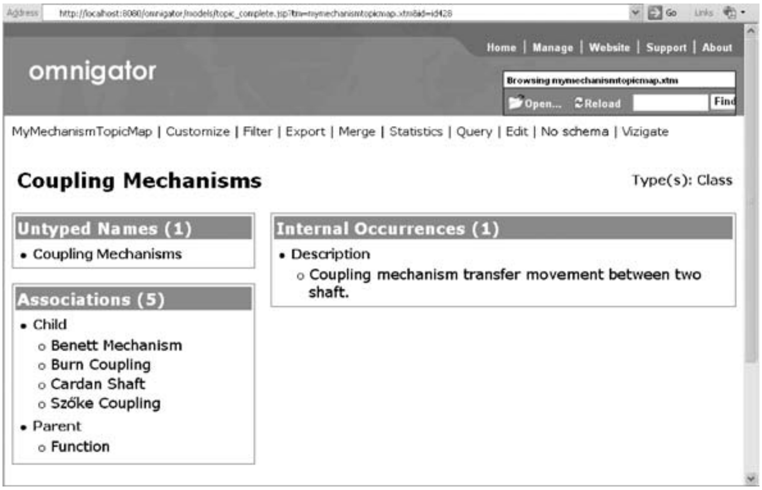
- mechanism structure, and
- design case studies.

Information available and relevant in the above respects has been stored on *bibliography cards*, each card having individual URLs. The topic map has been applied to organize these cards into different classification trees while facilitating the designer's search for right solutions in the set of pre-defined mechanism types.

For constructing the topic map, the object-oriented modelling approach should be taken. First, the *topic types* are established (corresponding to classes) with all their attributes and inheritance relations, then, these topics are instantiated for populating the complete map. The abstract hierarchy of topic types is presented in Fig. 10.

A base topic type, represented by the `Class`, defines all the common properties and relationships of topics. Note that due to the `Parent` and `Child` relations, an instance of `Class` can represent a node in a particular classification tree. More elaborated topic types – such as `Mechanism` and `DesignCase` – inherit these properties and add some more, like `URL_Card`.

The classification trees are built of the instances of the above topic types (see e. g., Fig. 11). Instances of the `Class` represent the classical mechanism classification groups in the topic map, while instances of the `Mechanism` and `DesignCase` represent the actual competence about the mechanisms and design



Browsing mymechanismtopicmap.xtm

Coupling Mechanisms

Type(s): Class

Untyped Names (1)

- Coupling Mechanisms

Associations (5)

- Child
  - Benett Mechanism
  - Burn Coupling
  - Cardan Shaft
  - Szőke Coupling
- Parent
  - Function

Internal Occurrences (1)

- Description
  - Coupling mechanism transfer movement between two shaft.

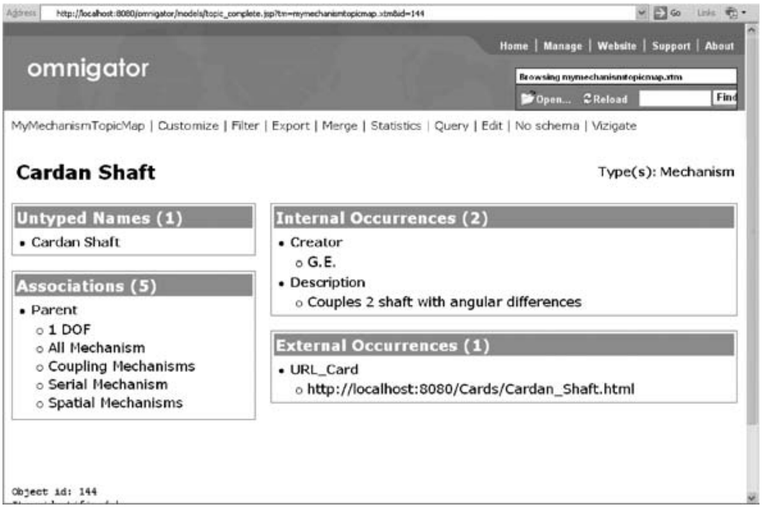
Fig. 13 Browsing the `Coupling mechanisms` cases themselves, respectively. These pieces of information can be reached via the appropriate `URL_Cards`. These URLs are, in fact, the *external occurrences* of the topic map. Note that the `Description` of each instance contains an *internal occurrence*.

Classification trees are built up by specifying the values of the `Parent` and `Child` properties of `Class` instances – these are in fact *associations* linking together topics in a tree-like hierarchy. An advantage of topic maps is that an arbitrary number of such trees can coexist (showing some resemblance to a faceted classification in this concern). Two examples of this hierarchical arrangement are shown in Fig. 12: properties and values of the topics `Coupling Mechanism` and `Function`. As one can see, `Coupling Mechanism` is a subordinate node of `Function` in a classification tree of mechanisms. In this setup, it is quite simple to assign a specific mechanism to a classification node: the topic representing this node should be set as the `Parent` of the mechanism.

A particular mechanism may belong to many different classification nodes at the same time. Crossing between the different classification trees is provided by the associations linking the same mechanism to several classification nodes.

To demonstrate browsing of the topic map, let us assume that we start the search with a functional classification and look for a coupling mechanism, as shown in Fig. 13. In this group there are currently four mechanisms available. We select the `Cardan Shaft` mechanism and browse its information as shown in Fig. 14.

The `Cardan Shaft` mechanism belongs to many other classification groups like the `Spatial Mechanism` group in the topology-based classification tree. By selecting the `Spatial Mechanism` topic, one can arrive at a different classification group shown in Fig. 15.



Cardan Shaft

Type(s): Mechanism

Untyped Names (1)

- Cardan Shaft

Associations (5)

- Parent
  - 1 DOF
  - All Mechanism
  - Coupling Mechanisms
  - Serial Mechanism
  - Spatial Mechanisms

Internal Occurrences (2)

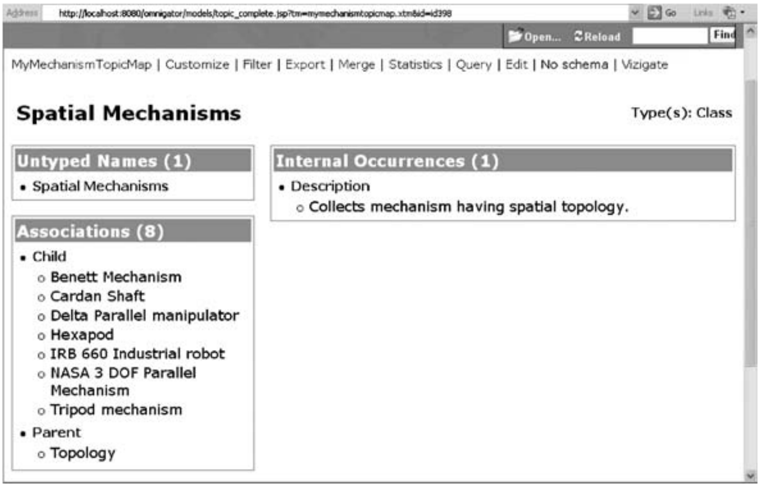
- Creator
  - G.E.
- Description
  - Couples 2 shaft with angular differences

External Occurrences (1)

- URL_Card
  - http://localhost:8080/Cards/Cardan_Shaft.html

Object id: 144

Fig. 14 Browsing the `Cardan Shaft` mechanism



http://localhost:8080/omnigator/models/topic_complete.jsp?tm=mymechanismtopicmap.xtm&id=id398

Spatial Mechanisms

Type(s): Class

Untyped Names (1)

- Spatial Mechanisms

Internal Occurrences (1)

- Description
  - Collects mechanism having spatial topology.

Associations (8)

- Child
  - Benett Mechanism
  - Cardan Shaft
  - Delta Parallel manipulator
  - Hexapod
  - IRB 660 Industrial robot
  - NASA 3 DOF Parallel Mechanism
  - Tripod mechanism
- Parent
  - Topology

Fig. 15 Browsing to the `Spatial Mechanism` classification group

This way, using only the basic associative bridges, provided by the topic map technology, one can easily “come and go” between the different classification groups. Another great advantage of this technology is extensibility. New classification groups can be readily introduced by adding new classification topics to the system and plugging them in by defining their `Parent` and `Child` properties. Of course, this process does not influence the previously defined classification groups, and can therefore be used for representing dynamically changing competencies, too, even for the case of distributed compilation of the topic map, which may become one of the important issues in a growing knowledge community.

4 Conclusions and Future Work

The paper addressed the problem of sharing information and knowledge resources in a community of practice where members strive for a consensus on terms of their expertise, even though, due to the distributed and heterogeneous nature of the community, the understanding of terms may show some local deviation. The most important challenge to be tackled for a working community is the efficient sharing of information and knowledge resources of various community members to others concerned, making the material not only accessible but also usable for the participants.

It was argued that the usual hierarchical tree-like information management systems do not suffice for the above purpose where more flexibility, expressive power as well as human navigation support are required. We proposed to apply the map metaphor and to construct a competence map of the community. Focusing on the representation (rather than the presentation) issue, two possible vehicles – semantic web technologies and topic maps – were examined. We concluded that topic maps are better for the given demands including human browsing, search, navigation as well as extensibility. Finally, a practical example of mechanism classification explained some details of the topic map.

The ultimate goal is to construct and maintain the competence map of our community of practice in a semi-automatic way. We can depart from a given, manually assembled taxonomy of the domain of interest of community members as the first foothold for gathering terms of more or less common understanding. Gathering resources (in our case, individual CVs and research topic descriptions in a well-structured format, with several links to external technical documents) provides raw material for manual or semi-automatic population of a competence map with instances. This construction process is regarded by us as a transition from information to knowledge management in our community of practice.

5 Acknowledgement

This work has been supported by the "Virtual Research Lab for a Knowledge Community in Production" FP6-NMP2-CT-2004-507487-2 grant and the 6th Framework EU project TraSer "Identity-Based Tracking and Web-Services for SMEs" under grant No. IST 033512.

References

1. Berners-Lee T, Hendler J, Lassila, O (2001) The semantic web. Scientific American May 2001:34–43.
2. Camus C, Brancaleon R (2003) Intellectual assets management: from patents to knowledge. World Patent Information 25:155–159.
3. Fattori M, Pedrazzi G, Turra R (2003) Text mining applied to patent mapping: a practical business case. World Patent Information 25:335–342.
4. Gaines BR, Shaw MLG (1995) Concept maps as hypermedia components. International Journal of Human-Computer Studies 43:323–361.
5. Gómez-Pérez A, Fernández-López M, Corcho O, (eds) (2004) Ontological engineering. Springer-Verlag London Ltd.
6. Hartenberg RS, Denavit J (1964) Kinematic synthesis of linkages. McGraw-Hill Book Company.
7. Hicks B, Culley S, Allen R, Mullineux G (2002) A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design. Int. Journal of Information Management 22:263–280.
8. Kásler L, Venczel Zs, Varga LZs (2006) Framework for semi automatically generating topic maps. In: Stein B, Kao O, (eds) Proc. of the 3rd International Workshop on Text-Based Information Retrieval (TIR-06), pp. 24–29.

9. Kostoff RN, Toothman DR, Eberhart HJ, Humenik JA (2001) Text mining using database tomography and bibliometrics: a review. Technological Forecasting & Social Change 68:223–253.
10. Kostoff RN, Boylan R, Simons GR (2004) Disruptive technology roadmaps. Technological Forecasting & Social Change 71:141–159.
11. KMODDL-Kinematic Models for Design Library (http://kmoddl.library.cornell.edu/).
12. Lagus K, Kaski S, Kohonen T (2004) Mining massive document collections by the websom method. Information Sciences 163:135–156.
13. Lenat DB, Guha RV (eds) (1990) Building large knowledge-based systems: Representation and inference in the Cyc project. Addison-Wesley, Boston, MA.
14. McMahon C, Lowe A, Culley S, Corderoy M, Crosslan R, Shah T, Stewart D (2004) Waypoint: An integrated search and retrieval system for engineering documents. Journal of Computing and Information Science in Engineering 4:329–338.
15. Nerkar A, Paruchuri S (2005) Evolution of R&D capabilities: the role of knowledge networks within a firm. Management Science 51:771–785.
16. Novak J, Gowin D (1984) Learning how to learn. Cambridge University Press.
17. Ong TH, Chen H, Sung W, Zhu B (2005) Newsmap: a knowledge map for online news. Decision Support Systems 39:583–597.
18. Ontopia, Official Website. (http://www.ontopia.net/).
19. Singh J (2005) Collaborative networks as determinants of knowledge diffusion patterns. Management Science 51:756–770.
20. Soo V, Lin SY, Yang SY, Lin SN, Cheng SL (2006) A cooperative multi-agent platform for invention based on patent document analysis and ontology. Expert Systems with Applications 31:766–775.
21. Tichkiewitch S, Shpitalni M, Krause FL (2006) Virtual research lab: a new way to do research. CIRP Annals – Manufacturing Technology 55:769–792.
22. tinyTIM – a tiny Topic Map Engine and TMAPI implementation (http://tinytim.sourceforge.net/).
23. TM4J – Topic Maps for Java; documentation (http://tm4j.org/documentation.html).
24. TMAPI – Common Topic Map Application Programming Interface; official website (http://tmapi.org/).
25. Tufte ER (1990) Envisioning information. Graphics Press.
26. Uren V, Shum SB, Bachler M, Li G (2006) Sensemaking tools for understanding research literatures: design, implementation and user evaluation. International Journal of Human-Computer Studies 64:420–445.
27. Yoon B, Park Y (2004) A text-mining-based patent network: Analytical tool for high-technology trend. Journal of High Technology Management Research 15:37–50.
28. Zhua D, Porter AL (2002) Automated extraction and visualization of information for technological intelligence and forecasting. Technological Forecasting and Social Change 69:495–506.