

Facet-based Exploratory Search in Topic Maps

Markus Ueberall and Oswald Drobnik

Telematics Group, Institute of Computer Science,
Goethe-University, D-60054 Frankfurt/Main, Germany,
{ueberall, drobnik}@tm.informatik.uni-frankfurt.de

Abstract. In this contribution, we address exploratory search where a user is faced with an information need concerning a domain he lacks specific knowledge. Based on the work of Delbru et al., which introduced metrics to measure the navigational quality of *automatically selected* facets for RDF data, we apply those findings to the semantically richer TMDM and show how exploratory search functionality can be combined with existing approaches.

1 Motivation

Exploratory search addresses information-seeking problems where a user needs to find out something about a domain where he has a general interest, but lacks specific knowledge. Therefore, he will usually submit tentative queries to a search engine and explore the retrieved information, selectively seeking and passively obtaining clues about his next steps [19]. An exploratory interface allows users to find information without a-priori knowledge of the information space.

Especially if the structure of the data is unknown and/or the dataset in question is large, a visual exploration technique like *faceted navigation* is necessary which does not require the formulation of explicit queries but derives them from the user's selections/navigation decisions. It provides the user with immediate results and also avoids "dead ends" by suggesting restriction values to iteratively narrow down the current view of the information space until a satisfying result is obtained (cf. fig. 1).

The underlying *faceted classification system* enables the assignment of multiple classifications—called *facets*—to an object and the flexible ordering of these classifications in multiple ways rather than following a pre-determined,

taxonomic order. A facet is a metadata attribute that should represent a single important characteristic or property of the classified objects. Intuitive facets describe properties that are either temporal (e.g., *date-of-birth*), spatial (e.g., *located-in*), personal (e.g., *author*), material (e.g., *topic*) or energetic (e.g., *action*). Unfortunately, these facets almost always have to be constructed manually using on-hand ontologies and can only be used efficiently on fixed data structures [3].

In the context of heterogeneous, dynamically changing datasets, however, an automated technique to identify facets—i.e., relationships between objects in the information space—is needed in order to immediately accommodate for changes and provide users with updated, context-sensitive classifications.

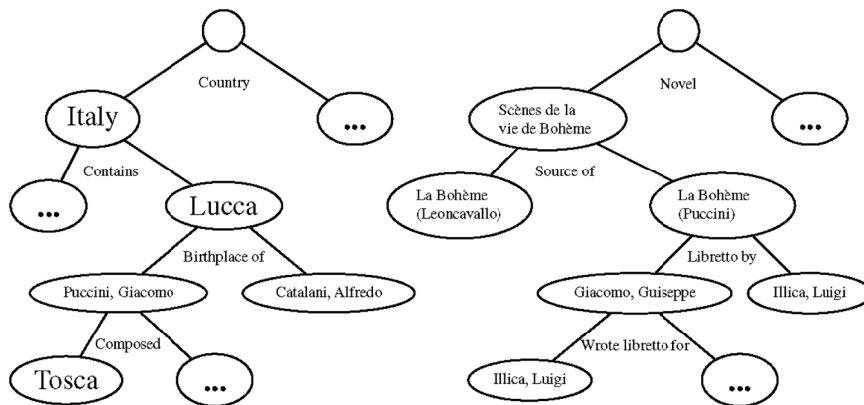


Fig.1. Faceted navigation seen as decision tree traversal: By iteratively choosing a facet and an associated restriction value, the information space is traversed

In the following, we adopt the approach of Delbru et al. [12] which introduces metrics for automatically selected facets for RDF data¹ and show how existing navigation/exploration support found in a number of Topic Map applications can be enriched with exploratory search, shielding the user from the underlying Topic Map based representation of the information space. Section two adapts the original definitions of the metrics with respect to a TMDM representation [10]. The architecture of a TMAPI² based prototype and preliminary experience with its interface is given in section three. Section four discusses the deployment of the forementioned exploratory interface and its perspectives. A summary and an outlook on further work concludes this contribution.

¹ cf. <http://www.browserdf.com>

² Common Topic Map Application Programming Interface, <http://www.tmapi.org/>

2 Facet Selection

A topic map representing the information space can be seen as a graph, the topics being the vertices, and the n-ary associations as well as the occurrences forming the edges. Let $G=(V,E,l_V,l_E)$ be such a graph, where V is the set of vertices, E the set of edges, and l_V and l_E the labeling functions for vertices and edges, respectively.

According to the TMDM [10], all edges are undirected, so there are no designated source and target vertices; however, in order to simplify the following definitions and to emphasise the fact that any navigation implies a direction, G shall—without loss of generality—represent a directed graph where every undirected edge has been replaced by a pair of directed edges pointing in opposite directions. Given an edge, the mappings $source:E \rightarrow V$ and $target:E \rightarrow V$ return the 'source' vertex (i.e., the topic representing the subject from the current point of view) and the 'target' vertex (i.e., another topic referenced by the association in question, representing an object), respectively.

To illustrate these definitions, the well-known 'employment' association example [1,6] is given in LTM notation [7] below. Employer(s) and employee(s) are connected by means of an association, so depending on the chosen *source vertex* for the next exploration step, i.e., either topic `person` or topic `company`, the respective other topic and the reified occurrences connected to the source vertex—`company-website` or `person-job`—are considered the *target vertices*:

```
[employer = "Employer"] [employee = "Employee"]
[employment = "Employment"
 = "Employs" / employer = "Employed by" / employee]
employment([person = "Person"] : employee,
 [company = "Company"] : employer)
[website = "Website"] [location = "Location"] [job = "Job"]
{company, website, "http://company.com/"} ~company-website
{company, website, "http://product.com/"} ~product-website
{company, location, "http://www.frankfurt.de"}
~company-location
{person, location, "http://www.frankfurt.de"}
~person-location
{person, job, [[consultant/programmer]]} ~person-job
```

Contrasting RDF, where a statement is a triple (*subject, predicate, object*) defining the property value (*predicate, object*) for an entity (*subject*) of the information space [3], TMDM offers richer semantics and supports multiple alternative constructs described by quintuples including identity and scope [14, 4]. E.g., instead of representing `job` and `website` by means of occurrences, associations could have been used, too. By effectively treating occurrences as

(binary) associations in the exploratory interface, users are not burdened with representational details.

2.1 Entities, Values, and Facets

Definition 1. An *entity* is a subgraph G' of an information space, extracted by taking all adjacent vertices of a given vertex v , i.e., $G' = (v, V', E', l_V, l_E)$ where $v \in V, V' \subseteq V, E' \subseteq E$ and $\forall e \in E' : source(e) = v \wedge target(e) \in V'$.

Definition 2. A *view* φ is a set of entities of an information space \mathcal{E} .

Delbru et al. [12] use the term *partition* instead of *view*. This might be misleading, since different views need not necessarily be disjoint. Also, the above notion of a *view* is essentially in line with the more extensive view definition we presented in [16].

Definition 3. In a view, a label is associated to one or several edges. A *facet* is a set of labeled edges, i.e., $f_i = \{e \in E \mid l_E(e) = l\}$. F denotes the set of facets of an information space. The *projection facet*: $\varphi \rightarrow F$ returns those facets associated with a view, i.e., $facet(\varphi) = \{f_i \in F \mid \forall e \in \varphi \wedge e \in E, \exists l : l_E(e) = l\}$

Labels can be scoped as demonstrated in the 'employment' example given above in order to reflect the semantics of the respective point of view though incorporating an undirected association, cf. Fig. 2.

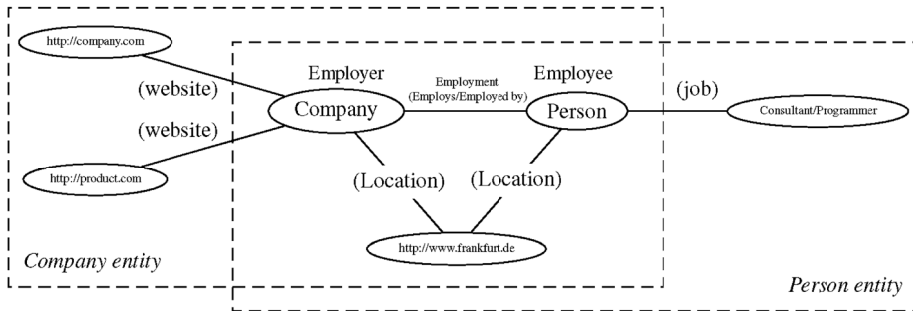


Fig. 2. A view consisting of two overlapping entities, *Company* and *Person*, and associated facets. Unnamed objects are represented by their given topic id in brackets.

Definition 4: The projection $Rv : F \rightarrow V$ returns the set of *restriction values* of a facet, that is, $Rv(f_i) = \{v \in V \mid \exists e \in f_i, target(e) = v\}$. With respect to a set of restriction values of a facet f_i , a view φ can be extracted from the information space. The view implies a new set of facets $F' = facet(\varphi)$, possibly empty.

2.2 Metrics for Navigation

Following Delbru et al. [12], three individual metrics are defined in order to measure the navigational quality of a facet.

Balance: As seen from fig 1, each branching decision optimises the decision power if the tree is well-balanced [20]; the balance of a facet therefore indicates its navigation efficiency. It is computed as the (non-linear) normalised variance of the number $n_s(o_i, f_i)$ of subjects for each object value o_i where $\mu_s(f_i)$ is the vector mean, n_s is the total number of subjects, and $n_o(f_i)$ is the number of different object values for facet f_i :

$$balance(f_i) = 1 - \frac{\frac{1}{n_o(f_i)} \sum_{i=1}^{n_o(f_i)} (n_s(o_i, f_i) - \mu_s(f_i))^2}{1 + \frac{1}{n_o(f_i)} \sum_{i=1}^{n_o(f_i)} (n_s(o_i, f_i) - \mu_s(f_i))^2}, \quad \mu_s(f_i) = \frac{1}{n_o(f_i)} \sum_{i=1}^{n_o(f_i)} n_s(o_i, f_i)$$

Cardinality: A suitable facet has a *limited* amount of object values to choose from. The object cardinality metric is computed as the number of different objects (restriction values) $n_o(f_i)$ for the facet f_i and normalised using a function based on the Gaussian density depending on the parameters μ_o and σ_o :

$$card(f_i) = \begin{cases} 0 & \text{if } n_o(f_i) \leq 1 \\ \exp(-(n_o(f_i) - \mu_o)^2 / 2\sigma_o^2) & \text{otherwise} \end{cases}$$

Frequency: Suitable facets occur frequently inside the collection: the more vertices/distinct concepts (represented by topics, possibly being reifiers) are covered, the more useful the respective facet is in dividing the information space. The frequency is computed as the number of subjects $n_s(f_i) = |\{v \in V | \forall e \in E : l_E(e) = l, source(e) = v\}|$ in the dataset for which the facet has been defined, and is normalised as a fraction of the total number of subjects n_s :

$$freq(f_i) = \frac{n_s(f_i)}{n_s}$$

These metrics can be combined into a final score through (probably weighted) multiplication. As stated in [12], they are solely an indication of usefulness, because they rank facets according to their navigational value, but not according to their descriptive value.

An example of the resulting values for the two entities in fig. 2 is shown in table 1. Evidently, low-ranked facets still have to be displayed in order to cover the entire information space. However, for datasets with a large number of facets, it

is recommended to hide/group them in order to guide the orientation of the users (see below).

facet	balance(f)	card(f)	freq(f)	score
Employment	1.0	0.72615	1.0	0.72615
(website)	1.0	0.72615	0.5	0.36308
(job)	1.0	0.0	0.5	0.0
(location)	1.0	0.0	1.0	0.0

Table 1: Sample metrics for the view consisting of the two entities *Company* and *Person* of fig. 2 ($\mu_o = 10$, $\sigma_o = 10$). The resulting score is the product of all three metrics. (The values will change if additional entities derived from the six vertices are taken into account as well.)

2.3 Additional Facet Classes

As mentioned at the beginning of this section, Topic Map associations differ in compositional granularity from RDF properties. E.g., their expression involves concepts such as role types coupled with role-playing topics [4, 18]. Since a facet browser needs to be able to present the instances of all available types (i.e., topic types, association (role) types, and occurrence types) and the relations between these types need to be made explicit and selectable by the user [9], these additional facet classes have to be considered, too.

The TMDM also defines the concept of scope—albeit it lacks descriptions of the formal semantics [6], which complicates the generic handling as discussed in subsection 3.1. While the introductory employment example merely demonstrated how to use this concept to introduce context-specific labels, its applications (e.g., multilinguality, provenance, opinion, time, audience, filtering) make clear that existing scopes—composed of a set of scoping topics—represent another class of important facets.

All metrics introduced in the previous subsection can be used for these additional classes of facets as well. The only side effect is an overall decrease of the relative navigational value of individual facets as shown in table 1 as their total number increases (cf. role types `employer` and `employee`).

3 Architecture of the Prototype

In order to evaluate the metrics presented in the previous subsection, a TMAPI based prototype has been developed. A text-based open-source version³ is available for interested readers to reproduce the results regarding the exemplary datasets used in this contribution (application of the presented metrics to other Topic Map driven programs is encouraged, the authors appreciate corresponding feedback).

Implementations of query backends are already available, cf. the `tologx` module for the TM4J⁴ engine. Therefore, the processor for the SPARQL query language utilised in the original *browseRDF* prototype could have been replaced by an TM based equivalent. However, we favored a more self-contained solution instead which only uses encapsulated basic TMAPI calls for computing set operations in order to support both the existing and the upcoming revision of that interface. Since all explorative actions are converted into a selection tree, it shouldn't be time-consuming to substitute the forementioned query backends if needed.

The prototype consists of a text-based user interface, a navigation controller (which provides all the functionality required to build a faceted navigation interface), the facet logic (which keeps the current state of the exploration up to date), a facet model (the representation of the facet theory concepts), and an abstraction layer which accesses a TMAPI1 or TMAPI2 compliant topic map engine. The latter also provides hooks for integrating filters and clustering algorithms in order to exclude topic map objects that should not be taken into account and to group both facets and topic map objects that are considered to represent atomic concepts, respectively.

At every iteration, the user may select a facet with or without restriction values (objects) in order to obtain a new view or combine two existing views using `union` and `intersect` operators. It is possible to switch between existing views and display a hierarchical representation of the underlying navigational decisions, which can be modified individually at any time. No (unique) view is ever modified, so it is always possible to backtrack and return to a previous starting point.

³ available for download via <http://www.tm.informatik.uni-frankfurt.de/Plone/Mitarbeiter/ueberall/tm-exploration/>

⁴ Topic Maps 4 Java, <http://tm4j.org>

3.1 Preliminary Experience

In spite of the objective to shield the user from representational details as much as possible and to present a unified navigation interface, it is necessary to provide him with associated information if he wants to. As an example, the different facet classes mentioned in the previous section potentially exhibit varying qualities regarding their “navigational value”, especially when the underlying topics are re-used in different contexts—which in fact should be avoided if possible, cf. [1]. Without using a windowing toolkit, however, the text-based user interface becomes overloaded quickly, therefore it is planned to provide a browser-based interface for the prototype in the run-up to this conference as well.

With respect to the `union` and `intersect` operators, which were introduced in order to conveniently combine results of different exploratory navigation paths, the missing formal semantics of scope are bound to come to the fore as soon as the user selects associated facets: In this case, the new combined view is likely to contain conflicting statements in different scopes or objects that are looked at from multiple perspectives at the same time, which illustrates a prominent example of the “and/or problem” [6]. The only known workaround for this is to provide the user with different selectable “interpretations”, which unfortunately bypasses the objective to shield the user from representational details at least to a certain extent.

Like the forementioned operators, the possibility to modify “sub-selections” (i.e., previous exploratory decisions on a longer navigation path) strictly speaking isn’t supposed to be considered a part of an *exploratory* interface, because these modifications may require additional decisions or lead to empty (intermediate) result sets. However, these options form an important basis of a computer-supported, both representation- and (query) backend-independent formulation of non-trivial queries which can be analysed by the user afterwards. The main benefit in conjunction with an exploratory interface is that the user is immediately able to relate (intermediate) result sets to atomic navigational decisions.

4 Topic Map Exploration

To get a first insight into navigational support, typical generic Topic Map user interfaces as implemented in the Ontopia Omnigator and Vizigator are looked at. Then we demonstrate how a facet-based exploratory interface extension can enrich navigational support. The Italian Opera topic map (revision 1.8) by Steve

Pepper⁵ is well suited as exemplary dataset. Finally, several perspectives are discussed to offer additional improvements.

4.1 Omnigator and Vizigator

The combined interfaces of the Ontopia Omnigator and Vizigator can be regarded typical and rather comprehensive representatives of applications that are able to display the contents of generic topic maps⁶.

While the Omnigator is a general purpose topic map browser which is not recommended for end users but can rather be considered a teaching aid, the Vizigator is designed for graphically browsing and navigating topic maps.

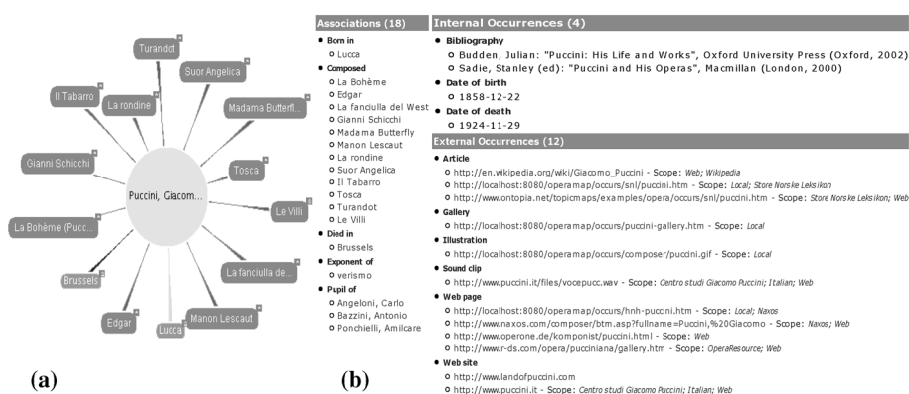


Fig. 3. The vertex representing Giacomo Puccini as shown by the Vizigator (a) and details about associations and occurrences as rendered by the Omnigator (b).

The initial Omnigator view lists all types of a topic map. The text-based browser interface supports navigation between all objects contained in the current topic map, grouping them based on their types according to the TMDM (cf. the lists of associations and occurrences in fig. 3(a)). Known taxonomic information, e.g., existing supertype-subtype relationships, is also displayed. Fulltext searches on names and contents/locators of internal and external occurrences are available.

The Vizigator view of a topic as shown in fig. 3(b) is comparable to the diagram of an entity as outlined in fig. 2. However, occurrences are not handled like

⁵ cf. http://www.ontopedia.net/omnigator/models/topicmap_nontopoly.jsp?tm=ItalianOpera.ltm

⁶ cf. <http://www.ontopia.net/download/freedownload.html>

binary associations, but are listed in a context menu resembling the Omnigator display. Here, only topic names can be searched.

While both interfaces support basic searches, more complex queries which include properties of/relations between topic map objects require the additional use of the tolog query language and thus a certain amount of knowledge regarding the underlying TM representation.

4.2 Faceted Navigation

Using the metrics defined in subsection 2.2, it is possible to group both associations and occurrences into a set of facets. The visualisation of the graph can be restricted to objects that are best suited to support the next exploration step. In this way, the user's orientation in the information space is facilitated. With high probability, dead ends of the search process are avoided. As shown in fig. 4, a view with a single vertex representing composer Giacomo Puccini would only contain five facets representing different "properties". For each set of properties, a limited number of available objects, i.e., possible restriction values is shown. For additional information, the user could either have a look at the individual topic map objects by reverting to a Omnigator/Vizigator like display or extend the current view by selecting additional objects/subjects.

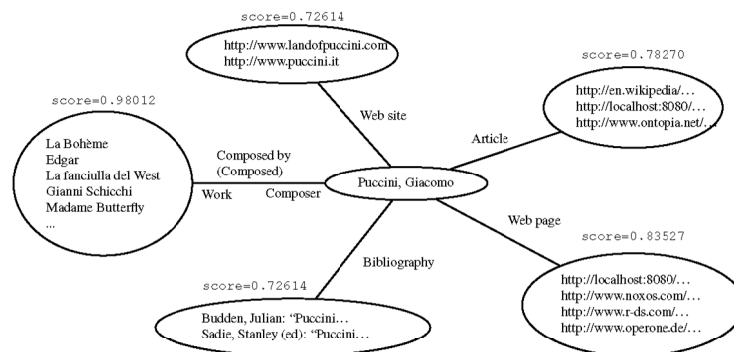


Fig. 4. Stylized display of the vertex representing Giacomo Puccini including only those facets—representing either associations or occurrences—of high navigation quality.

By combining multiple exploration steps or actions which either consist of basic selections of a restriction value, existential selections (i.e., arbitrary values must exist), union, and intersection operations, it is possible to generate a query which is far more powerful than the simple text-based searches. The query can include structural information about the topic map objects while still shielding the user

from the use of a query language and representational details (e.g., associations and occurrences have to be treated differently in tolog). Fig. 5 shows how to determine the name of a certain opera (Tosca) by providing a number of restrictions/constraints.

4.3 Discussion of Perspectives

For large information spaces, the number of facet values may explode. In this case, the navigation process can be improved by reducing the initial number of different facets associated with a view. Two approaches are possible and may be combined for additional user guidance:

1. entities can be partitioned in order to group facets: this requires that, e.g., existing structural knowledge about the domain can be exploited.
2. facet values can be clustered: the clustering has to be computed on the fly and also accommodate for different data types [8].

A good candidate for the first approach is the supertype–subtype relationship as well as a (limited) number of concepts found in the literature that have well-known PSIs [1], which underlines the importance of unified and published processes and concepts for building topic maps [11,17].

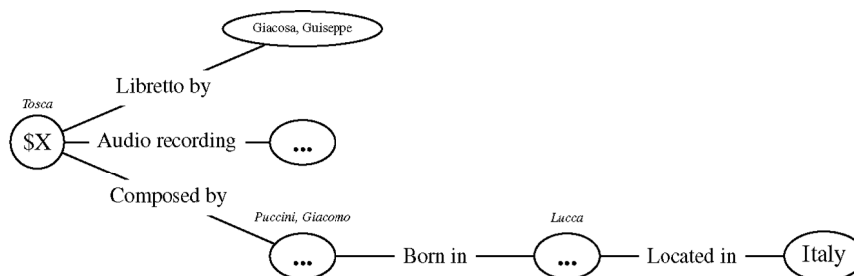


Fig. 5. Example of a resulting complex query generated by combining exploratory actions using basic selection (of a restriction value), existential selection, and join selection with the intersection operator

Aside from the general approach for arbitrary topic maps, an application built around a known ontology like, e.g., the OperaMap Application⁷ is able to filter topic map objects with respect to their usefulness to the exploratory search. Candidates for filtering are metadata or templates which should be presented to the user in a different way. This is especially important for datasets containing versioned concepts [17] which account for a dedicated navigation/user interface

⁷ cf. <http://www.ontopia.net/operamap/>

in order to prevent users from *unintendedly* utilising automatically selected facets that represent explicitly revoked connections.

Additional metrics like the concept of semantic distance between vertices as described in [2] could also be incorporated for determining the (potential) correlation of two automatically selected facets. However, as with filtering, these metrics are likely to be specific to the dataset or application domain. In any case, the three metrics of subsection 2.2 for automatic selection of facets can be used as a fallback.

In order to provide a user with a restricted set of initial topic map objects as a starting point for navigation based on, e.g., a list of interested topics which are not necessarily connected, the creation of a minimal sub-graph—basically a (set of) minimal spanning tree(s) containing the formentioned list of objects—as described in [5] could be considered. Using our current prototype, several manual merge and filter operations are necessary to obtain a comparable view for an initial set of objects which don't share (known) attributes. While their proposed algorithm only operates on existing associations of arbitrary type between topics, the combination of both exploratory interfaces should enable users to faster isolate topic map fragments of interest.

Finally, if the faceted navigation interface were to support the definition and reference of variables acting as sets of restriction values, more complex queries could be generated, e.g., returning a list of all persons within the Italian Opera topic map which were born and died in the same place.

5 Summary and Outlook

The presented exploratory search functionality based on automatically selected facets gives users the chance to inform themselves about an information space without specific domain knowledge. Such kind of interface may enhance existing navigation aids. It is possible to construct queries as depicted in fig. 5 just by following links between concepts of interest, regardless of the underlying query language (e.g., tolog or TMQL) and legend (e.g., the TMDM) which defines how typing information—among other things—is actually represented. As such, this interface can serve as a basis for a user's individual way of looking at concepts and relationships in order to increase the user's productivity as discussed in [13]. Just like the resulting queries, selected navigation paths could be stored in order to provide guidance to users with similar search interests.

Currently, the forementioned functionality is being integrated in the user interface of an Eclipse based prototype for software engineering support, a setting which involves heterogenous user groups with different domain

knowledge [15]. In order to match concepts from different domains, e.g., use cases and design patterns, members of different teams are repeatedly required to trace links between them, thereby crossing the boundary of their own domain of expertise. Exploratory search can complement pre-defined/customisable queries and domain-specific graphical presentation of concepts during the composition and instantiation of templates to come up with a successive formal structuring of information as discussed in [16, 17].

References

1. Ahmed, K.: Topic Map Design Patterns Tutorial. Second International Topic Maps Users Conference (2008).
2. Andres, F., Naito, M.: Application Framework Based on Topic Maps. Proc. 1st International Workshop on Topic Maps Research and Applications (TMRA). Springer Lecture Notes in Artificial Intelligence (LNAI) 3873 (2005) 42–52, http://dx.doi.org/10.1007/11676904_4
3. Delbru, R.: Manipulation and Exploration of Semantic Web Knowledge. Internship Report DERI and EPITA France, July 2006. <http://rdelbru.free.fr/doc/Report.pdf>
4. Dichev, C., Dicheva, D., Dicheva, B., Moran, M.: Translation between RDF and Topic Maps: Divide and Translate. Proc. Balisage: The Markup Conference 2008. <http://www.balisage.net/Proceedings/html/2008/Dichev01/Balisage2008-Dichev01.html>
5. Dichev, C., Dicheva, D., Fischer, J.: Identity: How To Name It, How To Find It. Proc. 16th International. Conference on World Wide Web (WWW) 2007. http://www2007.org/workshops/paper_133.pdf
6. Garshol, L. M.: A Theory of Scope. Proc. 3rd International Workshop on Topic Maps Research and Applications (TMRA). Springer Lecture Notes in Artificial Intelligence (LNAI) 4999 (2008) 74–85, http://dx.doi.org/10.1007/978-3-540-70874-2_9
7. Garshol, L. M.: The Linear Topic Map Notation: Definition and introduction, version 1.3 (rev. 1.23, 2006/06/17), <http://www.ontopia.net/download/ltn.html>
8. Hearst, M.: Clustering versus faceted categories for information exploration. Communications of the ACM Vol 49 No 4 pages 59–61, April 2006. <http://doi.acm.org/10.1145/1121949.1121983>
9. Hildebrand, M., van Ossenbruggen, J., Hardman, L.: /facet: A Browser for Heterogeneous Semantic Web Repositories. Proc. 5th International Semantic Web Conference (ISWC). Springer Lecture Notes in Computer Science (LNCS) 4273 (2006) 272–285, http://dx.doi.org/10.1007/11926078_20

10. ISO/IEC WD 13250-2: Topic Maps - Data Model (TMDM), 2008-06-03, International Organization for Standardization, Geneva, Switzerland.
<http://www.isotopicmaps.org/sam/sam-model/2008-06-03/>
11. Maicher, L.: Autonome Topic Maps, Dissertation, Leipzig 2007 (in german).
http://www.informatik.uni-leipzig.de/~maicher/publications/DISS_LutzMaicher_german.pdf
12. Oren, E., Delbru, R., Decker S.: Extending Faceted Navigation for RDF data. Proc. 5th International Semantic Web Conference (ISWC). Springer Lecture Notes in Computer Science (LNCS) 4273 (2006) 559–572,
http://dx.doi.org/10.1007/11926078_40
13. Park, J., Cheyer A.: Just for Me: Topic Maps and Ontologies. Proc. 1st International Workshop on Topic Maps Research and Applications (TMRA). Springer Lecture Notes in Artificial Intelligence (LNAI) 3873 (2005) 145–159,
http://dx.doi.org/10.1007/11676904_13
14. Schmitz-Esser, W., Sigel, A.: Introducing Terminology-based Ontologies. Proc. 9th International Conference of the International Society for Knowledge Organization (ISKO), 2006. <http://eprints.rclis.org/archive/00006612/>
15. Ueberall, M., Drobnik, O.: Collaborative Software Development and Topic Maps. Proc. 1st International Workshop on Topic Maps Research and Applications (TMRA). Springer Lecture Notes in Artificial Intelligence (LNAI) 3873 (2005) 169–176,
http://dx.doi.org/10.1007/11676904_15
16. Ueberall, M., Drobnik, O.: On Topic Map Templates and Traceability. Proc. 2nd International Workshop on Topic Maps Research and Applications (TMRA). Springer Lecture Notes in Artificial Intelligence (LNAI) 4438 (2006) 8–19,
http://dx.doi.org/10.1007/978-3-540-71945-8_2
17. Ueberall, M., Drobnik, O.: Versioning of Topic Map Templates and Scalability. Proc. 3rd International Workshop on Topic Maps Research and Applications (TMRA). Springer Lecture Notes in Artificial Intelligence (LNAI) 4999 (2008) 128–139,
http://dx.doi.org/10.1007/978-3-540-70874-2_13
18. W3C Working Group Note: A Survey of RDF/Topic Maps Interoperability Proposals (2006). <http://www.w3.org/TR/2006/NOTE-rdftm-survey-20060210>
19. White, R. W., Marchionini, G., Muresan, G.: Evaluating Exploratory Search Systems. Information Processing & Management, Volume 44 Issue 2 (2008) 433–436,
<http://dx.doi.org/10.1016/j.ipm.2007.09.011>
20. Xu, R., Wunsch II, D.: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks, Vol. 16 No. 3 (2005) 645–678,
<http://doi.ieeecomputersociety.org/10.1109/TNN.2005.845141>