Contextual Retrieval of Digital Content in Topic Maps

Christo Dichev, Darina Dicheva, Jan Fischer

Winston-Salem State University
601 M.L.K. Jr. Dr., Winston Salem, N.C. 27110, USA
{dichevc, dichevad, jfisc139}@wssu.edu
http://gorams.wssu.edu/faculty/dichevc/
http://myweb.wssu.edu/dichevad/

Abstract

In this paper we discuss the contextual support provided in TM4L - an environment for building and using Topic Maps-based learning repositories. We propose an extension of the Topic Maps model with contexts and an approach for contextual retrieval of information. Our approach is based on defining context primitives and a simple language for expressing *context queries*. The resources retrieved by running the queries can be directly examined or used as *seeds* for further locating relevant resources in the repository. The contextual retrieval functionality is provided through an intuitive user interface. We discuss the context representation and the contextual retrieval implementation in TM4L.

1 Introduction

The exploration of information resources is always affected by users' context. Consequently, in case of concept-driven access to digital collections, the conceptual structure should not be predetermined on the basis of many assumptions, but should rather allow views from different perspectives, depending on the context in which the digital collection is being used. There are two major problems related to information retrieval (IR) in such collections. The first problem is related to user's assumption of a certain structure when exploring the collection. This problem stems out of previous user's experience with digital classifications, textbooks, etc. that impose a particular view on the information. Indeed, since a classification is intrinsically contextual, digital collections tend to inherit certain individual perspectives from their authors. The second problem is related to searching information relevant to a particular subject. Although various search engines and classification systems are available, finding relevant information is still a challenging task. Whether the retrieved documents are relevant to the user's request depends on much more than just the subject of the documents. A big part of the problem is that relevancy is contextual and subjective to the individual who is performing the search. What is relevant depends on the context. Only the users can judge how relevant a particular bit of information is to what they are attempting to discover. The documents may be too technical or too general or out of date for one's needs.

Obviously the information retrieval would be much more efficient if the user's context is taken into account. In this paper we propose an approach to contextual retrieval of information in Topic Map-based digital collections. We extend the Topic Map model (Biezunski, Bryan, & Newcomb 2002) with a notion of context beyond scopes and attempt to solve the above mentioned IR problems by: (i) making topic map authors' vocabulary available to users for expressing their information needs (ii) using samples identified by the users to guide them to the location of relevant resources within the collection.

We have exemplified the proposed approach in TM4L – a Topic Map-based environment for building, maintaining, and using standards-based, ontology-aware e-learning repositories (Dicheva & Dichev 2006) (see http://compsci.wssu.edu/iis/nsdl/).

The paper is organized as follows. In Section 2 we discuss briefly the Topic Maps model and TM4L. In Section 3 we define contexts incorporated in topic maps and propose an approach for context mapping. In Section 4 we discuss the extension of TM4L to support contextual retrieval in learning repositories.

2 Building Topic Map Repositories

Topic Maps (TM) are a technology (Park & Hunting 2002) that provides a flexible and intuitive modeling paradigm for defining a *conceptual navigation layer* that supports finding of web resources of various kinds, such as documents, images, database records, audio/video clips, etc. The advantage of using the TM technology for developing digital learning collections is twofold: from one side it provides convenient and intuitive presentation of interrelated concepts embedded in information resources, and from another, the material is in a standards-based format, which makes it interchangeable and interoperable.

Basically, topic maps are collections of topics, associations, resources, and scopes. In the Topic Maps model the concepts are reified in topics, and they can be categorized using types. TMs describe by means of topics what a resource is about. Associations express semantic relationships between topics, and the extent of validity of topics, associations, and resources is called scope. The concept of scope is used in topic maps basically as a constraint on the interpretation of a topic characteristic. A typical application of scopes is to resolve name collisions. For example, a topic map may contain two topics named *Washington*, one in the scope of *City* and the other in the scope of *State*. A user may specify a scope while searching a topic map in order to reduce the number of returned hits. However, a scope does not provide a mechanism for defining regions of related topics within a given topic map. The basic concepts of TMs such as topics, associations, etc., are simple but expressive enough for representing complex structures. TMs can be viewed as a method for structuring and organizing information on the semantic and metadata level.

TM4L is an environment for building and using ontology-aware learning repositories represented as topic maps. The TM4L Editor allows creating hierarchies of topics, topic types and instances as well as relations between topics and learning resources. It supports TM-related operations such as merging, browsing, searching, and scoping. The learning content created by the Editor is compliant with the ISO XML Topic Maps (XTM) standard (Biezunski, Bryan, & Newcomb 2002), thus interoperable with any XTM-based applications.

In order to account for personalization in the topic map navigation, we propose to mediate the access to the learning collection by using *contexts*. Contexts can be considered as specific views on a domain, dependent on the users' information needs. Our approach is based on the assumption that a successful information support application should not force users to change their way of looking at the learning repository, as an externally imposed schema might be perceived either as unfamiliar or unusual. The adopted Topic Map platform provides the *scope* mechanism that allows representation of context sensitive information. Exploiting the scope features, we extend the Topic Maps paradigm in order to incorporate the notion of context.

3 Using Context in Topics Maps

In an ideal world, information seeking applications would respond to their users presenting always the precise set of resources they might need to complete their current task, automatically putting away any irrelevant documents. This type of interaction implies that the application is aware of the user's personal *context*. There are many ways to model or represent context, and *scope* is just one means explicitly provided by the TM model. The *scope* mechanism enables any topic characteristic to be qualified by defining a range within which the information is valid. Scopes may be used to define different perspectives on the same set of information. For example, scopes may be used to separate "beginner" resources from "advanced" resources, thus enabling different sets of information to be presented to users with different knowledge levels. The scope concept is intended as a syntactic shorthand in cases where more elaborate modeling of context by other means is beyond the intended 'expressiveness' of the topic maps.

Intuitively, we define a context as a collection of entities grouped on the basis of their relations to a common set of features determined by the current user's goal. Entities can be any objects, facts, statements, resources, etc. The common set of features defines the *grouping criterion* for entities. The goal specifies the boundaries of the context, which consists of all entities directly or indirectly related to the defining set of features and relevant for achieving the goal. The set of assumptions that provides a background of a particular task such as finding a book in a library is an example of such a context. Here, the context includes the assumptions about the library organization, classification system, location of specific shelves, etc.

Often certain assumptions are left implicit. That is why the notion of context is so elusive, despite of the various models proposed and developed (Bouquet et al. 2005; Bouquet et al. 2004; Dey 2000; Giunchiglia 1993; Guha, R. 1991; McCarthy 1987). In this work we don't intend to propose a general framework for modeling contexts, we rather take a pragmatic approach.

In the Topic Maps world all objects of interest are mapped to topics, associations, occurrences and scopes. Our intuitive interpretation of context in Topic Map terms is as the minimal environment that surrounds and gives meaning of the topics of interest. In this sense, context can be interpreted as the minimal sum of all related topics and related occurrences. Thus, *we define context as a set of facts (topics or occurrences), grouped on the basis of their relations with a common set of topics, associations, or scopes, determined by a given goal.* The set of all topics related *to a given topic* is an example of such a context, where the task is to find all relevant topics. In this example all objects that belong to the context share a common topic. Another example is the set of all pairs of topics generated by the *whole-part* associations starting from a particular topic. This context can be used when exploring digital content from the viewpoint of its natural parts, such as chapters, subchapters, etc. Learning materials for advanced students in a course support system is yet another example with context - the set of topics and occurrences sharing a common scope ("Advanced students").

3.1 Contexts in TM4L

The typical use of TM4L is for supporting users in their task of finding relevant resources. Therefore, in TM4L we use the above context definition with the restriction of the set of entities to a set of *occurrences.* The common set of features defining the context of a TM4L user's information needs is a combination of topics, relation (association) types, resource types, and themes (scopes).

We describe contexts using a simple language for creating *contextual expressions*. A contextual expression is interpreted as a query for information resources that represents a user's current information needs. The evaluation of the query results in a set of resources satisfying the context expression. Each context expression is defined as follows:

```
<cont_expression> -> <topics> $ <relations>
     & <themes> & <resource_types> & <range>
<topics> -> [topic_name]
<relations> -> {relation_type_name}
<themes> -> {theme_name}
<resource_types> -> {resource_type_value}
<range> -> full | terminals
```

For simplicity we restrict our consideration to binary relations. The intended use of relations within the contextual expressions is for generating *descendant topics*. Generating "descendants" assumes some kind of directionality. For example, when using the *whole-part* relation we typically generate the *parts* from the *whole*. In a similar fashion, for all predefined relations, TM4L applies a default directionality, which defines the order of the topic traversal. Analogously, for relation types defined by the authors this directionality is determined by the order of the roles in the definition of the corresponding relation type. For example, *simpler(less-simple, more-simple*) specifies that, starting from the topic playing a role of *less-simple*, we generate its children playing the role of *more-simple* topics.

The term *range* refers to the level of resource inclusion/filtering given the set of descendant topics, generated from the set of topics, specified in the context expression. The range value of "full" denotes the set of resources linked to the full set of descendant topics, while the value "terminals" refers to the resources corresponding to the terminal topics generated in each branch of the topic generation procedure.

If the context expression includes no relations, resource types, themes or range, then the interpretation of such an expression is respectively *all relation types*, *all resource types, no themes*, and *full range*. The list of resources is obtained by adding to an initially empty list, all resources linked to the set of topics specified in the context expression, followed by generating all children topics of the latter set of topics, by applying the relations specified in the context expression. This procedure is recursively applied to the list of the newly generated topics until no more new children can be generated. An additional filtering is performed at each step, using the themes and resource properties specified by the remaining part of the context expression.

In creating a context expression, the user may constrain the retrieved resources to introductory material with easy explanation of the key concepts, by selecting the theme "Beginner". In addition, he may indicate his preferred resource types, e.g. "Online notes" or "Examples". The user may also wish to constrain the resources based on the resource coverage of the topical structure generated by the context expression. Assume that the topic "List Processing" includes subtopics "List Representation" and "Operations on Lists". If the user wants only resources linked to the terminal topics (leafs) of the structure rooted by the topic "List Processing" (i.e. "List Representation" and "Operations on Lists"), he may specify that by including in the context expression the value "terminals" for the range.

The following is a more precise account of the context expressions evaluation.

(a) $t \& R$.

Interpretation: The set of resources accessible from a topic t following the relation R.

Example: Find all resources related to list processing, following the *class-subclass* relationship. The evaluation of the expression $t \& R$. results in the set of resources t_R , determined as follows: $t_R = \{r| R(t, t_1),R(t_1,t_2),..,O(t_n ,r)\}$. Note, that the resources can be distinguished from the topics based on the *topic-occurrence (O)* relation: resources are those entities r that are related to topics t_n through a *topic-occurrence* relation.

(b) $(t_1 , t_2) \& R \equiv (t_1 \& R), (t_2 \& R)$.

Interpretation: This equivalence relation indicates that expressions of type *(b)* are reduced to two expressions of type *(a)* connected by *"or" ("," is interpreted as "or").*

Example: Find all resources related to list processing or recursion following the *class-subclass* relationship.

(c) $t \& Q, R$

Interpretation: The set of resources accessible from a topic t following the relations Q or R.

*Example.*Find all resources related to list processing following the *class-subclass* or *partonomy* relationships.

The evaluation of the expression *t & Q , R.* results in the set of resources $t_{Q,R}$ determined as follows:

$t_{Q,R} = \{r|\ R_i(t, t_1), R_i(t_1,t_2),\ ...O(t_n,r),$ where R_i denotes Q or $R\}$

(d) *E&v, v* is a value of the property resource type.

Interpretation: The set of resources resulting from the contextual expression *E* restricted to resources of type *v*.

Example. Find all resources related to list processing that are *code* examples. The evaluation of the expression *E&v* results in the set of resources t_v determined as follows: $t_v = \{r|\ P(r) = v,\ r \in t_E$ *, where P denotes recourse type}*.

(e) *E&T, T* is a predefined theme.

Interpretation: The set of resources resulting from the contextual expression *E* restricted to the theme *T*.

Example. Find all resources related to list processing for beginners. The evaluation of the expression *E&T* results in the set of resources t_T determined as follows: $t_T = \{r|\ T(r),\ r \in t_E$, where *T* denotes *a theme}*. For the sake of simplicity we abuse the notation by denoting *T(r)* to mean recourses accessible within the theme *T.* In TM4L all topic characteristics can have a *theme*. The theme concept is used as a constraint on the interpretation of a topic characteristic (e.g. hiding resources related to *difference lists* from the users searching resources for Prolog beginners).

3.2 Information Exchange in Context

The underlying assumptions in the following considerations regarding our context specification are that each individual has his own conceptualization of the world, which is partial and reflects the individual's viewpoint on the domain, e.g. a viewpoint on the organization of a library collection and services (Bouquet et al. 2005; Bouquet et al. 2004; Dichev & Dicheva 2005; Dicheva & Dichev 2006). Current research in context mapping (including ontology mapping) focuses mainly on improving communication between computers or between communities and computers with less attention to communication between humans (as individuals) and computers.

The task of context mapping between a human and an autonomous agent such as computer is in general unsolved problem. The main source of difficulty comes from the fact that computers typically deal with explicitly represented contexts while the human context is not explicitly represented. Obviously in such cases mappings cannot be defined in advance, as they presuppose a complete understanding of the two conceptualizations, which in general is not available. While representing communities' context is feasible, representing contexts on an individual level may require unrealistic efforts. One possible approach is to try finding approximate, partial mappings assuming that these mappings are discovered dynamically via some kind of *mapping negotiation* (Bouquet, Serafini, & Zanobini 2005).

A context as an individual and partial depiction of a domain is not independent from what holds in other contexts. The representation of two contexts can describe a common part of a given domain. This is the case when the two contexts *overlap*. The central idea in our approach to mapping negotiation is based on: (i) developing methods for detecting overlaps in two contexts; (ii) using part of the detected overlaps as a means to partially understand the population of the other context. The second part of our approach is described in the following section through *context mapping*. It addresses context dependencies observable when an individual is interested in importing information by trying to (partially) make sense of the context of another agent.

4 Implementing Contexts in TM4L

4.1 Defining Contexts in TM4L

As we already mentioned, TM4L supports "user-defined" contexts. Our approach to contextual retrieval involves two distinct stages. The first stage includes constructing a *contextual query* which is translated to a contextual expression (as described in Section 3). Contextual queries can be performed independently of the second stage; they have independent value in terms of locating resources. In the second stage, the set of resources found in the first stage are used as *seeds* for locating some other relevant resources. The idea is to use the resources corresponding to the user's context in order to provide an orientation with regard to the organization of the collection. Semantic search requires understanding of the context in which a particular search occurs. By showing the location of the resources found during the first stage with respect to the topical grouping of the other resources, we can provide an aid to schematic cognition of the structure of the collection. This process can be viewed as a partial mapping between the user's view of the collection and the author's view.

To clarify our intuition, let us consider an analogy with the library search. Assume you want to locate some books on Programming Languages such as Java, Perl, Python, Prolog, and Lisp. You tell the librarian that you are looking for "Introduction to Java" showing the Dewey decimal classification, e.g. QA76.73.J38 L52 2005. The librarian knows where to look for the book and takes you to the right shelf. Now you are left in the right place for your task; you can locate other relevant titles, without any further assistance from the librarian. To make the analogy clear, assume that from the collection of resources found in the first stage the user selects certain relevant resources and asks the system to display them within the collection. If the system is able to show them linked to the minimal set of topics containing them, this display will be a guide to the right topics (right shelves) to explore further.

Context expressions, as defined in the previous section, provide an internal language for defining contexts and filtering resources. They are intended for the TM4L engine and are invisible to the users. Instead, the users define contexts, using as a vocabulary the objects defined in the current topic map; the latter are provided to them through a friendly and intuitive interface (see Fig. 1) - as lists of features (topics, relationship types, resource types, and themes) to select from.

Figure 1 shows the TM4L Context interface. The user defines a context by selecting context specifiers from the *Specify Context* pane and *Add Topics*, *Add Association Types*, *Add Resource Types* and *Add Themes* pop-up windows. In the example below the following context specifiers have been already selected: *Topics: Prolog* and *List Processing*, *Association Types: Superclass-Subclass, Resource Types: Examples, Themes: Advanced*. The topic *Prolog Lists* being selected in the *Add Topics* window is a new topic to be added to the topic specifiers of the current context. The specified range of coverage is *All Nodes* (*Full*). When the user completes the selection and runs the query, the system will retrieve the resources that satisfy the context definition and display them in a separate window.

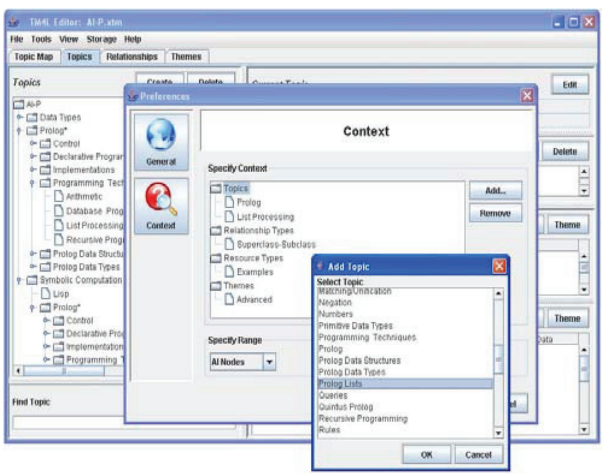

Fig. 1. Defining a context in TM4L.

4.2 Context Mapping and Structure Cognition

Context mapping is a relation between a context called *source* and another context called *target*. The mapping can be formally represented by a triple $< c_s; M; c_t >$, where c_s and c_t are the source and target contexts, respectively and M is the mapping, i.e. the relation between the explicit representations of c_s and c_t. For a Topic Map representation we define the mapping (a special case of (Bouquet 2005) as a triple $<c_s; f; c_t>$, where f is a mapping function

$$r_1 \xrightarrow{f} o(t, r_2),$$

mapping a resource r_1 from source context c_s to resource r_2 from target context c_t. such that r_2 is linked to a topic t through an occurrence relation $o(t,r_2)$. It is possible that f maps different resources from the source context into resources in the target paired with different topics. This would mean, for example, that all resources of the source collection are scattered in a disorganized fashion over the target structure. However, our intuition indicates that such a disordered mapping is unlikely in related contexts. Instead, in most cases this mapping will be distributed between logically related topics.

We denote by $A_s = \{r_i | \ r_i \in c_s\}$ the set of the source resources and by $A_t = \{r_i | \ o(t, r_i) \in c_t\}$ their images in c_t, where $f(A_s) \subseteq A_t$.

Assume that there is a device such that for each $r_j \in c_s$ it is able to displays r_j*'s* image in c_t. Function f supplemented by such a device is able to reveal the structure of the target context c_t and make it cognizable for an external agent observing the distribution of f's images. This task is problematic however as it presupposes that we know entirely the source context c_s.

Assume now that the agent carrying the context c_s is able to express partially fragments of its context using the vocabulary V_t of the target context. Let c_1 denotes a fragment (portion) of the context c_s expressed using the vocabulary V_t. Let $A_p = \{r_i | \ r_i \in c_1\}$ be the set of all resources of c_1. The fact implies that $f(A_p) \subseteq A_t$. Denote next by T_p the set of all tuples $o(t, r_i)$ with a fixed topic t (e.g. the set of all resources r_i linked to the topic t): $T_p = \{r(t, r_i) | \ r(t, r_i) \in c_t, i= 1,2,..,m\}$. Assume now that there is a device such that for each $r_j \in c_s$ it is able to display r_j*'s* image in c_t within T_p (e.g. r_i within the group of the corresponding resources linked to t along with t and its ancestors).

The key insight here is that the knowledge acquired by observing the traces of $f(A_s)$ (e.g. the *locations* and the *distribution* of resources from A_s inside the target context structure) will help the agent in exploring the target context c_t in a more systematic way and make the search for *relevant recourses* $r_j \notin T_p$ more logical. The actual goal of this approach is to provide a framework for information exchange based on topic maps *not by eliminating differences* between TM authors and users or between users but by offering a system that enables *semantic interoperability* between different types of users.

4.3 Use of Contexts

TM4L allows formulation of *context queries*, which enable on one hand, to specify the boundaries of the contextual space, and on the other, to filter the result on the basis of certain semantic properties.

The advantage of such a customized context is that it brings in results with double utilization. First, they can be used as resources satisfying the specified conditions; second, they provide a starting point for further exploration for relevant resources. Our approach to relevancy exploits the fact that resources on the same subject are typically *shelved* together. By submitting the samples from the contextual query, the user is just pointed to the topics where he can look to find other relevant material.

Suppose that a user has defined a context as {"Prolog", "Advanced material", "Examples"} and has all corresponding resources displayed. If the user selects a resource that happened to be related to "Recursion" (e.g. http://ktiml.mff.cuni.cz/~bartak/prolog/learning.html), chances are for this resource to be shown under the topic "List Processing" in the *Partonomy* (*chapter-subchapter*) view of the collection. Using then *neighborhood navigation*, the user can find some relevant examples in the topic map author's original (default) context. Such an approach allows a user looking, for example, from a "Programming Language" perspective to find resources placed under "Models of Computation".

The implementation of TM4L, including the described here extension, is in Java and uses TM4J, an open source API providing a topic map engine (www.tm4j.org).

5 Related Work

In the first TM applications the most common way of searching topic maps was by walking through topics, occurrences and associations. While such approach is suitable for small and not complicated maps, it does not turn out to be useful for large sets of topics. It became obvious that there was a need of creating a query language specialized for topic maps. There are several different proposals for TM query languages:

- Topic Maps Query Language – TMQL (Wrightson 2000) - an XML-based extension of the Structured Query Language (SQL), developed for meeting the specialized data access requirements of Topic Maps.
- AsTMa? (Barta, 2003) – a query language similar to SQL, with queries in the form of 'LET – IN – WHERE – RETURN' expressions. AsTMa? is part of the AsTMa* language family, a set of languages for Topic Map authoring, constraining, manipulation, and querying.
- Tolog (Garshol 2001) – a query language based on the logic programming language Prolog. Using Tolog one can ask for all topics of a particular type, the names of all topics of a particular type in a particular scope, etc.

However, these languages are designed for selecting specified TM elements within the whole (possibly scoped) topic map. In our approach, we can constrain the query to certain localities defined by a set of topics and a set of relations. With regard to its constraining power, our proposal is more expressive also than SPARQL (Dodds 2005) - the query language defined in terms of the W3C's RDF data model, since it allows specifying look-up boundaries based on chaining relations from a specified set. For example, we can ask for all predecessors (parents, grandparents, grand-grandparents, etc.) of a specific topic with regard to a specific transitive relationship, e.g. *is-ancestor-of*. Analogously, one can retrieve the prerequisites of a topic by chaining *is-prerequisite* relations.

In terms of Web, a great part of the contextual search work is centered around automatic building of user profiles on the basis of user's previous searches, search results, and Web navigation patterns (Bharat 2000; Kraft et al. 2005). The information system uses the profiles, representing users' interests, for refinement of future searches. In the Context Learning approach (Goker 1999; Goker et al. 2004), the focus of learning is on judged relevant documents, query terms, or document vectors. Context as a query is another approach (Henzinger 2003), which treats the context as a background for a topic-specific search, and extracts the query representing the context depending on the task at hand. In the proposed approach we provide a vocabulary and language to enable users to express the task- and context-dependent features of the information of interest.

6 Conclusion

Efficient information retrieval requires information filtering and search adaptation to the user's current needs, interests, knowledge level, etc. The notion of context is intrinsically related to this subject. In this paper we propose an approach to context modeling in Topic Map-based digital library applications. It is based on the standard TM support for associations and scopes and defines the context as an abstraction of grouping of related information based on a specified task. The proposed model of context is utilized in an extension of TM4L, an e-learning environment aimed at supporting the development of efficiently searchable and reusable learning repositories, focused on aiding the retrieval of relevant information.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. DUE-0333069 "NSDL: Towards Reusable and Shareable Courseware: Topic Maps-Based Digital Libraries" and Grant No. DUE-0442702 "CCLI-EMD: Topic Maps-based courseware to Support Undergraduate Computer Science Courses".

References

Barta, R. 2003. AsTMa? Language Definition. Bond University Technical Report,, Online: http://astma.it.bond.edu.au/astma%3Fec.dbk?style=printable.

Bharat K. 2000. Searchpad: Explicit Capture of Search Context to Support Web Search. *International Journal of Computer and Telecommunications Networking*, North-Holland Publishing Co., 33(1-6), 493–501.

Biezunski, M., Bryan, M., and Newcomb, S. 2002. ISO/IEC 13250:2000 Topic Maps: Information Technology,http://www1.y12.doe.gov/capabilities/sgml/sc34/document/0323.htm. [Last viewed April 15, 2006].

Bouquet P., Giunchiglia F., Van Harmelen F., Serafini L., and Stuckenschmidt H. 2004. Contextualizing Ontologies. *Journal of Web Semantics*, 1(4), 1-19.

Bouquet, P., Serafini, L., and Zanobini, S. 2005. Peer-to-Peer Semantic Coordination. *J. of Web Semantics*, 2(1).

Dey, A.K. 2000. *Providing Architectural Support for Building Context-Aware Applications*. Ph.D. Thesis, College of Computing, Georgia Institute of Technology. Online: http//www.cc.gatech.edu/fce/ctk/pubs/dey-thesis.pdf.

Dichev C. and Dicheva D. 2005. Contexts as Abstraction of Grouping, *Proceedings of the Workshop on Contexts and Ontologies*, 12th National Conf. on Artificial Intelligence, AAAI 2005, Pittsburgh, 49-56.

Dicheva, D. and Dichev, C. 2006. TM4L: Creating and Browsing Educational Topic Maps, *British Journal of Educational Technology - BJET*, 37(3), 391-404.

Dodds, L. 2005. Introducing SPARQL: Querying the Semantic Web, *O'Reilly XML*, Online: http://www.xml.com/pub/a/2005/ 11/16/introducing-sparql-querying-semantic-web-utorial.html.

Garshol, L.M. 2001. Tolog: Topic maps query language. *Proceedings of the XML Europe 2001 Conference*, IDEAlliance, http://www.ontopia.net/topicmaps/materials/tolog.html.

Giunchiglia F. 1993. Contextual reasoning, Epistemologia, Special issue on *I Linguaggi e le Macchine,* XVI, 345–364.

Goker A. 1999. Capturing information need by learning user context. In *Proceedings of Sixteenth International Joint Conference in Artificial Intelligence: Learning About Users Workshop*, 21–27.

Goker A, Watt S., Myrhaug H.I., Whitehead N.,Yakici M., Bierig R., Nuti S.K., and Cumming H. 2004. User context learning for intelligent information retrieval. In EUSAI'04: *Proceedings of the 2nd European Union symposium on Ambient Intelligence*, ACM Press, 19–24.

Guha, R. 1991. Contexts: A Formalization and Some Applications, *Technical Report ACT-CYC-423-91*, MCC, Austin, Texas.

Henzinger B, Chang B.W, Milch B., and Brin S. 2003. Query-free news search. In *Proceedings of the 12 Twelfth International World Wide Web Conference* (WWW-2003), Budapest, Hungary, May 20-24 2003.

Kraft R., Maghoul F., Chang C. C. 2005. Y!Q: Contextual Search at the Point of Inspiration CIKM 2005, ACM Fourteenth Conference on Information and Knowledge Management (CIKM), Bremen, Germany, November, 2005.

McCarthy, J. 1987. Generality in Artificial Intelligence, *Communications of the ACM*, 30(12) 1030–1035.

Park, J., Hunting, S. 2002. *XML Topic Maps: Creating and Using Topic Maps for the Web*, Addison-Wesley.

Wrightson A. 2000. TMQL Draft (Topic Map Query Language). Ontopia, BSI. Online: http://www.y12.doe.gov/sgml/ sc34/document/0186.doc.