

A Comparative Study of RDF and Topic Maps Development Tools and APIs

Fakhre Alam, Shaukat Ali, Muhammad Abid Khan, Shah Khusro, Azhar Rauf

Abstract— Users' freedom resulted into the exponential growth of the World Wide Web containing billions of web pages but aggravated the problem of finding specific information precisely. Semantic Web technology is deemed as solution to the problem by annotating web resources with metadata. To realize the idea of Semantic Web, several technologies are invented in the past several years. Resource Description Framework (RDF) and Topic Maps are the two prominent technologies developed for turning the vision of Semantic Web into reality by facilitating knowledge integration and sharing. Both RDF and Topic Maps provides constructs to enrich web contents with semantic information to make them useful for both humans and machines. However, RDF and Topic Maps differs and suffers from the interoperability problem due to difference in their fundamental architectures, approaches, techniques, and owner organizations and might result into two separate islands on the Web of future. To facilitate developers in understanding the technologies and accelerate their development process to produce valuable applications, a number of tools are invented by the researchers, academia, and organizations for both RDF and Topic Maps. This paper presents a comprehensive overview of the tools by investigating, analysing, classifying, and comparing them using a set of parameters. The available tools are broadly classified into three categories: storage, editing, and visualization, and their interoperability problem is addressed. We hope that this paper will provide a compact platform for both researchers and users to help them in selecting tools rich enough to satisfy their needs effectively.

Index Terms— World Wide Web; development tools; API; Semantic Web; RDF; Topic Maps

I. INTRODUCTION

Exponential growth of the World Wide Web aggravates the problem of finding specific information precisely almost impossible because of the losing rich semantics and necessitates human presence for Web contents interpretation. Tim-Berners Lee, the creator of the Web, suggested Semantic Web to solve the information retrieval problem by associating metadata to Web contents to make them machine understandable and processable. Resource Description Framework (RDF) and Topic Maps are the two lethal technological weapons invented for turning the vision of Semantic Web into reality. These technologies share the same vision of making the Web a useful resource by implementing metadata relationship between the web resources.

Fakhre Alam, Shaukat Ali, Muhammad Abid Khan, Shah Khusro, Azhar Rauf, Department of Computer Science, University of Peshawar, Peshawar, Pakistan. Email: khusro@upesh.edu.pk. Manuscript received January 28, 2014; revised July 27, 2014 and October 23, 2014.

Metadata relationship implementation could be beneficial in numerous ways such as to optimize search engines to find Web resources quickly and easily, and enables sophisticated intelligent software agents development which will work on behalf of users and will exchange and share knowledge as well as catalogue websites and libraries effectively.

The two technologies work the same but in different ways. RDF technology supports infrastructure for the Semantic Web by creating RDF models. A RDF model is composed of statements where each statement relates two web resources by using the analogy of subject, predicate, and object. Topic Maps technology supports expressing and exchanging information on the Web in a meaningful way to be manipulated quickly and easily. Originally this concept was used for book indexes, but after further generalization and with the introduction of meta-model, this technology is also used for wider applications e.g. Semantic Web. Anything on the Web can be expressed by the Topic Maps in the form of topic, association between topics, and the occurrences of topics and associations.

The growing numbers of Web resources with complex relationships have added increasing complexities into RDF and Topic Maps technologies. To decrease time and cost, and to increase performance and conformance, software tools are required to automate the development process and help programmers in creating, restoring, maintaining and supporting different systems¹. To promote and take full advantage of RDF and Topic Maps technologies easily, a number of commercial and open source tools are developed for analysing performance, managing and creating applications, linking and integrating different RDF and Topic Maps components, retrieving information quickly and intelligently, browsing and visualizing information, and errors reporting. These tools will provide great ease in creating, editing, and managing RDF and Topic Maps and minimizes the associated complexities.

While being overwhelmed with a number of tools, it is rather cumbersome for researchers and users to select an appropriate tool rich enough to provide convenient environment for creating RDF and Topic Maps and addresses the associated expressing problems effectively. This research paper is aimed to provide a comprehensive overview and analysis of the available RDF and Topic Maps tools, covering all of their possible aspects, and pros and cons. It also attempts to find out how much RDF and Topic Maps tools can be used together. Main contributions of this paper include:

¹ <http://www.superwebgroup.com/what-do-software-tools-do/>.

- The key contribution is the collection, study, and organization of on hand knowledge about RDF and Topic Maps tools in a concise manner to help readers in understanding and instigate their interests. The tools are compared using a number of parameters to provide more insight knowledge about the tools.
- The work is almost unique in its integrity and opens new areas of research. As no prior work exists analyzing and comparing tools belong to the two Semantic Web islands.
- The paper is expected to provide a compact platform for researchers to find new research dimension and for users to select a tool satisfying their needs effectively.

II. SOFTWARE PACKAGES CLASSIFICATION

Software packages are essential for computer technologies, because they not only enhance the quality of developers but also increase the motivation of humans to work properly with computers and associated technologies. Software packages can be used for several purposes in computer science including performance analysis, system debugging, ensuring precision, creating applications, memory testing and correction, relating and integrating different components, and text editing etc. Software packages provided for users' usage can be either software tools or APIs. Software tools are explicitly installed on users' computers for varied purposes such as managing components of an organization, execution of a specific operation, and helping developers in their creations etc. Organizations desiring to share, and integrate their software components with other systems over a network provides APIs. In WWW paradigm, APIs are software interfaces made available publically by software companies to help programmers to develop their new applications by integrating functionalities Web accessible tools and applications without any requiring human interference or knowledge. APIs can be advantageous for developers, users, and software providers such as speeding up software development process, no extra charges for software tools are associated hardware, and no staff training requirement etc.

RDF and Topic Maps are provided with a number of software tools and APIs by the researchers, academia, and organizations for their smooth and accurate implementation. Although the tools differs significantly in terms of encoding languages, programming constructs, and features etc. but they aim to facilitate implementation of Semantic Web software systems in their own ways either on individual computer or on the Web. Generally, few of the tools and APIs are rich enough to provide complete set of functionalities, whereas, others enables functionalities in a particular domain. Therefore, the available tools and APIs can be further categorized into storage, editing, and browsing/visualization as shown in Fig. 1.

- Storage provides an environment of automatically storing, managing and recovering the instructions as well as data for helping administrators in increase in their productivity and decrease in their mistakes. The editing tools are used to create, change or modify the

software system or application program by creating programming code in the backend. Browsing and visualization tools are used to view create and modify data in graphical form for easy understanding.

- Storage APIs the capabilities of software systems by providing virtual interface among multiple storage systems for storage, transfer, and proper management of data and files. Editing APIs helps to add editing functionalities (i.e. adding, deleting, combining, and reshaping of data) to applications which could be useful for creating interactive and expressive Web application, and controlling resources. Browsing/visualization APIs enables applications to create graphics, and charts over structured and non-structured data which could be integrated directly into websites.

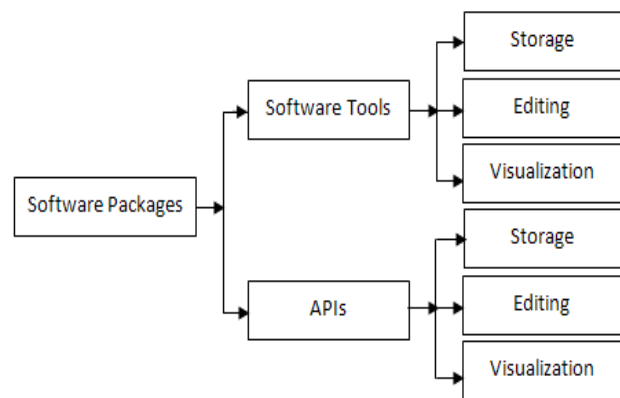


Fig. 1. Software packages classification.

III. SOFTWARE PACKAGES FOR TOPIC MAPS

Topic Maps technology was developed by ISO for expressing and exchanging knowledge in a meaningful way. The technology was originally intended for representing books indexes but extended by researchers for wider applications including exchanging and conveying knowledge on the Semantic Web. Topic Maps represents information in the form of topic, occurrences, and associations between topics and their occurrences. To facilitate creation, maintenance, and implementation of Topic Maps a number of commercial and open source software tools and APIs are produced by the researchers, academia, and organizations for storage, editing, and browsing/visualization as shown in Fig. 2.

A. Topic Maps Storage Tools and APIs

A Topic Map consists of a rich set of metadata information about concepts, objects, and their relationships in an explicit format to be automatically processed by the machines. These information needs to be stored in an organized format which could be readily accessed by the applications effectively for performing operations. Storage tools provide guaranteed methods for storing, managing, and retrieval of information. Information in Topic Maps can be stored in numerous ways including relational databases and files. Several types of commercial and open source tools (i.e. called Topic Maps engines) which stores Topic Maps data automatically or manually, provide user

friendly interfaces, and provides unique APIs for effective access and retrieval of stored data [1]. Table I analyses and compares the on hand Topic Maps storage tools and APIs using a set of evaluation parameters.

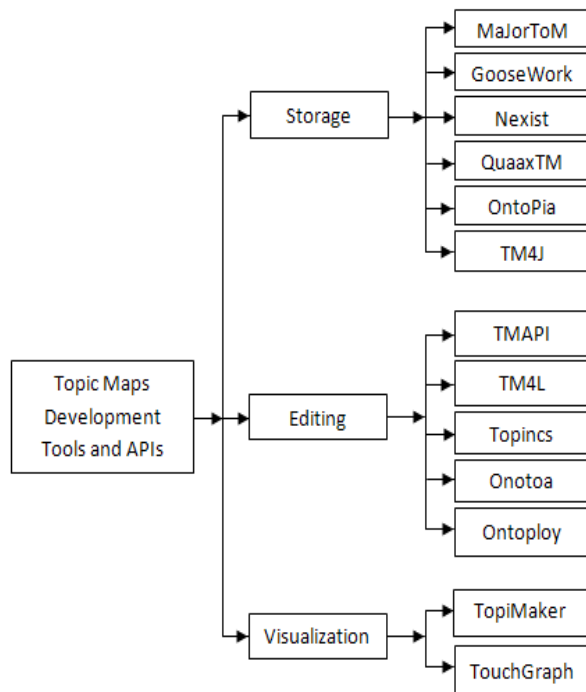


Fig. 2. Topic Maps development tools and APIs.

1) *Ontopia*: Ontopia Topic Map Engine is developed by Ontopia. The engine is a java based open source and cross platform engine for storing Topic Maps data, released under Apache 2.0 license. The tool is expressive enough providing a comprehensive set of services which are easy to use, thus making him worldwide acceptable in Topic Maps paradigm [2]. The engine is rich enough to build and manage all of the operations of an effective Topic Maps application [2]. Using the engine Topic Maps data can be accessed even from XML documents, modified, and stored in DB2 database. The engine has core API and built-in support for Tolog (Topic Maps query language) which could be used by applications for effective accessing and retrieving of Topic Maps data stored in a database. Some distinguishing features of the engine includes complete correspondence with XTM 1.0, reading and writing XML files in different serialization formats (i.e. XTM, LTM and HyTM), robustness, scalability at a large scale, storing data in relational databases, and a rich set of utilities for a variety of operations (e.g. filtering of scope, selection of name, merging, association, and character encoding etc). The tool is applicable to enterprise level systems and having pluggable backend requiring programmers to use a single API for the whole system. Although the Engine is a full fledge storage system for Topic Maps data but it suffers from the limitation of being not applicable for all types of applications development.

2) *QuaaxTM*: QuaaxTM also called PHP Topic Maps engine is a long-term storage for Topic Maps data [3]. QuaaxTM stores multiple Topic Maps and the storage server is compatible with TMDM which provides integrity and support. QuaaxTM can merge topics automatically while supporting every type of import and export from Topic Maps to files and vice versa. The engine uses PHPTMAPI (i.e. an API for PHP5) for creating and manipulating Topic Maps. QuaaxTM is a small scale storage tool supporting transaction and referential integrity due to using MySQL and InnoDB² as storage engines. QuaaxTM is an open source tool released under GNU Library or Lesser General Public License version 2.0 and suitable for implementing Topic Maps on the Web.

3) *TM4J*: TM4J Topic Map Engine is an open source tool written in java for parsing and storing Topic Maps in memory [4]. This engine comes with two types of storage formats called Hibernate RDBMS Backend and In-memory Backend along with the core Topic Maps API query and serialization services. Programmers can create and change Topic Maps structures using a wide-range of APIs supported by TM4J Topic Map Engine. The engine stores Topic Maps data in both RDBMS and Object Oriented Database format while providing an easy support for querying, and importing and exporting from one file into another. The engine provides several command line utilities for both Windows and UNIX platforms for manipulating Topic Maps including merging, performing statistical calculation, applying XTM 1.0 rule for compression and conversion, and creating a small Topic Maps from a group of large Topic Maps³. The TM4J Topic Map Engine has user friendly interface providing TM4J Administrator tool for manipulating Topic Maps in persistent store and importing, exporting and deleting from/into XTM/LTM files. The engine supports querying Topic Maps using Tolog and TMAPI.

4) *The Merging Java Topic Maps Engine (MaJorToM)*: The Merging Java Topic Maps Engine (MaJorToM) is java based open source tool released under Apache license 2.0 as backend storage for Topic Maps. This engine is based on Topic Maps API v.2 and comes with new and advanced features with simple and flexible architecture. MaJorToM deals with the application domain and Information individually [5]. This engine is downward compatible with all other Topic Maps engines based on TMAPI. Using a set of objects, the engine can communicate with and access the stores encapsulating knowledge. The engine has features of include time and space modeling and independent transaction creation. The architecture of the storage tool is simple, user friendly, flexible and separate the information and application domains from each other [5]. Application can communicate with the central store in a simple way and the central storage implements sophisticated methods for effective use.

² <http://dev.mysql.com/doc/refman/5.0/en/innodb-storage-engine.html>

³ <http://tm4j.org/tm4j/docs/tools/index.html>

TABLE I. ANALYSIS AND COMPARISON OF TOPIC MAPS STORAGE TOOLS AND APIS.

Tool/API Name	Availability	Development Platform	Platform Support	Operating System	Pluggable /non-pluggable	Databases	APIs	Query Language	Market value
Ontopia	Open Source	Java	Independent	Any operating system	Pluggable	DB2	Core Java API	Tolog	Very High
QuaaxTM	Open Source	PHP	Independent	Any operating System	Pluggable	MySQL, InnoDB	PHPTMAPI	-----	Low
TM4J	Open Source	Java	Independent	Windows and UNIX	Pluggable	RDBMS and ORDBMS databases	TMAPI, and other APIs	Tolog	High
MaJorToM	Open Source	Java	Independent	Any operating system	Pluggable	MySQL, PostgreSQL	TMAPI version 2.0	-----	Medium
Goose-Works	Open Source	Python and C	Independent	Any operating system	Pluggable	RDBMS with slight variation	Core API	L1, and STMQL	High
Nexist	Open Source	Java	Independent	Any operating system	Non pluggable	RDBMS	Core API base on XTM	-----	Low

5) **GooseWorks**: The GooseWorks Topic Map Toolkit developed by Jan Algermission and San Hunting [1] is an open source tool-kit release under Apache license for processing Topic Maps consisting of an engine and API developed in Python and C. The engine stores Topic Maps in different form of relational databases and implements the model proposed by Newcomb and Biezunski. The tool can work with any type of Topic Maps files and mark-up while providing command line interface for merging and filtering of Topic Maps. The Toolkit uses L1 and STMQL for importing and querying Topic Maps respectively. The toolkit supports for Hyper Text Transfer Protocol (HTTP) for viewing statistical information and Maps in a web browser.

6) **Nexist**: Nexist is an open source and platform independent tool written in Java, which provides effective API for client-server environment [6]. Nexist API, based on XTM, provides collaborative and multi-user environment for creating and modifying Topic Maps information directly. Topic Maps structure and related information can be stored persistently in relational databases by Nexist. This tool is mostly used in educational environment for sharing and exchanging ideas in collaborative form.

B. Topic Maps Editing Tools and APIs

Computer programmers typically finds editing tool quite beneficial as they would be helpful in accelerating the development process, decreasing development tasks, saving development time, and decrease in development mistakes. Topic Maps are provided with sophisticated editing tools providing environment to create, update, and change a Topic Maps while producing programming code in the backend. They provide easy ways for users to interact with Topic Maps. Table II analyses and compares the on hand Topic Maps editing tools and APIs using a set of evaluation parameters.

1) **TM4L**: TM4L Editor is an open source Java based editor for the creation and editing of Topic Maps. The editor is a part of TM4L (Topic Maps for e-Learning) environment, which provides learning repositories for Topic Maps creation and maintenance using ontology driven approach [7]. TM4L

Editor encourages Topic Maps data creation and maintenance in numerous ways (i.e. graphical, tabulated, and form based) and Topic Maps data reuse and extension by merging existing repositories with the newly created repositories without losing their meaningful structure. The editor uses graphical mode for topics and their association creation, and table and form modes for entering details about resources and their occurrences in a more semantic way [7]. Using the multi-mode editing capability, TM4L Editor is flexible enough for programmers to edit and design Topic Maps effectively. TM4L editor is platform-independent and can be used on any platform including Windows and Linux/Unix with Java JRE or SDK, version 1.5 or above.

2) **Topincs**: Topincs is an open source client-server based editing tool for Topic Maps using Apache, MySQL, and PHP on the server and any web browser on the client side [10]. Topincs is platform independent supporting Linux, Windows and Mac OS X platforms, however the recommended one is Linux. Topincs tool-kit is composed of three parts: a client interface for editing maps, a server for storing Topic Maps data, and a wiki for Topic Maps data. Users uses HTML based web browser for creating and maintain Topic Maps documents containing semantic information in the form of wiki stored in repository. Topincs provides collaborative and multi-users environment where a community can contribute and share their knowledge within a Topic Maps based wiki. Like TM4L, Topincs also supports creating of Topic Maps in the form of tables and forms. Topincs has support for Android devices enabling users to view, edit, and manipulate their data anywhere using their mobile devices [10]. Topincs has time and space based tracking feature and multi-language support for presenting and translating information.

3) **Onotoa**: Onotoa is a free and eclipse based graphical editor for Topic Maps schema called TMCL (Topic Maps Constraint Language) [11]. Onotoa is used to create and define constraints for Topic Maps ontologies. Onotoa can be installed on any operating system in both stand-alone mode and as a plug-in for eclipse, however, requiring the pre-installation of JRE Version 5.0. Onotoa has a visual editor for easy understanding of Topic Maps

diagrams instead of understanding hard code and users can also import and export CTM (Compact Topic Maps) files from and to the visual editor. The tool provides the facility of creating multiple diagrams for Topic Maps schema. The interface provided by Onotoa is similar to UML, due to which Topic Maps schema can be created and edited in an easy way [8]. The interface is also equipped with several other features in a separate view like validator, which shows the errors and analyses the schema for correctness. The searching tool is also very effective and supports user-defined context while finding specific types of topic.

4) *Ontopoly*: Ontopoly is part of the Ontopia Knowledge Suit (OKS) used to create Topic Maps data step-by-step in table and form mode [9]. User interface is web-based representing elements in HTML and providing a convenient environment for user to design their own Topic Maps ontology without any clutter and confusion. Ontopoly is suitable in a distributed environment where multiple users not only share their knowledge but create, edit, and access the Topic Maps from the same repository. However, view would get more and more complex with the increasing number of topics and associations. The editing environment provided by Ontopoly user friendly and easy for creating Topic Maps in two steps⁴. In the first step, ontology and schema for Topic Map is created and in the next step the ontology is populated by creating instances of resources.

5) *TMAPI*: TMAPI is a java based Topic Maps editing API, is the standard way to interact with XML Topic Maps programmatically from Java. The TMAPI specification defines a set of core interfaces which must be implemented by a compliant application as well as a set of additional interfaces which may be implemented by a compliant application or which may be built upon the core interfaces [4]. This API is used to retrieve and process data stored in Topic Maps. TMAPI hopes to do for topic maps what SAX and DOM did for XML - provide a single common API which all developers can code to and which means that their applications can be moved from one underlying platform to another with minimum fuss.

TMAPI has been developed in an open process by developers working on topic map processors and topic map applications and placed into the public domain. There are no restrictions on its use.

C. Topic Maps Browsing and Visualization Tools and APIs

The increasing growth of information the Web necessitates the need of some techniques to help users for manipulation and retrieval of required information easily and quickly. Topic Maps browsing and visualization tools can solve users' problems of retrieving relevant information more effectively because it is far easy to deal with visual interface as compared to textual interfaces. The complex set of associations between topics in a Topic Map makes it difficult for users to interact and understand the complex arrangement of information from XML based code. Therefore, the issue of Topic Map visualization and navigation is essential. To retrieve a specific topic from Topic Map, users can issue a query from a visualization tool against a Topic Map and the relevant information will be retrieved quickly. Topic Maps visualization tools provides an easy to use environment to arrange all topics, associations and occurrences in such a way to be manipulated and navigated for relevant information in an effective way. Visualization tools use different methods to arrange Topic Map information in a way which should be easy to understand, navigate, and view. These methods include index or list-based visualization, tree-based visualization, graph-based visualization, and map-based visualization. Table III analyses and compares the on hand Topic Maps browsing/visualization tools and APIs using a set of evaluation parameters.

1). *TopicMaker*: TopicMaker is an open source tool, developed in Java to visually represent the structured knowledge contained in a Topic Map conveniently. TopicMaker represents a Topic Maps structured graph in a fixed 2D layout plane in an interactive 3D world [10]. Topics can be evaluated to planes that are parallel to the ground plane. In this graph, topics in the form of nodes are displayed in a cylindrical shape while the associations in the form of edges are represented as lines between the

TABLE II. ANALYSIS AND COMPARISON OF TOPIC MAPS EDITING TOOLS AND APIS.

Tool/ API Name	Availability	Development Platform	Platform Independent	Operating Systems	Pluggable /Non-pluggable	Architecture	AP Is	Market value
TM4L	Open Source	Java	Independent	Any operating system	Pluggable	Stand alone	TMAPI	High
Topincs	Open Source	Apache, MySQL, and PHP	Independent	Windows, Linux, and MAC OS X, Android, iOS	Pluggable	Web based	PHP API	Low
Onotoa	Open Source	Java	Independent	Any operating system	Pluggable	Stand alone	----	High
Ontopoly	Open Source	Java	Independent	Any operating system	Pluggable	Web based	----	High
TMAPI	Open Source	Java	Independent	Any operating system	Pluggable (Extendable)	Web based	TMAPI	High

⁴ <http://www.ontopia.net/doc/5.2.1/ontopoly/user-guide.html>

nodes. Tree-like features describing certain association types are displayed in a separate window. Features in this visualization technique including searching, filtering, importing, exporting, modifying, querying, and merging makes TopiMaker an ideal tool for Topic Maps visualization. TopiMaker makes the representation and navigation of Topic Map graph easy to understand and use by mapping it on the 2D plane in a 3D word.

2) *TouchGraph*: Topic Maps concepts in a network graphs structure can be easily browsed and visualized with Touchgraph. TouchGraph supports a wide range of relationships types where edges can be directed, uni-directed, and bi-directed. The tool helps in discovering relationships between topics by configuring the graph appearance and filter settings. Text and numerical values can be associated with nodes and edges, moreover, images can be associated with nodes as well in this tool. Touchgraph provides easy sharing and navigation for the end users. To visualize large amount of information networks requires high level of interactivity, fast manipulation, pan and zoom capability, and locality control [11]. All these types of requirements are the built-in features of Touchgraph. This open source tool is fully developed in Java and its graphical applications includes GoogleBrowser, PubMed, OI-Modeler and in Planetarium. Several others features includes support sorting mechanism, advance cluster computation reveals inherent grouping, and co-citations and co-occurrences analysis clarifies dense networks.

IV. SOFTWARE PACKAGES FOR RDF

Several types of tools and APIs are available for RDF to store, maintain, manage, edit, navigate and visualize its resources as shown in Fig.3. Each of these tools and APIs are categorically describe in the subsequent sub-sections. Generally, the tools developed for RDF are more and have high market value as compared to those of Topic Maps.

A. RDF Storage Tools and APIs

RDF data consists of RDF triples and schema information, stored in the form of files and databases. A repository which consists of files and databases define a storage mechanism for RDF store and provides a flexible and easy interface for the applications using them. Users can use APIs and other integral components of the tools to execute addition, deletion, querying, and format exchange against the data stored in a RDF repository [12].

To store RDF data consistently, numerous types of commercial and open source tools are available, storing resources in the form of triple and quad. Some of the tools are developed from the scratch, whereas, others are built using the existing tools utilizing relational databases and object oriented databases. Table IV analyses and compares RDF storage tools and APIs using a set of parameters.

1) *Jena*: Jena is an open source tool, developed in Java for storing and manipulating RDF Graph persistently as well as supporting RDFs, OWL and SPARQL. Jena 1 (first version) was released 2000 and Jena 2 (second version) with advanced features was released in 2003. Jena is platform independent and stores RDF data as graph in memory, backend database, and an inferred graph. This RDF graph can be easily manipulated by its strong and valuable API. Due to multiple and flexible RDF graphs presentations, Jena has become an eyeball for the Semantic Web programmers for easy navigating and exposing of RDF data in triples [13]. Jena has a rich API for allowing applications to manipulate stored data and supporting various RDF serialization formats including RDF/XML, N-Triple, N3 and Turtle. Jena supports query languages for RDF (i.e. SPARQL) using its ARQ engine module. Jena extends support for numerous relational databases including MySQL, Oracle, SQL Server and PostgreSQL⁵ using Java Database Connectivity (JDBC). Jena uses ranges from individuals to enterprises levels, therefore, 80% of Semantic Web applications are using Jena for RDF storage. The astonishing features of Jena has increased its market value and has encountered more than 20,000 download upto 2013.

2) *Sesame*: Sesame is an open source and platform independent tool, developed in Java for efficient and persistent storage, and retrieval of RDF data while extending its support for RDFs and SPARQL. Apart from Java, some of the sesame components are developed in other programming languages as well including python, and PHP. For Semantic Web programmers, sesame presents an enormous scale of tools effectively using the power of RDF and RDFs6. Sesame Sail API extends sesame's functionality by it with other stores such as 4store⁷, Bigdata and OWLIM⁸. Like Jena, Sesame also

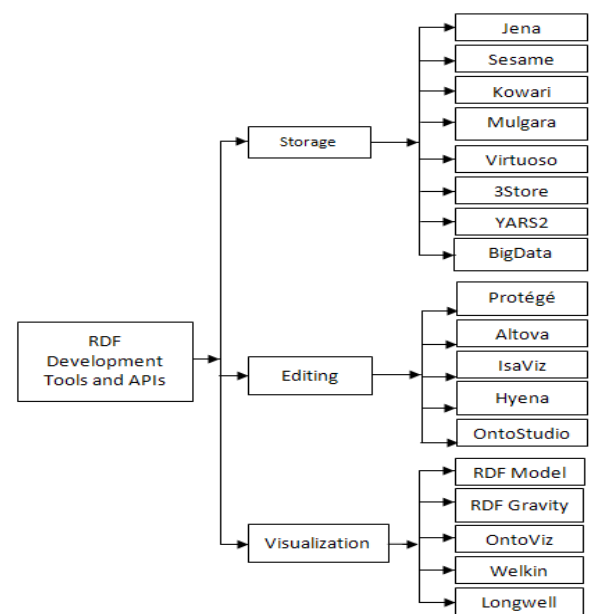


Fig. 3. RDF development tools and APIs.

⁵ <http://www.postgresql.org/>

⁶ <http://www.w3.org/2001/sw/wiki/Sesame>

⁷ <http://4store.org/>

⁸ <http://www.ontotext.com/owlim>

TABLE III. ANALYSIS AND COMPARISON OF TOPIC MAPS BROWSING/VISUALIZATION TOOLS AND APIs.

Tool/ API Name	Availability	Development Platform	Platform Dependency	Operating Systems	Pluggable /Non- pluggable	User Interface	APIs	Market value
TopiMaker	Open source	Java	Independent	Any operating system	Pluggable	GUI	----	Very High
TouchGraph	Open source	Java	Independent	Any operating system	Pluggable	GUI	----	High

3) Supports some of the relational databases e.g. MySQL and PostGreSQL and query languages i.e. SPARQL. The Sesame framework consists of some additional layers and modules performing different functions such as RDF parsing and writing, abstraction from storage and inference detail, and repository's handling and accessing by HTTP server.

4) *Kowari*: Kowari is an open source tool implemented in Java for storing and manipulation of RDF and OWL data in multiple database formats⁹. The tool has built-in RDF support, full text search functionality, support of W3C specifications/guidelines, and simple query implementation. Kowari has built-in capabilities of storing large number of RDF triples in a low space, storing and retrieving RDF data quickly, and ensuring integrity and reliability but APIs (e.g. JRDF¹⁰, Jena, Web-based query interface, and SOAP etc) can be used to extend its capabilities. XA Triple storage engine increases Kowari's scalability by storing data in a 64 bit structured form, and multiple and concurrent reading sessions with online backups and permanent integrity.

5) *Mulgara*: Mulgara is an open source tool, developed in Java for storing and retrieving RDF data in SPO format. The tool is derived from Kowari and inherits most of its features from Kowari¹¹. Data stored in Mulgara can be retrieved by SPARQL queries, connection API, and command line interface called TQL. Mulgara uses simple SQL like query language, small footprint, and has the functionality of full text search. The performance and scalability of Mulgara is also high and can store large data sets. The tool is reliable due to its consistent integrity, support for full transaction, and clustering capabilities. Mulgara uses the same types of APIs used in Kowari and is platform independent. It can be easily managed due to the availability of web based tools for configuration and monitoring.

6) *Virtuoso*: Virtuoso is a Java based platform independent data storage server for RDF triples and is available in both commercial and open source licenses¹². Virtuoso architecture consists of command line loaders, connection API, support for data in multi-formats (i.e. relational, RDF, and XML), full Text indexing support, web application server, and linked data server for SPARQL querying and data uploading over HTTP. The role of Virtuoso in the current information age is much high as compared to other storage tools [14]. The performance of virtual database engine for distributed computing is high. It provides an easy access to already built data sources from

different database vendors and its simultaneous connections for client applications and services such as ODBC, JDBC, UDBC and OLE-DB to data within database engines such as Oracle, DB/2, Progress and Microsoft SQL Server. Data from diverse sources can be easily brought together with the use of Virtuoso to speedup information production. Virtuoso can create real time dynamic XML document from HTML documents and SQL databases (heterogeneous or homogenous) and associate them to SQL databases efficiently.

7) *3Store*: 3Store, 4Store and 5Store are storage tools used for RDF, developed in C, PHP, Ruby, Python and Java [14]. 3Store is a MySQL and Berkeley DB based triple store. The server software does not provide any interface to the users but can be queried by a number of services including direct RDF browser and column based view. It provides access to the RDF data via RDQL or SPARQL over HTTP, on the command line or via a C API. 3Store engine has now been replaced by 4Store because it is more reliable, efficient, easy to use and faster than 3Store. 4Store is implemented in ANSIC99 and licensed under General Public License (GPL), version 3. The platforms used for the installation and running are Linux and Mac OS and capable for holding 10+ billion triples. The extensibility mechanism in 4Store is poor. Therefore, a new version was developed by Garlik Inc.¹³ called 5Store. 5Store provides the same features like 4Store along with some additional features. Its efficiency and scalability is much better than 4Store but it is a commercial software and not publically available.

8) *Yet Another RDF Store, Version 2 (YARS2)*: YARS2 is an open source platform independent RDF storage developed in Java and released under the GNU [14]. The storage tool uses a distributed system for organizing huge amounts of graph-structured data. YARS allows for querying RDF based on a declarative query language, which offers a somewhat higher abstraction layer than the APIs of RDF toolkits such as Jena or Redland. YARS uses Notation3 as a way of encoding facts and queries. The interface for interacting with YARS is plain HTTP (GET, PUT, and DELETE) and is built upon the REST principle. It supports keyword searches and content-negotiable result formats (RDF/N3 and XML) and uses Tomcat to implement the HTTP interface. Berkeley DB is the storage environment for RDF resources in B+trees form. The interface provided by YARS2 is easy to use because it is web based interface consisting of plain HTTP. Due to some limitations such as its applicability for simple graph data, the tools was not continued after.

⁹ <http://kowari.sourceforge.net/>

¹⁰ <http://jrdf.sourceforge.net/>

¹¹ <http://docs.mulgara.org/overview/index.html#overview>

¹² <http://docs.openlinksw.com/virtuoso/overview.html>

¹³ <http://www.garlik.com/home>

9) *Bigdata*: Bigdata is an open source tool developed in Java and designed for Linux platform to store RDF triples [14]. The tool is suitable and designed for order data (B+Tree) that is available under the GNU General Public License (GPL). Bigdata is an extendible tool and the extensibility can be achieved by plugging additional data services dynamically at run time. This tool enables users to load and query massive amount of data sets that is even beyond the processing capabilities of high and powerful server platform. Bigdata is useful in situation when the datasets are heterogeneous and interesting data mostly appears in non integrated form. The tool enables users to go into the detail of source and maintain the authentication of data.

B. RDF Editing Tools and APIs

RDF technology is rich enough to implement the vision of Semantic Web. However, there is a need for easy to use API's and editing tools for the programmers to easily learn and work with RDF without being going into the detail. Therefore, the problem is addressed by developing a number of editing and designing tools and APIs to facilitate programmers, having little knowledge about RDF, by providing flexible environment, and graphical interface with plethora of options and features for easy manipulation and designing of RDF data in varied serialization formats as well as import and export among different formats. Table V analyses and compares RDF editing tools and APIs using a set of parameters.

1) *Protégé*: Protégé¹⁴ is an open source editing and ontology modeling tool developed in Java. This tool comes with several types of plug-ins, and APIs to extend the capabilities of protégé. The external applications can use the core API and OWL API (extension of core API for OWL ontologies) to directly access the protégé knowledge base. Ontologies in several formats can be created, visualized, and manipulated due to the advanced knowledge modeling structures and functionality implemented by Protégé. Protégé provides a customized view, based on specific domains, for the users to enter data and create knowledge models. Protégé uses two ways model ontologies: Protégé-Frames editor and Protégé-OWL editor. The market value of Protégé is much higher as compared to other RDF tools and currently having millions of registered users belonging to varied fields. Protégé is platform independent and can run on Windows, UNIX, Linux, and Mac OS X platforms.

2) *Altova SemanticWorks*: Altova SemanticWorks is an RDF/RDFs editing and ontology designing tool enable graphical editing and designing of RDF/RDFs documents and OWL ontologies¹⁵. The trial version of this tool is freely available for 30 days but can be installed and run on Windows platform only. Altova allows users to concentrate on development tasks (i.e. logic of the semantic design etc) while performing other tasks (i.e. interchange ontology in graphical format to RDF/XML format and N-Triple format

etc) automatically. Due to this auto generated facility, programmers can see the graphical design and code by just switching from graphical view to text view. The tool is becoming practically feasible for users to create Semantic Web applications because of its intelligent editing features and visual design paradigm.

3) *IsaViz*: IsaViz is a visual authoring tool developed in Java for creating RDF documents as graphs [15]. This tool provides a user-friendly interface in which RDF graphs are created using ellipses, boxes and arcs very smoothly. In IsaViz, graphical resources can be imported and exported to RDF/XML, Notation 3, and N-Triple formats and vice versa. IsaViz can be installed on any platform, however, requires a JVM 1.3.x or later to run. Since version 2.0 as well as GraphViz for some features. IsaViz can render RDF graphs using GSS (Graph Stylesheets), a stylesheet language derived from CSS and SVG for styling RDF models represented as node-link diagrams.

4) *Hyena*: Hyena creates RDF metadata contents and supports various RDF vocabularies. Hyena¹⁶ is different from all other RDF editing tools because most of its concentration is on wiki contents and while working with it, users must be familiar with the wiki syntax. The tool is very complex and designed for professional programmers and can run in two different formats of Ajax and Eclipse both online as web application or offline as desktop application. The trial version of Hyena is freely available for Windows system. This tool can also support Active Directory and other Windows management tools.

5) *OntoStudio*: OntoStudio is IBM elclipse based framework, a commercial and widely used tool for RDF and ontology editing and designing with advance functionality of OntoBroker server used for enhancing and creating ontology at the same time¹⁷. In OntoStudio, instances of classes, properties are created first and refined later. Through graphical mapping, this tool can match multiple dissimilar structures easily and quickly. OntoStudio can be installed on Windows based platforms and as ontology languages it supports W3C-standards OWL, RDF(S), RIF, and F-Logic for the logic-based processing of rules. The tool and import information from UML 2.0, Database schemas (Oracle, MS-SQL, DB2, MySQL), Excel tables, and Outlook E-Mails. Among the most important functions are the mapping tool, the graphic rule editor and the integrated test environment. With the mapping tool it is possible to map heterogeneous structures onto each other quickly and intuitively. The graphic rule editor and the integrated test environment assures the quality of the modeling. With OntoStudio, several editors can provide and extend ontologies at the same time by using the OntoBroker Collaboration server. Queries can be exported as web services and be integrated into any application. The NeOn Toolkit is an open source licensed version of this tool.

¹⁴ <http://protege.stanford.edu/>

¹⁵ <http://www.componentsource.com/products/altova-semanticworks-installed/index.html>

¹⁶ <http://www.w3.org/2001/sw/wiki/Hyena>

¹⁷ <http://semanticweb.org/wiki/OntoStudio>

TABLE IV. ANALYSIS AND COMPARISON OF RDF STORAGE TOOLS AND APIs.

Tool/API Name	Availability	Development Platform	Platform Dependency	Operating System	Pluggable /non-pluggable	Databases	APIs	Query Language	Market value
Jena	Open Source	Java	Independent	Any operating system	Pluggable	MySQL, Oracle, SQL Server, PostgreSQL	Core API	SPARQL	Very High
Sesame	Open Source	Java, Python, PHP	Independent	Any operating system	Pluggable	MySQL, PostgreSQL	Sesame Sail API	SPARQL	High
Kowari	Open Source	Java	Independent	Any operating system	Pluggable	XA triple store	JRDF, Jena, SOAP	-----	Very High
Mulgara	Open Source	Java	Independent	Any operating system	Pluggable	-----	JRDF, Jena, SOAP	SPARQL, TQL	High
Virtuoso	Commercial & open source	Java	Independent	Any operating system	Pluggable	DB2, Oracle, MS SQL Server	-----	SPARQL	High
3Store+	Open source & commercial	C, PHP, Ruby, Python, Java	Dependent	Linux and MAC OS	Non Pluggable	MySQL, Berkeley DB	C API	RDQL, SPARQL	Low
YARS2	Open Source	Java	Independent	Any operating system	Non Pluggable	Berkeley DB	-----	SPARQL	Zero
BigData	Open Source	Java	Dependent	Linux	Pluggable	BigData RDF database	-----	SPARQL	Medium

C. RDF Browsing and Visualization Tools and APIs

An RDF model may contain millions of resources, interconnected with one another in a complex relationships confirming to triple format (subject, predicate and object). The complex nature of interconnection between resources makes it difficult for RDF users to interact directly with them in textual format such as RD/XML and N-Triple serialization formats. A

user can execute a query against RDF model to retrieve a specific and relevant resource but it would be more difficult for a user to navigate the RDF model for finding relevant data without visualizing the RDF model. Therefore, various RDF browsing/visualization tools are developed by Semantic Web programmers which provide easy to use interface and arrange all RDF resources in a much better way for navigation and manipulation. Table VI analyses and compares the RDF browsing and visualization tools and APIs using a set of parameters.

1) *RDF Model Browser*: RDF Model Browser is used for visualizing and representing RDF models in a hierarchical layout [16]. Unlike other tools visualizing RDF models in graphs, mapping RDF graphs in tree structure layout may result graphs, mapping RDF graphs in tree structure layout may result in appearing some nodes of a graph more than once in RDF Model Browser. The tool provides an interactive interface and different parts of a model can be visualized properly because a model can be viewed in a variety of ways in RDF Model Browser. The tool loads RDF model for visualization from a URL or from a local file stored on disk and enables visualizing several models concurrently. RDF Model Browser can be easily

deployed and integrated with other applications due to its implementation as Java Servlets 2.3 web application. It has been tested with Apache Tomcat and should be compatible with any other J2EE Servlet version 2.3 container. The tool is open source and available under the GNU General Public License (GPL) and platform independent.

2) *RDF Gravity*: RDF Gravity is a powerful tool for visualizing RDF and OWL resources [17]. It has been developed in Java, implemented using the JUNG Graph API and Jena semantic web toolkit and is freely available for download while requiring JVM1.3 or higher for installation

along with JUNG¹⁸ Graph API and Jena semantic web toolkit (Jena 2.0). RDF Gravity can filter local and global namespaces, visualize multiple RDF model from files, and makes full text search possible. Users can select nodes in a graph in numerous ways and can move the selected nodes to anywhere without changing the layout of a graph. Navigation between individual nodes is easy and a whole graph layout can be obtained by clicking a button available at the interface.

3) *OntoViz*: OntoViz is a highly sophisticated tool for visualizing RDF based Protégé ontologies [18]. The OntoViz tab is bundled with full installation of protégé. The tool has several features including visualization a part of an ontology by picking a set of classes, showing different slots and their edges in a colourful format, visualizing sub-classes and super-classes locally. The tool represents nodes and links in both hierarchical (tree) shape and a 2D graph view [19]. In contrast, Protégé OntoViz cannot provide

¹⁸ http://www.datalab.uci.edu/papers/JUNG_tech_report.html

keyword based searching facility, and the contraction and extension of nodes. OntoViz is an open source tool which can be installed on any operating system, however, the installation requires the prior complete installation of Protégé.

4) *Welkin*: Welkin is a general purpose browsing and visualization tool for RDF graphs. Originally, Welkin was developed by Stefano Mazzocchi and Paolo Ciccarese, working in the SIMILE project¹⁹. The tool is an open source developed in Java and published under BSD license. It is interactive browsing tool and can be installed on several platforms including, Linux, Mac and Windows. Welkin is specially designed for data analysts who want to test a potential mappings method between resources and ontologies and needs a RDF visualisers to quickly visualize a model to any type of errors such as spell mistakes in namespaces and URI's, or mistakes in the whole model etc. Some other features of Welkin includes support for Turtle/N3 RDF Serialization formats, applying colours to code resources, capability to change the link for individual predicates, and automatic selection and filtering of nodes on their graph notional properties.

5) *Longwell*: Longwell is an open source platform independent tool developed in Java as a web application and is license under BSD style license by the Simile project²⁰. However, its installation requires prior installation of Java 1.5 or later and Apache Maven 2.0 or later. A servlet container is required only if longwell is to be deployed on an existing servlet environment otherwise not needed. Longwell is one of the highly powerful tool for browsing RDF model containing large amount of data sets [29]. One of the important features of Longwell distinguishes it from other browsing tools is the faceted browsing UI paradigm. Due to this faceted feature, users can create user friendly and animated websites without knowing the detail of code and can browse, filter and visualize complex and random data sets. Longwell has support for serialization formats including RDF/XML, N3 and N-Triple.

V. EVALUATION OF TOPIC MAPS AND RDF DEVELOPMENT TOOLS AND APIS

Topic Maps and RDF standards of different organizations, ISO and W3C respectively, are regarded as competitors to win the race of Semantic Web. Topic Maps has its roots in traditional finding aids such as back-of-book indexes, glossaries, and thesauri. RDF has its roots in formal logic and mathematical graph theory. RDF is the knowledge representation applied to information management from the perspective of machines, whereas, Topic Maps from the perspective of humans. RDF is resource centric, whereas Topic Maps are subject centric. RDF is more "low-level" than Topic Maps. The subject of every assertion (or statement) in an RDF model is a resource, identified by a URI. The subject of every assertion in a topic map is a topic, representing a subject,

which may be addressable or non-addressable. Addressable subjects are identified by their URIs (as in RDF); non-addressable subjects are identified by the URIs of (one or more) subject indicators. The subject of every assertion (or statement) in an RDF model is a resource, identified by a URI. The subject of every assertion in a topic map is a topic, representing a subject, which may be addressable or non-addressable. Addressable subjects are identified by their URIs (as in RDF); non-addressable subjects are identified by the URIs of (one or more) subject indicators.

However, Topic maps and RDF have a number of similarities. They both attempt to alleviate the same general problem of infoglut by applying knowledge representation techniques to information management. They both define abstract models and interchange syntaxes based on XML and both have models that are simple and elegant at one level but extremely powerful at another: In topic maps, most things are topics (not just the topics themselves); in RDF, the value of a resource's property may itself be a resource, which in turn has properties of its own. The models of topic maps and RDF are sufficiently similar that it is possible to define generic mappings between the two in either direction. However, doing so does not yield useful results in terms of the target paradigm. An RDF triple can in theory be mapped to at least six different topic map constructs, but without knowledge of the semantics of the predicate, an optimal choice cannot be made. Likewise, topic characteristics can be mapped generically to RDF triples but without an RDF schema for topic maps the higher level of semantics are lost; and even with such a schema, the results are totally inadequate from the point of view of RDF processing. At the level of the schema, on the other hand, it is possible to describe two-way mappings that are extremely useful. Once the semantics of a RDF statement is known, it become easy to select which kind of topic map constructs to map it. Similarly, semantics that might otherwise be lost when mapping from topic maps to RDF can be expressed in an RDF schema. This suggests that the chances of unifying the two models in the short term are very slight. The immediate goal should rather be interoperability.

To proliferate and strengthen the technologies and turn the vision of Semantic Web into reality both of these technologies are provided with a strong support of software applications and APIs to serve in a varied ways such as to help users in understanding the technologies easily, to help programmers to speed up the development process and produce qualitative products. Most of the tools and API for both of the technologies are open source, provided to users for free usage, and platform independent (usable on varied operating systems) because of their development in cross-platform languages such as Java, and PHP etc.

¹⁹ <http://simile.mit.edu/welkin/>

²⁰ <http://simile.mit.edu/wiki/Longwell>

TABLE V. ANALYSIS AND COMPARISON OF RDF BROWSING/VISUALIZATION TOOLS AND APIS.

Tool/ API Name	Availability	Development Platform	Platform Independent	Operating Systems	Pluggable /Non- pluggable	User Interface	Visualiza- tion Method	APIs	Market value
RDF Model Browser	Open Source	Java Servlets 2.3	Independent	Any operating system	Pluggable	Web based Client server	Tree view	-----	High
RDF Gravity	Open Source	Java	Independent	Any operating system	Pluggable	GUI	Graph view	JUNG Graph API	High
OntoViz	Open Source	Java	Independent	Any operating system	---	Plug-in	Tree view	-----	Low
Welkin	Open Source	Java	Independent	Windows, MAC OS, Linux	Pluggable	GUI	Graph view	-----	High
Longwell	Open Source	Java	Independent	Any operating system	Pluggable	Web based client server	GUI	-----	High

TABLE VI. ANALYSIS AND COMPARISON OF RDF EDITING TOOLS AND APIS.

Tool/ API Name	Availability	Development Platform	Platform Dependency	Operating Systems	Pluggabl e /Non- pluggable	Architecture	APIs	Market value
Protégé	Open Source	Java	Independent	Windows, UNIX, Linux, and MAC OS	Pluggable	Stand alone	Core APL, OWL API	Very High
Altova	Commercial	Java	Dependent	Windows	Pluggable	Stand alone	OWL API	Medium
IsaViz	Open Source	Java	Independent	Any operating system	Pluggable	Stand alone	-----	Low
Hyena	Commercial	Java eclipse	Dependent	Windows	Pluggable	Stand alone and Web based interface	-----	Low
OntoStudio	Commercial	-----	Dependent	Windows	Pluggable	Stand alone	OWL API	High

Most of the tools provided are pluggable, providing opportunities to the users to create their own plug-ins for the tools to extend their technologies and participate and contribute to the overall development process. Most of the tools and APIs found interoperable in their respective technology by supporting importing and exporting and translation to and from many serialization language of many formats. However, the quality of translator would create a matter and there are no experimental results about the possibility of exchanging Topic Maps and RDF documents between different tools and knowledge on the loose in the translation process. The tools analysed are found varying in their knowledge representation point of view, while tools following either frame-based approach, first order logic based approach, or any hybrid approach. Moreover, tools also differentiate in their knowledge visualization methods, where tools are following either tree-based visualization, or graph-based visualization. Tools are found varying in their degree of inference services by having either their built-in inference engines or supporting external inference engines and query language for checking the constraints and consistency checking, automatic classification, and exception handling etc. Although, most of the tools found promoting their usability by having graphical user interfaces, but still they vary in the degree of services and features provided such as collaborative working, reusing of existing documents, interface and visualization customization, taxonomy viewing, pruning and zooming, and readily availability of help for end users. Most of the storage tools in both of the

technologies are found complaint to existing market oriented database standards (.e.g. MySQL, SQL Server, DB2, and Oracle etc.) which is not only useful for users to easy to understand and use but can be interchanged with other applications as well.

Although, the tools and API provided for both of these technologies are good enough by extending considerable help using their features to users to solve their problems and one can be found good in one situation while another in another situation. However, RDF has a leap over Topic Maps because of being successful in getting attention of most of the research and developers in the area. Therefore, the list of tools and APIs provided is much greater as compared to Topic Maps and still evolving, whereas most of the Topic Maps tools have been stopped working and newer version not released since years.

RDF tools and APIs are pretty much mature, providing enormous features to help users in solving their problems and have high market values. However, instead of dividing the potential and creating two separate islands, tools and APIs of both of the technologies can be regarded as complementary instead of competitors and to look for ways of realizing the potential synergies between the two. Ontopia has clearly demonstrated this potential through its use of RDF (under the covers, as it were) in the automated generation of topic maps. The tools and APIs can be increased to define generic mapping between the two in either direction. However, doing so does not yield useful

results in terms of the target paradigm but the issues can be resolved and problems can be fixed. At the schema level, it is possible to describe an extremely useful two-way mappings paradigm. Once the semantics of a RDF statement is known, it becomes easy to select that which kind of topic map constructs to map it. Similarly, semantics that might otherwise be lost when mapping from topic maps to RDF can be expressed in an RDF schema. However, the chances of unifying the two models in the short term are very slight. Therefore, the immediate goal should rather be interoperability. The notion of published subject indicators (PSIs) is based on URIs, which is general enough to solve the interoperability problem for both Topic Maps and RDF and make it easier to exploit the synergies between the two.

VI. CONCLUSION

The World Wide Web is a collaborative and free medium containing billions of pages scattered over the Internet in different forms and shapes. However, this tremendous Web growth comes at the cost of losing rich semantics, necessitating human presence for information interpretation and retrieval. Semantic Web is deemed as solution to the problem. To make Semantic Web practical, two technologies (i.e. Topic Maps and RDF) are invented focusing on associating metadata with web resources to make them humans as well as machine understandable. For users and developers to fully exploit the features and benefits of Topic Maps and RDF, strong software support is essential. Software tools and APIs are essential for users and developers to decrease cost and time, and increase performance and conformance in the software development process. To successfully implement the Semantic Web technologies and improve their performance, a number of software tools and APIs are developed by the academia, researchers, and organizations to maintain, store, edit and design, and visualize Topic Maps and RDF. These tools have different capability, scalability, and applicability standards and can be taken as one good in one situation while another in another situation.

In this paper, we have presented a comprehensive overview of RDF and Topic Maps tools and APIs covering the storage, editing, and browsing/visualization areas. The on hand knowledge about the available tools is presented wisely to elaborate their capabilities and are compared and analysed in a tabular format to give insight knowledge to the users to select one suitable for their needs. A number of commercial and open source tools and APIs have been investigated and categorized according to their capabilities of storage, editing, and browsing/visualization. During investigation, it has been observed that tools are rich enough to fulfil the varied needs of their respective technology. However, these tools are found fruitful in their respective domains but only a handful of tools are found quite capable of addressing all of the domains in their respective technology (e.g. Portege in RDF etc). Similarly, none of the tools is found capable of storing, editing, and browsing/visualizing both RDF and Topic Maps. By comparison, the tools support for RDF is found higher as compared to Topic Maps. RDF tools are highly featured and more mature as compared to Topic Maps tools. But, instead of regarding them as competitors, the existing

commonalities in both of the technologies can be exploited effectively to make them interoperable. Tools such as protégé, Ontopia Jena, IsaViz, and TM4J etc can be extended to support both RDF and Topic Maps. To provide ease and advancements, universal tools capable of storing, editing, and browsing/visualizing both RDF and Topic Maps data are needed to be investigated.

REFERENCES

- [1] K. Ahmad, "Topic Maps - A Practical Introduction With Case Studies", in *Proc. of. Proceedings of XML Europe 2002*, Barcelona, Spain, 2002.
- [2] K. Kritikos, and M. Stratakis, "Products and Services Company Ontopia", Work Students at HY-566 Knowledge Management in Training Networks and the Internet, 2003.
- [3] J. Schmidt, and A. Stephan, "QuaaxTM", 2011, Available: <http://quaaxtm.sourceforge.net/>, [Access Date: 05/04/2013].
- [4] T. J. Project, "TM4J Topic Map Engine", Available: <http://tm4j.org/tm4j-engine.html>, [Access Date: 25/03/2013].
- [5] S. Krosse, "MaJorToM", 2011, Available: <http://code.google.com/p/majortom/wiki/Intro>, [Access Date: 25/04/2013].
- [6] K. Ahmed, and J. Park, "Techquila's Topic Map World Topic Map", 2003, Available: <http://www.techquila.com/topicmaps/tmworld/11812.html>, [Access Date: 02/04/2013].
- [7] D. Dicheva, and C. Dichev, "TM4L: Creating and browsing educational topic maps", *British Journal of Educational Technology*, vol. 37, no. 3, 2006, pp. 391-404.
- [8] H. Niederhausen, "Onotoa Features", 2011, Available: <http://code.google.com/a/eclipselabs.org/p/onotoa/wiki/features>, [Access Date: 08/04/2013].
- [9] T. Witt, "Conception and Implementation of a Visual Editor for Topic Maps," Diploma Thesis, Department of Computer Science, Teaching and Research Unit of Media Computer Science, Ludwig-Maximilians-University Munich, 2009.
- [10] B. Grand, and M. Soto, "Topic Maps Visualization," in *Visualizing the Semantic Web*, V. Geroimenko and C. Chen, Eds., 2003, pp. 49-62.
- [11] A. Hatzigaidas, A. Papastergiou, G. Tryfonand, and A. Manitsaris, "A Topic Map Editor and Navigation Tool," *International Conference on WWW and Internet 2004 (ICWI 2004)*, Madrid, Spain, pp. 359-365, 2004.
- [12] A. Hertel, J. Broekstra, and H. Stuckenschmidt, "RDF Storage and Retrieval Systems," in *Handbook on Ontologies*, S. Staab and R. Studer, Eds., 2009, pp. 489-508.
- [13] K. Wilkinson, C. Sayers, H. Kuno, and D. Reynolds, "Efficient RDF Storage and Retrieval in Jena2," *Proc. First International Workshop on Semantic Web and Databases*, 2003.
- [14] B. Haslhofer, E. Momeni Roochi, B. Schandl, and S. Zander, "Europeana RDF Store Report," University of Vienna, Vienna, Technical Report 2011.
- [15] E. Pietriga, "IsaViz User Manual", 2005, Available: <http://www.w3.org/2001/11/IsaViz/usermanual.html>, [Access Date: 20/05/2013].
- [16] D. Skvortsov, "RDF Model Browser", Available: <http://visualrdf.sourceforge.net/index.html>, [Access Date: 29/05/2013].
- [17] S. Goyal, and R. Westenthaler, "RDF Gravity (RDF Graph Visualization Tool)", Salzburg Research, Austria, Available: http://semweb.salzburgresearch.at/apps/rdf-gravity/user_doc.html, [Access Date: 30/05/2013].
- [18] M. Sintek, "OntoViz", 2010, Available: <http://protege.wiki.stanford.edu/wiki/OntoViz>, [Access Date: 10/06/2013].
- [19] R. Sivakumar, and P. V. Arivoli, "Ontology Visualization Protege Tools - A Review," *International Journal of Advanced Information Technology (IJAIT)*, vol. 1, no. 4, 2011.
- [20] S. Kushro, and A. M. Tjoa, "Fulfilling the Needs of a Metadata Creator and Analyst- An Investigation of RDF Browsing and Visualization Tools," in *Canadian Semantic Web*, vol. 2, M. Koné and D. Lemire, Eds., 2006, pp. 81-101.