Proceedings of the Pakistan Academy of Sciences 52 (1): 1–14 (2015)

ISSN: 0377 - 2969 (print), 2306 - 1448 (online)

Pakistan Academy of Sciences

Research Article

Resource Description Framework and Topic Maps: Complementary or Competitive?

Fakhre Alam[1]*, Muhammad Abid Khan[2], Sami ur Rahman[1], Shah Khusro[2], and Shaukat Ali[2]

[1]Department of Computer Science & IT, University of Malakand, Lower Dir, Pakistan
[2]Department of Computer Science, University of Peshawar, Peshawar, Pakistan

Abstract: Resource Description Framework (RDF) and Topic Maps (TMs) are the two prominent technologies, envisioned for realizing Semantic Web. RDF and Topic Maps are independent technologies developed by separate organizations, both providing mechanisms for enriching web contents with metadata for representing semantic relationships among them. This, ultimately, makes the Web more useful and 'understandable' both for humans and machines and improves knowledge sharing and integration across different domains. Both of these standards have established their respective user-communities. Because of using different semantic representation mechanisms, this results in the problem of interoperability potentially dividing the Semantic Web into two separate islands. This study aimed at investigating the two technologies and to discover that how their interoperability problem can be solved by enabling them to work together. To achieve this objective, a comprehensive literature review of the two technologies was performed and is reported here to describe the architectures, serialization formats, tools & APIs, query languages, and real-world applications of both the technologies. For testing purposes, a Book Ontology was developed in both RDF and TMs standards by employing their respective ontology development tools and serialization formats. The ontology was imported in their respective applications and results were evaluated which justify that the interoperability between RDF and TMs is possible, enabling them to work in a complementary fashion.

Keywords: Semantic web, Resource Description Framework (RDF), Topic Maps (TMs), serialization, Tools and Application Programming Interfaces (APIs), ontology

1. INTRODUCTION

The World Wide Web (www) at present contains billions of web pages interlinked using hypertext [1]. Finding, accessing, using and adding any content to the Web is very easy due to its simplest design. However, this simplicity comes at the cost of losing rich semantics and necessitates human presence for web content interpretation. In the current web with billions of web pages, finding specific information precisely is almost impossible and this problem aggravates more as size of the Web increases. Semantic Web is deemed as solution to the problem,tossed by the original creator of the Web. Semantic Web emphasizes on attaching metadata with web resources/contents for making them machine-processable. Semantic Web is an extension of the current Web, insisting the creation of implicit and meaningful relationships among the Web resource in a manner to be directly understandable by the machines [2]. Semantic Web will establish an environment where people and machines will function in cooperation and will make the availability of right information at the right time and the right place possible.

Resource Description Framework (RDF) and Topic Maps (TMs) are the two prominent standards developed by W3C and ISO respectively for realizing the vision of Semantic Web. RDF and TMs serve as the backbone for the Semantic Web and have gained high level of popularity [3]. Although both RDF and TMs share the same goal, but differ in their architectures due to being developed by

Received, May 2014; Accepted, January 2015
* Corresponding author: Fakhre Alam; Email: fakrealam@uom.edu.pk

different organizations. Architectural heterogeneity emerges the problem of interoperability between RDF and TMs, which could potentially divide web of the future into two disjoint islands. Researchers around the globe contributed to solve the interoperability problem by devising several serialization formats, tools and APIs, query languages, and applications etc. but they are focusing on narrow domains. The problem of broad interoperability encompassing all aspects of these technologies, however, remains uninvestigated.

This paper presents, a comprehensive literature study of both RDF and TMs technologies and articulates comparisons of their architectures, serialization formats, tools and APIs, query languages, applications, and models .A thorough investigation of all aspects of both of the technologies is performed for determining the directions in which they are evolving and to find their specific domains of applications. It also attempts to find out how far RDF and TMs can be used together beyond providing support for import/export of serialization formats. An organized procedure has been employed to analyze and categorize RDF and TMs using their underlying protocols structures and determine similarities and differences between them. To practically test the concept, book ontology is developed in both RDF and TMs standards using their relevant ontology development tools and serialization formats. The ontology is imported and evaluated in applications. Comparing RDF and TMs and covering all their aspects comprehensively in single document is never done before. Therefore, this paper will not only provide a jump start in the field for new comers but will also help researchers in finding new research dimensions as well as finding solutions to the existing ones.

2. WORLD WIDE WEB (WWW)

WWW has changed dynamically changed living style of the people by providing free and easy access to rich sources of information in the form of text, images, audios, animations, and videos. Simplicity and easy to use nature of the WWW releases users from the constraint of learning sophisticated computer programs to use[4]. Users can create and contribute almost any type of information to the WWW by just clicking a few objects and the information will be broadly available in no time. Despite all these improvements, the current web is still faces with a number of limitations. The main reason behind these limitations is the synthetic nature of the Web, providing no mechanisms for defining semantic contents and semantic relationship between the Web contents[5]. The free nature of the Web boosted the exponentially growth of the Web, overwhelming users with tons of information making retrieval of precise information very much difficult and time consuming. Thus, necessitates improving the current syntactic web into a more meaningful web, which will enable accessing of information fast, specific and with reasoning capability by the machine.

3. SEMANTIC WEB

The increasing amount of information on the WWW instigated the problem of Infoglut. The information overload problem has made the retrieving of web resources, extracting relevant data from web resources, and aggregating information from diverse sources for accomplishing a particular goal significantly difficult, error prone and time consuming. It is obvious that to find precise information of one's own interest needs excessive efforts, therefore, Semantic Web is envisioned as web of the future, which would be more powerful, collaborative, and enable web information to be processed equally by the machine and human beings[2]. The main purpose of Semantic Web is to enhance the current web in such a way that the problem of finding precise information at the right time becomes possible. Semantic Web elevates the information overload problem by improving the synthetic web using advanced techniques of knowledge representation, ontologies, computational linguistics, intelligent agents, and machine based searching.

4. SEMANTIC WEB TECHNOLOGIES

To implement the vision of Semantic Web and make it a useful medium for both humans and machines, the need of technologies and standards arises to effectively understand web contents and traditional tools. With the development of Semantic Web technologies, such as RDF and TMs it would become possible to make the Web useful for humans and understandable for machine by facilitating knowledge integration and sharing.

4.1 Resource Description Framework (RDF)

The main purpose behind the development of RDF was to provide infrastructure for the Semantic Web. RDF provides constructs for enabling web resources to be processed automatically by machine, and interoperability between diverse web applications etc. by implementing metadata relationship between web resources [7]. Metadata implementation enables search engines optimization for finding web resources quickly and easily and replacing traditional software agents with sophisticated intelligent software agents. RDF standard has several features and characteristics which are describe in the following sub-sections.

4.1.1. Triples (Subject, Predicate, Object)

RDF model is composed of statements where each statement represents metadata relationship in the form of triples (Subject, Predicate, and Object). A subject could be the URI of an entity whereas object could be the URI of an entity or a literal. Predicate represents the semantic relationship between a subject and an object in a RDF statement. A RDF model graphically represented is called RDF graph where triples are symbolized in the form of ellipse and arrow symbols[6]. In a RDF graph, ellipse is used for representing subject and object and arrow is used to represent predicate of a RDF statement. Relationship in a triple between subject and object is binary relationship, which always goes from subject to object. In RDF graph, subjects and objects can be shared and one subject can be the object of another subject.

4.1.2. XML-based Serialization

A RDF model can be serialized into XML based syntax, called RDM/XML, which exploits the potential of XML and to upgrade its syntax to a higher level for creating well-formed XML documents. RDF/XML can easily merge and interchange information from multiple sources and express web resources in a more meaningful way.

4.1.3. Uniform Resource Identifier (URI)

Everything on the Web, whether it is web address, a literal, or a blank node, must have a unique URI [8]. Although on the traditional web URI means URL, which is primarily used to find and access web document through web browsers, however, in RDF this term is used to find a unique and specific resource on the Web. In other words, one can say that anything on the Web must have a URI.

4.1.4. Reification

Assertion about things can be easily done with the help of RDF property called reification. It is a process in which we refer one statement to another one[9].

4.1.5. RDF Schema and Web Ontology Language (OWL)

To describe semantic relationships between resources, a schema language for RDF called RDF schema (RDFs) is used which can be expressed in a RDF model itself [10]. This framework further extends the original RDF model with some special semantic mechanisms to add classes of resources and the properties specific to these resources. RDFs uses the concept of inference, through information can be deduced using the existing information. RDFs provides excessive constructs for expressing classes, sub-classes, class properties, instances, and constraints restrictions for properties in the form of domains and ranges. However, the power of RDF and RDFs failed when it is applied to the solving of complex problems such as semantic annotations between different types of contents on the current web [11]. These problems can be solved with the help of ontology, which gives formal meaning to the web contents which are further interpreted and transferred into the semantic annotation. Ontology is used for effective reasoning, better syntax specification, representation of knowledge precisely, and the manipulation of knowledge from shared vocabularies.

To solve the problems of complex annotations among web resources, a more advanced and expressive language called OWL (Web Ontology

Language) was developed. OWL is W3C standard and offers better semantic integration and interoperability between web resources, as compared to RDFs. OWL is used to create ontologies for the Web and works closer to machines. It can also be used for the validity and consistency of implicit knowledge and making it explicit to the users [12]. The problems and weaknesses of RDFs can be controlled by OWL, by adding more vocabularies for describing classes and properties including disjointness properties, equality properties, symmetric and transitive properties, and restriction on classes and properties.

4.2. Topic Maps (TMs)

Topic Maps is another technology for the Semantic Web which can express and exchange knowledge in a meaningful way [13]. With TMs, relevant information on the Web can be manipulated quickly and easily. Anything on the Web can be expressed by the TMs in the form of topics, associations, and occurrences [18]. The features and characteristics of TMs standard is described in the following sub-sections.

4.2.1 Topic

A resource within any domain is called a topic. Topic represents some real-world subjects in concrete and simple form [14]. Examples of such subjects could be a book name (.e.g.,The Pathan etc.), and an author (.e.g., Ghani Khanetc.). Topic can be created by a method called reification. A subject becomes effective when topic for it is reified, which means that the subject now comes under the discussion of Topic Maps paradigm. Therefore, subject in TMs paradigm is anything such as book, city, region, country, person, author, motive, branch of study, and unit of an academic institution. Each subject may have some existence, explicit characteristics, and generally discussed by human beings. In particular, subject is the main focus of discussion and conversation of TMs authors. Subject becomes valid for machine processing due to topic.

4.2.2. Association

In Topic Maps paradigm, a topic is linked to other topics through relationships called Associations[15]. In topic type, topics are grouped according to their type. Similarly, associations in a single class can also be grouped called association type that may or may not be pointed out openly. Each individual association (i.e., indicated explicitly) is derived from a particular association type. For example in statements “Charsada is in Pakistan” and “Ghani Khan was born in Pakistan”, there are two association types “born in” and “is in”.

4.2.3. Occurrences

In index, at the back of a book, page numbers represent links or references to one or more pages for a particular topic, providing useful information for the reader. Similarly, when someone wants to retrieve relevant information about a subject on the Web, he/she can do it by means of the topic characteristics called occurrences [16]. The relationships between subjects and information resources can be represented by occurrences, and the information provided by the occurrences is relevant to a given subject. In TMs occurrences, each topic has some relevance to one or more information resources regardless of their scattered locations [17].

4.2.4. Scope

To qualify and convey statements about TMs, scope can be attached to anything like name, association and occurrence. To assign characteristics to the topic, to check the degree of validity and the circumstances under which topics are linked through association is stated by the scope. On the Web there are lots of names with the same meaning which are used for the same subject, therefore by using these scopes one can differentiate between them. If there is no scope, the default scope is single scope in which all subjects are merged. In TMs paradigm, a constraint is imposed by the scope called naming constraint in which the namespaces are created for the base names of topics[14]. Topic naming constraints say that any topic having the same base name in the same scope should be merged together.

4.3. Comparison of RDF and Topic Maps

Table 1 shows the comparison of RDF and TMs. In the Table, comparison is based on standardization,

Table 1. Analysis and comparison of RDF and Topic Maps.

Technology	Standardization	Representation	URI	Serialization	Tools & APIs	Query Language	Applications	Constraints Language	Assertion	Reification	Scope
RDF	W3C	Resource, Properties Statement	Uses URI s to find web resources.	RDF/XML turtle, N3, TriX, N-Triple.	Jena, Sesame, Kowari, Mulgara, YARS2, Bigdata, Protege, RDF Model ..	SPARQL, RDQL, RQL, SeRQL, XsRQ..	Redland, RDF Gateway, RDF Mapper etc	RDF Schema and OWL	Yes	Difficult	No built- in mechanism
Topic Maps	ISO	Topic, Associations Occurrences.	Uses URIs and Scopes to find web resources.	XTM, CTM, CXTM, GTM, HyTM, LTM.	Ontopia, QuaaxTM, TM4J, MajorTom, TMAPI, Topinics, Onotoa, Ontopoly..	TMQL, tolog, TMRQL, AsTMa, Toma	Wandora	TMCL	Yes	Easy	Built-in mechanism

information representation, resources finding techniques, serialization, tools and APIs, query languages, applications, constraints languages used by standards, assertion capability, reification, scope, and their other fundamental features. It is obvious from the literature study that reification is automatically performed in TMs, whereas, in RDF it requires explicit declaration each time. Moreover, RDF has no built- in mechanism to attach scope to resources, while TMs can easily attach scope to resources. Therefore, research efforts are required to either implement TMs reification mechanism and scoping property in RDF or new reification and scope techniques are required to be developed which should be acceptable to the both worlds.

5. RDF AND TOPIC MAPS DEVELOPMENT TECHNIQUES

Originally TMs were developed only for the purposes of representation, merging and processing of indexing on the back of a book [19]. Latter on this technology, like its counterpart RDF, was used in finding resources on the Web and in establishing metadata relationship between web resources. After being RDF and TMs standardized by W3C and ISO respectively, a comprehensive work start on them in parallel to fulfill the idea of Semantic Web. Different types of techniques such as serialization formats, tools and APIs, languages (.i.e., query, constraints, schema), and applications were developed for RDF and Topic Maps, which are described briefly in the following sub-sections.

5.1. RDF and Topic Maps Serialization Formats

The process of serialization converts data semantically in one format into other format for storage and transmission purpose[3]. It is due to the serialization that data in one computer environment can be reused later in another computer environment, without changing in the original data format [20]. To interchange data for storage and transferring on the Web persistently in human readable format, several technologies were developed. XML is the one which is mostly used due to its simplest nature and no need to go into the programming detail. Therefore, RDF and TMs also make use of different types of XML and non-XML interchange syntaxes for serialization of data.Knowledge representation and navigation can also be done with simplicity and flexibility with serialization due to its support for different software, and implementation on multiple platforms. RDF and TMs serialization formats convert data semantically into a format which is suitable for storage and transmission over the WWW[21]. After the conversion the data can also be easily reused in another computer environment, without affecting the original data format. The popular serialization formats for TMs are XTM, LTM, CTM, CXTM and HyTime. Similarly, several types of serialization formats based on XML and non-XML are also available for RDF including RDF/XML, TriX, N3, N-Triple and Turtle.

5.2. RDF and Topic Maps Tools and APIs

RDF and Top Maps Tools are developed to implement RDF and TMs technologies effectively without any crash, to improve their performance and to easily access the data from different software tools and APIs. It is due to RDF and TMs tools and APIs that applications are created for numerous purposes including analysing performance of a system, quick and intelligent information retrieval, integration and linking of different RDF and TMs components, reporting system errors, and effective browsing and visualization.

Several types of commercial and open source tools and APIsare developed for RDF and TMs to achieve the vision of Semantic Web. The available RDF and TMs tools and APIs can be divided into three categories: Engines, Navigators, and Editors [22]. Programmers can easily construct, change, import, export, and access RDF and TMs documents with the help of Engines along with comprehensive APIs. User can also browse and navigate RDF and TMs very easily in a human readable form using navigation tools. Moreover, RDF and TMs Editors provide an environment for the users to create and modify models according to their requirements. RDF and TMs APIs provide interfaces for accessing and integration of web available tools and applications [23]. Using APIs, tools and applications can interact with each other without any explicit need of human interference or knowledge. APIs enables quick development of applications by integrating the functionalities from the existing ones.

5.3. RDF and Topic Maps Query Languages

Effective query languages having easy and user friendly interfaces are needed to handle large and complex systems [24]. RDF and TMs technologies have their respective lists of query languages having logic and inference capabilities along with basic SQL constructs, can easily and effectively retrieve required information from the underlying complex architectures. TMs query languages such as TMQL, tolog, TMRQL, AsTMA, and Toma can retrieve topics, associations between topics, and their occurrences effectively. On the other hand, RDF community has investigated a list of query languages including RDQL, RQL, SeRQL, XsRQL and the latest standard SPARQL. These query languages have the capability to effectively manipulate RDF metadata information available on different platforms.

5.4. RDF and Topic Maps Applications

Applications are needed to embed the semantic structure in the current web. Applications are also used to integrate data and improve search mechanism to more specialized and intelligent levels. RDF and TMs applications can provide adaptive and customized views by analysing users' current tasks/activities, and accordingly gives responses to the users in a particular context. Several types of powerful applications are available in the market for RDF and TMs authoring, accessing, viewing, visualizing, and merging information including Wandora for TMs, and Redland, RDF Gateway, and RDF Mapper for RDF.

4.5. Comparison of RDF and Topic Maps Development Techniques

A detail analysis and comparison of the main features and techniques developed for RDF and TMs technologies are presented in Table 2 Table 3 respectively. It is obvious from the tables that every technique have their own capability to create, describe, manipulate, facilitate, maintain, and implement RDF and TMs documents. In the tables, several serialization formats developed for RDF and TMs are analysed and categorized according using the criterions of XML and non-XML basis, their expressibility/computability, interface for the users, compactness, tools and APIs support, availability,and applicability. From the Table 2, we came up with a conclusion, that the most appropriate serialization format to represent RDF graphs is Notation-3 (N3). Similarly from Table 3, it is find out that the most appropriate serialization format to represent TMs is XTM. The main reasons due to which Notation-3 and XTM have superiority over its companion are their logic and inference mechanisms, readable syntax which is also compact and persistent, advanced parsing mechanisms, automatic translation facilities, availability of open source visual and text based editors, and expressiveness to represent any type of RDF and TMs relationships and constraints.

The tools and APIs for RDF and TMs are also analysed and evaluated according to set of criteria (Table 2, 3). These criterions include availability

Table 2. Detail analysis and evaluation of the techniques developed for RDF.

Resource Description Framework (RDF)							
Serialization Formats		**Tools and APIs**		**Query Languages**		**Applications**	
RDF/ XML	XML Based Vey High Expressibility/ Computability Simple and User Friendly Less Compact Tools Support is Available Applicable for Both Small & Large Scale	**Jena**	Open Source , Very High Market Value, Development Platform: Java Platform Independent and Pluggable Architecture Database: MySQL, Oracle, SQL Server, PostGreSQL APIs: Core API, Query Language: SPARQL	**SPARQL**	Organization/Project under which developed: W3C RDF Data Access Working Group (DAWG) Implemented Language having Very High Query Results and Accuracy Scalable & Extendable No support for negative statements and path expression	**Redland**	Open Source Availability Development Platform is C Language Tools/ APIs are Available High Market Value Killer Application: No
		Sesame	Open Source , High Market Value, Development Platform :Java, Python and PHP Platform Independent and Pluggable Architecture Database: MySQL, PostGreSQL APIs: Sesame Sail API, Query Language: SPARQL				
TriX	XML Based High Expressibility/ Computability Simple Syntax Compact Tools Support is Available Applicable to Small Scale only	**Kowari**	Open Source , Very High Market Value, Development Platform: Java Platform Independent and Pluggable Architecture Database: XA Triple Store APIs: JRDF, Jena, SOAP, Query Language: ??			**RDF Gateway**	Commercial Based Availability Development Platform : Java Language Tools/ APIs are Available Market Value Unknown Killer Application: No
		Mulgara	Open Source , High Market Value, Development Platform : Java Platform Independent and Pluggable Architecture Database: ??. APIs: JRDF, Jena, SOAP Query Language: SPARQL, TQL	**RDQL**	Organization/ Project under which developed: Hewlett Packard Implemented Language having High Query Results and Accuracy Scalable & Extendable Data-oriented and Declarative Performance issues, Restriction of OR operation		
N3	Non XML Based Medium Expressibility/ Computability Easy Parsing Highly Compact Tools Support is Available Applicable for Both Small & Large Scale	**Virtuoso**	Commercial & Open Source , High Market Value, Development Platform: Java Platform Independent and Pluggable Architecture Database: DB2, Oracle, MS SQL Server APIs: -----. Query Language: SPARQL			**RDF Mapper**	Open Source Availability Development Platform: Ruby Language Tools/ APIs are Available Market Value is Low Killer Application: No
		3Store+	Open Source & Commercial, Low Market Value, Development Platform: C, Ruby, Java, Python, PHP Platform Dependent and Non Pluggable Architecture Database: MySQL, Berkeley DB, APIs: C API, Query Language: RDQL, SPARQL	**SeRQL**	Organization/ Project under which developed: Sesame Implemented Language Low Query Results and Accuracy Scalable & Extendable Can Easily Parse Can Implement Boolean Constraints Perform Operations on set theories & Nested Queries		
		YARS2	Open Source , Zero Market Value, Development Platform : Java Platform Independent and Non Pluggable Architecture Database: Berkeley DB, APIs: -----, Query Language: SPARQL				
		Bigdata	Open Source , Medium Market Value, Development Platform: Java Platform Dependent and Pluggable Architecture Database: BigData RDF Database APIs: -----, Query Language: SPARQL				

Table 2 Contd...

Table 2 (Contd)

Resource Description Framework (RDF)							
Serialization Formats		**Tools and APIs**		**Query Languages**		**Applications**	
Turtle	Non XML Based High Expressibility / Computability Very Simple Syntax Highly Compact Tools Support is Available Applicable for Large Scale	**RDF Gravity**	Open Source , High Market Value, Development Platform: Java Platform Independent and Pluggable Architecture APIs: JUNG Graph API. User Interface: GUI, Visualization Method: Graph View	**SquishQL**	Organization under which developed: RDF Data Access Working Group Not Implemented Yet Medium Query Results and Accuracy Not Scalable & Extendable Simple and Stylish Interface Cannot Support Closure Operation	**FOAF**	Open Source Availability Development Platform : Java Language Tools/ APIS are Available Market Value is Low Killer Application: Not
		Protégé	Open Source , Very High Market Value, Development Platform: Java Platform Independent and Pluggable Architecture APIs: Core API & OWL API, Architecture: Stand Alone				
N-Triple	Non XML Based High Expressibility/ Computability Simple & Easier to Read and Write Less Compact Tools Support is Available Applicable to Large Scale only	**Altova**	Commercial , Medium Market Value, Development Platform: Java Platform Dependent and Pluggable Architecture APIs: OWL API, Architecture: Stand Alone	**RQL**	Organization/ Project under which developed: ICS-FORTH Research Project Implemented Language Medium Query Results and Accuracy Scalable & Extendable Support for both Data and Schema	**SKOS**	Open Source Availability Development Platform is Java Language Tools/ APIS are Available Market Value is High Killer Application: Not
		OntoViz	Open Source , Low Market Value, Development Platform: Java Platform Independent. User Interface: Client , Visualization Method: Tree View				
		IsaViz	Open Source , Low Market Value, Development Platform: Java Platform Independent and Pluggable Architecture APIs: ---, Architecture: Stand Alone				
		Hyena	Commercial , Low Market Value, Development Platform: Java Eclipse Platform Dependent and Pluggable Architecture APIs: OWL API, Architecture: Stand Alone & Web Based Interface				
		Longwell	Open Source , High Market Value, Development Platform: Java Platform Dependent and Pluggable Architecture APIs: ---. User Interface: Web based Client Server Visualization Method: GUI				
		OntoStodio	Commercial , High Market Value Platform Dependent and Pluggable Architecture APIs: OWL API, Architecture: Stand Alone				
		RDF Model Browser	Open Source , High Market Value Development Platform is Java Platform Independent and Pluggable Architecture APIs: ---. User Interface: Web based Client Server Visualization Method: Tree View				

Table 2 Contd...

Table 2 (Contd)

Topic Maps (TMs)							
Serialization Formats		**Tools and APIs**		**Query Languages**		**Applications**	
XTM	XML Based Vey High Expressibility Simple Syntax Low Compact Tools Available Applicable for Both Small & Large Scale	**Ontopia**	Open Source , Very High Market Value, Development Platform: Java Platform Independent and Pluggable Architecture Database: DB2. APIs: Core Java API Query Language: Tolog	**TMQL**	Organization/ Project under which developed: ISO/IEC JTC1 SC34 WG3 Implemented Language High Query Results and Accuracy Scalable & Extendable Applicable to huge quantity, continuously varying information and for semi structure environment Too Hard to understand and Implement	**Wandora**	Open Source Availability Development Platform is Java Language Flexible and Pluggable Architecture Browse and Visualize both TM Data & Graph Better Locking Mechanism Import and Export TM data in several Interchange formats such as XTM and LTM Tools/ APIS are Available Market Value is High Killer Application: Not
		QuaaxTM	Open Source , Low Market Value, Development Platform: PHP Platform Independent and Pluggable Architecture Database: MySQL, InnoDB. APIs: PHPTM API, Query Language: ...				
CXT-M	XML Based High Expressibility / Computability Complex Syntax Less Compact Tools Available Applicable for Small Scale Only	**TM4J**	Open Source , High Market Value, Development Platform: Java Platform Independent and Pluggable Architecture Database: RDBMS and ORDBMS. APIs: TMAPI, Query Language: Tolog	**Tolog**	Organization/ Project under which developed: Ontopia Implemented Language Medium Query Results and Accuracy Scalable & Extendable Contains Logic and Inference Capability Cannot Manipulate Every Type of resource from TM		
		MaJorToM	Open Source , Medium Market Value, Development Platform: Java Platform Independent and Pluggable Architecture Database: MySQL, PostGreSQL. APIs: TMAPI Version 2.0. Query Language:------				
Hy-Time	Non XML Based Medium Expressibility Simple Syntax Not Compact Tools Available Applicable for any Type of Hypermedia	**Goose-Works**	Open Source , High Market Value, Development Platform: Python and C Platform Independent and Pluggable Architecture Database: RDBMS with slight variation. APIs: Core API Query Language: L1, and STMQL				
		Nexist	Open Source , Very High Market Value, Development Platform:Java Platform Independent and Non Pluggable Architecture Database: RDBMS. APIs: Core API based on XTM Query Language: Tolog	**TMRQL**	Organization/ Project under which developed: Networked Planet Non Implemented Language High Query Results and Accuracy Provides easy accessibility Difficult to implement due to its complex nature and inconsistencies in SQL support		
CTM	Non XML Based Very High Expressibility Simple Syntax Highly Compact Tools Available Applicable for Both Small Scale	**TM4L**	Open Source , High Market Value, Development Platform: Java Platform Independent and Pluggable and Stand Alone Architecture APIs: TMAPI				
		Topincs	Open Source , Low Market Value, Development Platform: Apache, MySQL and PHP Platform Independent, Pluggable and Web Based Architecture APIs: PHP API				

(i.e., open source or commercial), development platforms, pluggable/ non pluggable, databases and query languages, APIs support, basic architecture, user interface, visualization methods, and market value. From the analysis and evaluation it is obvious that most of the tools and APIs for RDF and TMs are open sources, platform independent because of their development in cross-platform languages such as Java, and PHP. The available tools are mostly pluggable due to which users can extend their functionalities. However, it is obvious from the experimental results that the problem of losing semantics and accuracy of information occurs duringtranslating/interchanging RDF and TMs documents which needto be solved. It is also deduced from the detail study and comparison that RDF has a leap over TMs in the race of tools and APIs. The main reasons of RDF success in getting attention of most of the Semantic Web researchers and developers includesits big list of tools which are still evolving, providing enormous features for the users for solving their complex problems, and high market value. Therefore one can strongly say that RDF and TMs tools and APIs are complementary but require discovering of methods for realizing the potential synergies between the two.

RDF and TMs query languages are also analysed (Table 2, 3) based on the parameters of development organization/ projects, implementation, results accuracy, scalability, and extendibility. It is obvious from the tables that almost every type of query constructs is similar to SQL along with logic and inference capabilities. In these query languages SPARQL, which was originally developed for RDF was further enhanced and tested for the manipulation of TMs data in terms of RDF schema. This was a successful test due to its sophisticated architecture called TM-viewer architecture and triple mapping technique called TMSPARQL. TMSPARQL takes SPARQL query and translates it into a set of matches against a TMs data store.

The available applications for RDF and TMs are also analysed (Table 3, 4). Most of these applications are open source, have high market value, developed in Java, several types of tools/ APIs are available for each application but there is no killer application developed yet for each of

Table 3. Detailed analysis and evaluation of the techniques developed for Topic Maps.

Topic Maps (TMs)							
Serialization Formats		**Tools and APIs**		**Query Languages**		**Applications**	
GTM	Non XML Based High Expressibility Simple Syntax Highly Compact Tools Not Available Applicable for Both Small Scale	**Onotoa**	Open Source , High Market Value, Development Platform: Java Platform Independent, Pluggable and Stand Alone Architecture APIs: -------	**AsTMa**	Organization/ Project under which developed: Topic Maps Lab Implemented Language Low Query Results and Accuracy Scalable & Extendable Easy and simple to operate Lazy evaluation is not define yet		
		Ontopoly	Open Source , High Market Value, Development Platform: Java Platform Independent and Pluggable and Web Based Architecture APIs: ------				
LTM	Non XML Based Low Expressibility Simple Syntax Highly Compact Tools Available Small Scale	**TMAPI**	Open Source , High Market Value, Development Platform: Java Platform Independent and Pluggable (Extendible) and Web Based Architecture APIs: TMAPI				
		TopiMaker	Open Source , Very High Market Value, Development Platform: Java Platform Independent and Pluggable Architecture User Interface: GUI. APIs: ---------	**Toma**	Organization/ Project under which developed: Ontopia (Space Applications) Non Implemented Language High Query Results and Accuracy Scalable & Extendable Provides more powerful syntax features Currently, Toma queries can only be run using a command-line client		
		Touchgraph	Open Source , High Market Value, Development Platform: Java Platform Independent and Pluggable Architecture. User Interface: GUI APIs: -----				

the technologies. Therefore, to bring RDF and TMs into the main stream market there is an intense need of killer applications which should be powerful, smart to work according to contexts, have clear and different types of visual effects (.i.e., tree, and graph etc.), flexible, cheap, and environment friendly.

6. RDF AND TOPIC MAPS INTEROPERABILITY

RDF and TMs were mainly developed to add metadata and define metadata relationship between web contents so that they should be more machine processable while reduce human time and efforts. However, the two technologies are the standards of two different and independent organizations, developed independently way,and in opposite direction.Therefore, a need arouse for the web researchers to develop strategies for the interoperability and inter-conversion of RDF and TMs. The interoperability between the two technologies should be so flexible that each standard should use the serialization formats, query languages, application programs, and tools of another standard without the problems of the underlying structures. When technologies designed for the same purposes are not interoperable, and each uses its own proprietary mechanism, it becomes difficult and time-consuming for the programmers to get maximum benefit out of them.

The parallel and rapid development in RDF and TMs technologies, the development of complex and sophisticated tools for them, and the lack oftheir interoperability compelled web researchers to solve great need of interoperability. Therefore, an alarming situation arouse that the consequences will be two different islands on the Web of future, which will be complex, not flexible, and out of reach of the common users. In short, we can say that the main reason for the need of interoperability of both of the standards is to elevate dependence of the future web on two different and independent technologies and ensure its availability and easy access to the common users.

6.1. Interoperability Problems

To fulfill the gap of interoperability between RDF and TMs technologies, different techniques were introduced by the researchers named, Garshol, Lacer, Moore and Ciancarini. The techniques developed by the Web researchers worked upto a minimum extent while the major problems remain till now. These problems are due to the different underlying structures, storage, accessing mechanisms, and tools of both of the technologies. The problems taking place at the time of inter-conversion between RDF and TMs are also due to the interoperability issues. Some of the issues are:

- URIs, Addressable and Non-Addressable Subjects: RDF URIs are used for finding anything on the Web whether it is web resource, literal, or blank nodes. TMs URIs find topics, associations between topics, and their occurrences uniquely and universally [24]. From URIs point of view, RDF still has ambiguities because there is no understandable and reliable justification of how URIs can be associated with the Web.
- Association Roles in TMs and Binary Relationships in RDF: A major interoperability problem between TMs and RDF is related to the representation of relationships. TMs use n-array associations to represent a statement while RDF uses binary relationships. In RDF, a single statement is used in the form of SPO (.i.e., Subject, Predicate, and Object), in which Predicates establishes a relationships between Subjects and Objects. TMs, on the other hand, consist of non-binary relationship in which topics and their occurrences are related by associations and each topic plays multiple roles using an association, also called association role. When the concept of multiple association roles is applied to RDF, itcould create large and ambiguous RDF assertions which are not satisfactory from the programmers point of view.
- Scopes in Topic Maps and Reification in RDF:The main reason of applying scope to the TMs data is due to the large number of resources with the same names and meanings. With the help of scopes, these resources can easily be distinguished from each others. On the other hand, there is no successful mechanism defined for describing contextual information using scopes in RDF. An assertion, which describes something about another statement, can be easily done with the

help of RDF property called reification. In TMs reification is complex and can be implemented in several steps.

- Serialization: One of the leading factor effecting RDF and TMs interoperability are the different machine processable syntaxes used by the underlying technologies for mapping concepts and their associations. If these technologies are to be used completely interchangeably then we should go beyond schema level interoperability by identifying equivalent and distinct constructs and devising mechanisms for conversion from one to another.
- Tools& APIs: RDF and TMs tools and APIs have different scalability standards where one is effectivein one situation while another in another situation. Several tools developed for RDF and TMs storage, editing, and browsing/visualization can work effectively to store, create, edit, browse, and visualize Semantic Web metadata. However, there is no such tool yet developed to store, write, edit, browse, and visualize both RDF and Topic Maps resources. To provide ease and advancements, universal tools capable of browsing and visualizing both RDF and Topic Maps data are needed to be investigated.
- Query Languages: Topic Maps query languages have gained less popularity as compared to RDF query languages. Therefore, a common set of protocols or standards needed to be investigated for applying RDF query languages equally to TMs with the same ease, reliability, and higher performance.
- Applications: The Lack of killer application is the prominent factor affecting the wide spread use of RDF and TMs. In spite of availability of a number of applications and tools working efficiently with RDF and TMs data with much better features and characteristics, killer applications are still missing. It is due to the lack of adaption of these technologies by the main stream market, poor definition and exploration, failure to model knowledge in a standardized graphical notation, absence of use cases, repository and projects, no agreement on scope usage and merging, and less number of application developers due to missing of RDF and TMs courses in university curriculum.

6.2. Success in Interoperability

RDF and TMs models both heavily depend on metadata and ontologies. Therefore, for the successful interoperability between these two models, efficient and standardized methods are necessary to establish relationship between metadata and ontologies of both of the standards. To fulfil this need, some of the successful ontologies and metadata standards such as Dublin Core, FOAF, SKOS, and DCMI have already been implemented by different communities for the integration of both of the standards. However, despite of all efforts, a little success has been achieved for solving the interoperability problem of RDF and TMs.

- Schema Level Mappings: RDF uses RDF-Schema and Web Ontology Language (OWL) languages, while TMs uses constraints languages (.e.g., TMCL) for creating schema. A useful two way mapping between RDF and TMs became is possible at the schema level[25]. Besides this, direct conversion is also possible from TMs constraint language into RDF constraint language, although the scope of this conversion is limited.
- Both the technologies use URIs as identifiers where RDF uses direct (using subject locator) mode for identification and TMs uses both direct and indirect (using subject identifiers) modes for identification. This need has been recognized by the Semantic Web community and the researchers, Garshol, Moore and Ogievetsky have contributed their efforts for solving the problem up to some extent. The best solution to fulfil the need of interoperability is to consider and use URIs as both subject identifiers and subject locators and both of these should be allowed to act as URIs. For the successful implementation of this need, Garshol's work is significant. Garshol explicitly treated multiple identifiers and postulated that the resultsproduced more interoperability between TMs and RDF.
- W3C Working Draft for Interconversion: W3C established a group for the purpose of the interconversion of RDF and TMs data called RDF/TMs Interoperability Task Force

(RDFTM). The main goal of the group is to develop approaches for making possible the interoperability between RDF and TMs at the data level. With the successful implementation of this approach, the interconversion of RDF and TMs data will be possible, regardless of the possibility of losing unacceptable information or variation in the semantics. It will also be possible that after the successful implementation, sharing of vocabularies between these two technologies and the translated results can be queried in terms of the target model. The five major interconversion proposals included in W3C surveys includes Moore proposal, Stanford proposal, Ogievetsky proposal, Garshol proposal, and Unibo proposal. A general criterion was set for the evaluation of all interconversion proposals, which was completeness and naturalness. All these approaches solve the problems of interoperability problems upto little extent but the major problems remains till now.

6. CONCLUSION AND FUTURE WORK

Semantic Web technologies RDF and TMs aim to make web information machine processable by enriching them with semantics and reasoning capabilities. Although sharing the same vision, RDF and TMs were developed independently by two different organizations, resulted into complex and non-interoperable interchange syntaxes, query languages, schema languages, constraints languages, and tools and applications. The independent technologies resulted into the possibility of creating non-interoperable islands on the Web of future. To solve this problem and make both of the technologies work in interoperable manner, several techniques were developed by different people over the past several years. The successes of these techniques are practically possible but very limited mainly due to the different underlying storage structures, access mechanisms, different interchange syntaxes, and different constraint and schema languages.

This paper presented, a comprehensive literature study of both RDF and TMs technologies and articulates comparisons of their architectures, serialization formats, tools and APIs, query languages, applications, and models. A thorough investigation of all aspects of both of the technologies is performed for determining how far RDF and TMs can be used together beyond providing support for import/export of serialization formats. It is observed that the interoperability between RDF and TMs is possible,subjected to solving certain issues, and they could work together in complementary fashion. However, it lost of certain semantics and accuracy might occur during the inter-conversion process. Comparatively, RDF is found more suitable for high grained annotation of web resources, whereas, TMs is useful for providing a high level ontological interface to a resource space. We found that that the two technologies are the two equivalent medicines for the same disease. They are aiming to address the same Infoglut problem in their own but in parallel ways. Therefore, sophisticated methods should be developed to make them complementary not contradictory.

To practically test the concept, book ontology is developed in both RDF and Topic Maps standards by employing their respective ontology development tools and serialization formats. The ontology will be presented in the future work. The ontology is imported in applications and results are evaluated. Results justified that that the interoperability between RDF and TMs is possible enabling them to work in complementary fashion.

7. REFERENCES

1. Berners-Lee, T.& R. Cailliau. The World-Wide Web. *Communications of the Association for Computing Machinery* 37(8):76 – 82 (1994).
2. Berners-Lee, T. J. Handler. & O. Lassila. The semantic web. *Scientific American* 284*:* 34-43 (2001).
3. Alam, F., M. A. Khan, S. Ali, & S. Khusro. The Jigsaw of Resource Description Framework (RDF) and Topic Maps Serialization Formats: A Survey. *Proceedings of the Pakistan Academy of Sciences* 51(2): 101-114 (2014).
4. Kinsella, S. B. A. Skobeltsyn, G. Michel, S. John, G. Breslin, & K. Aberer. From Web 1.0 to Web 2.0 and Back – How did your Grandma Use to Tag?. *Proceeding of the 10th Association for Computing Machinery (ACM) Workshop on Web Information and Data Management*, USA (2008).
5. Singh, D., R.K. Mishra, & Chandra Shekhar. Exloring Semantic Web Using Ontologies. *Stem Cell* 3(2): 11-15 (2008).

6. Tolle,K.Introduction to RDF.[Online]. Internet Available: http://www.dbis.informatik.uni-frankfurt.de/~tolle/RDF/DBISResources/RDFIntro.html (Accessed: January 21, 2013).
7. Powers, S. *Practical Resource Description Framework (RDF), 1st ed*. O'Reilly & Associates, North Sebastopol, Canada (2003).
8. Sauermann, L. & R. Cyganiak. *Cool URIs for the Semantic Web*. Technical report, W3C Interest Group(2008).
9. McBride, B. *Jena: Implementing the RDF Model and Syntax Specification*, Technical report. Hewlett Packard Laboratories, UK (2001).
10. Brickley, D., & R.V. Guha. *RDF Vocabulary Description Language 1.0: RDF Schema*.Technical report. W3C Interest Group (2004).
11. Horrocks, I. Ontologies and the semantic web. *Communications of the Association for Computing Machinery* 51(11): 58–67 (2008).
12. Antoniou, G. & F.V. Harmelen.*A Semantic Web Primer*. The MIT Press, Massachusetts Institute of Technology (2004).
13. Pepper, S. Topic maps. In: *Encyclopedia of Library and Information Sciences*. CRC Press, Boca Raton (2009).
14. Pepper,S. & G. Moore. XML Topic Maps (XTM) 1.0 [Online]. Internet Available: http://www.topicmaps.org/xtm/ (08 June 2001) (Accessed: November 13, 2012).
15. Pepper, S. Topic maps. In: *Encyclopedia of Library and Information Sciences*, *3rd ed*. Taylor & Francis Group, USA (2010).
16. Weber, G.E. R. Eilbracht, &S.Kesberg.TopicMaps as Application DataModel for Subject-centric Applications. *Fourth International Conference on Topic Maps Research and Applications (TMRA)*, Germany (2008).
17. Newcomb, S.R. *A Perspective on the Quest for Global Knowledge Interchange*. XML Topic Maps: Creating and Using Topic Maps for the Web, Wesley, Boston (2003).
18. Pepper, S. *Topic Maps: Encyclopedia of Library and Information Sciences*, *3rd ed*. Taylor & Francis Group, USA (2010).
19. Ahmed, K. &G. Moore. An Introduction to Topic Maps. [Online]. Internet Available: http://msdn.microsoft.com/en-us/library/aa480048.aspx (July 2005) (Accessed: January 26, 2013).
20. Serialization.Serialization [Online]. Internet Available: http://en.wikipedia.org/wiki/Serialization (Accessed: February12, 2013).
21. What is Serialization in .NET, types of Serialization and why we need it while developing an application? [Online]. Internet Available: http://www.c-sharpcorner.com/interviews/answer/499/ (Jul 27, 2006) (Accessed: January19, 2013).
22. Hatzigaidas, A., A. Papastergiou, G. Tryfon, & D. Maritsa. Topic maps existing tools: a brief review. *Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics (ICTAMI 2004)*, Greece (2004).
23. Roos, D. How to Leverage an API for Conferencing. [Online]. Internet Available: http://money.howstuffworks.com/business-communications/how-to-leverage-an-api-for-conferencing1.htm (Accessed: December 28, 2012).
24. Pepper, S., & S. Schwab. *Curing the Web's Identity Crisis*. Technical report, Ontopia (2003).
25. Pepper, S. *Ten Thesis on Topic Maps and RDF*. Technical Report, Ontopia (2002).