

# REPRESENTING INFORMATION FOR KNOWLEDGE MANAGEMENT: STRUCTURES FOR COMMUNICATION

John Zacharias Theophanous & Erica Cosijn  
Department of Information Science, University of Pretoria  
jtheophanous@postino.up.ac.za; ecosijn@postino.up.ac.za

## **ABSTRACT**

This paper focuses on the revolution that the Semantic Web, and its related technologies, have promoted. The Semantic Web simply refers to a conglomeration of technologies that enable the seamless interoperability of computerised devices. Database systems are the core of the Semantic Web. Database systems ought to be developed with a platform-independent and interoperable structural markup language, such as XML. XML may be displayed and presented on a variety of media with the application of a formatting markup language, such as XSL. The database structure, and the data structures thereof, should be described to a computer system with the use of an XML schema. In order for the computer system to distinguish a newer version of a database system from an older version, the database system should be accompanied by a set of metadata. Ontologies are required to describe concepts to a computer system. Typical ontologies consists of taxonomies and inference rules. Topic Maps are used to merge two or more ontologies together.

## **1. INTRODUCTION**

Imagine a world where all devices will be able to communicate with one another seamlessly. Fridges will be able to automatically order groceries via the Internet. Mobile telephones will have the capability to book airline tickets. Personal digital assistants (PDA's) will be able to liaise between the computer at work, the computer at home, and the laptop, in order to track appointments and schedules.

The above-mentioned scenarios are not going to appear too far into the future; in fact, their theoretical foundations are being constructed presently. Berners-Lee, Hendler, and Lassila (2001) have proposed the idea of the Semantic Web, which is basically a mechanism for enabling semantic interconnectivity. The Internet in its present state allows interconnectivity via hyperlinks with the utilisation of the HyperText Markup Language (HTML). The major problem with respect to the linkage abilities of the present-state Internet refers to the fact that the hyperlinks are created by humans and not by machines, which means that the links may or may have any semantic meaning with the document contents. The Semantic Web envisages links that will be created by machines based upon the semantics of the document contents (i.e., semantically associated links). The possibilities espoused by the Semantic Web are limitless, since databases may be seamlessly merged and linked with one another by machines, without the need for any human intervention.

An example is necessary at this point in order to clarify certain issues. Suppose that I have three computer systems: one at work, one at home, and a laptop. On each of my computer systems, I have installed an organiser application that allows me to schedule and organise my day in order to track appointments and schedules. The organiser functions by making use of a back-end relational database management system in order to store and retrieve the data. Presently, the fact that I own three computer systems means that all three of my computer systems have to be linked via a private network subnet (i.e., not via the Internet). In addition, there is no way to update my personal digital assistant (PDA) while I am on the move, except to utilise one of my computer systems as a docking station in order to load the relevant data, on the proviso that that particular computer system's organiser database has been recently updated in order to reflect any changes. With the advent of the Semantic Web, all of the three devices would be able to communicate seamlessly with one another via the Internet, constantly updating themselves, from a central server utilising only one database.

## **2. THE SEMANTIC WEB**

The power of the Semantic Web may be illustrated by means of a simple example. Consider a fridge that is connect to the Internet and is able to scan all of its contents for available amounts, expiration dates, and so on.

If the fridge senses that there is a shortage of a certain item, the fridge server contacts the supermarket customer server. Then the supermarket customer server accesses the customer database and requests that the profile of this customer be activated. Consequently, the supermarket server retrieves the record of this customer.

The supermarket server then contacts the bank of the customer and requests the bank server to retrieve the security and permissions profile of this customer. The supermarket server checks the security and permissions profile of this customer and then requests the bank server for permission and access to this customer's account in order to enable live transactions between the bank and the supermarket. If the customer profile allows for permission, the bank server enables transaction-sharing with the supermarket server.

The supermarket server checks the order from the fridge database. In addition, the supermarket server creates a list of items that were requested, each with a corresponding price, and the total of all the items together. The supermarket server contacts the bank server and requests whether the customer has enough money in his or her account. If the customer possesses enough money in his or her bank account, the bank server gives the supermarket server the go-ahead for the purchases.

Then the supermarket server requests that the required items be brought to the cashier. The supermarket server then transfers money from the bank account to the bank account of the supermarket (i.e., by electronic funds transfer (EFT)). All the transactions and connections are then terminated. The customer then gets his or her order delivered to his or her home.

In order to harness the power of the Semantic Web, certain pertinent technologies ought to be embraced. Obviously, the core of the Semantic Web is the existence of databases. Databases ought to be developed with a platform-independent and interoperable structural markup language, such as the eXtensible Markup Language (XML). The XML may be displayed and presented on a variety of media with the application of a formatting markup language, such as the eXtensible Stylesheet Language (XSL). The database structure, and the data structures thereof, should be described to a computer system with the use of an XML schema. In order for the computer system to distinguish a newer version of a database system from an older version, the database system should be accompanied by a set of metadata. In order to describe concepts to a computer system, ontologies are required. A typical ontology consists of taxonomies and inferences. Topic Maps are utilised in order to merge two or more ontologies together.

## 2.1 DATABASES

Databases may be considered as simply being tables of related data. As with all tables, database tables are composed of a number of columns and rows. The columns of a database table are known as the records (i.e., tuples) of the database. The rows of a database table are known as the fields of the database. Therefore, in terms of a database table, records (i.e., columns) consist of a number of fields (i.e., rows). To a computer system, the data (i.e., contents) present within a database is meaningless; it is the structure of the database that is of meaning to a computer system. Consider the following database table:

	Record 1	Record 2
Name	John Zacharias Theophanous	Erica Cosijn
Work Telephone Number	+27 (0)12 420 4026	+27 (0)12 420 3669
Fax Number	+27 (0)12 362 5181	+27 (0)12 362 5181
Work Address		
Address	Office HSB 17-1, University of Pretoria	Office HSB 17-1, University of Pretoria
City	Pretoria	Pretoria
Postal Code	0002	0002
Country	South Africa	South Africa
Electronic Mail	jtheophanous@postino.up.ac.za	ecosijn@postino.up.ac.za

As may be seen from the database table above, there are two records in the database system. Each record consists of five fields (i.e., Name, Work Telephone Number, Fax

Number, Work Address (Address, City, Postal Code, & Country), and Electronic Mail). Databases, such as the one above, ought to be developed with a platform-independent and interoperable structural markup language, such as the eXtensible Markup Language (XML).

## 2.2 STRUCTURAL MARKUP LANGUAGES & XML

The eXtensible Markup language (XML) is a programming language standard that was released by the World Wide Web Consortium (W3C) in 1996. XML is a standard that is utilised for the logical representation of data. As such, XML may be employed for representing the data that is present in a database table. XML allows for the seamless interchange and interoperability of data. The above database table may be represented in XML by means of the following code:

```
<?xml version = "1.0" encoding = "utf-8" standalone = "no"?>
<?xml-stylesheet type = "text/xsl" href = "XSLTFile.xsl"?>
<address_book>
  <record number = "1">
    <name>John Zacharias Theophanous</name>
    <work_telephone>+27 (0)12 420 4026</work_telephone>
    <fax_number>+27 (0)12 362 5181</fax_number>
    <work_address>
      <address>Office HSB 17-1, University of Pretoria</address>
      <city>Pretoria</city>
      <post_code>0002</post_code>
      <country>South Africa</country>
    </work_address>
    <electronic_mail>jtheophanous@postino.up.ac.za</electronic_mail>
  </record>
  <record number = "2">
    <name>Erica Cosijn</name>
    <work_telephone>+27 (0)12 420 3669</work_telephone>
    <fax_number>+27 (0)12 362 5181</fax_number>
    <work_address>
      <address>Office HSB 17-16, University of Pretoria</address>
      <city>Pretoria</city>
      <post_code>0002</post_code>
      <country>South Africa</country>
    </work_address>
    <electronic_mail>ecosijn@postino.up.ac.za</electronic_mail>
  </record>
</address_book>
```

When the above XML dataset is rendered in an XML parser and viewer (in this case, Microsoft Visual Studio .NET), the following data table is generated:

Data for record					
	name	work_telephone	fax_number	number	electronic_mail
▶ ⊕	Erica Cosijn	+27 (0)12 420 3669	+27 (0)12 362 5181	2	ecosijn@postino.up.ac.za
⊕	John Zacharias Theophanous	+27 (0)12 420 4026	+27 (0)12 362 5181	1	jtheophanous@postino.up.ac.za

The “+” sign next to each record indicates that another table has been created within the same database in order to house the “work\_address” field, and its various subfields (i.e., “address”, “city”, “post\_code”, and “country”).

The idea of records consisting of a number of fields remains the same for XML as for database tables. XML is a programming language standard that places emphasis upon the structure of data within a database system. The angle brackets in XML and other markup languages are known as tags and their role lies in structuring the database system. The text within the tags refers to the field names of the database, whilst the text outside of the tags refers the data of the fields. The tags and the field data are collectively referred to as a dataset. XML requires that the programming logic be well-formed, since, as was previously mentioned, the data (i.e., contents) present within a database is meaningless to a computer system; it is the structure of the database that is of meaning. XML does not allow for any presentation and formatting capabilities of the data within a dataset. XML may be displayed and formatted for use on a variety of media with the application of a formatting markup language, such as the eXtensible Stylesheet Language (XSL).

### 2.3 FORMATTING MARKUP LANGUAGES & XSL

The eXtensible Stylesheet Language (XSL) is programming language that complements XML. XSL is an application of the XML specification. Whilst XML focuses upon the structure of the data within a dataset, XSL focuses upon the presentation and formatting of the data within a dataset. The real benefit of XSL is in its ability to process one XML dataset and format it for presentation on a variety of media, such as: a desktop computer or a laptop, a mobile telephone, a personal digital assistant, and so on. The above XML dataset may be formatted for presentation on a desktop computer or laptop in XSL by means of the following code:

```
<?xml version = "1.0" encoding = "utf-8" standalone = "no"?>
<xsl:stylesheet xmlns:xsl = "http://www.w3.org/1999/XSL/Transform" version =
"1.0">
  <xsl:template match = "/">
    <html>
      <head>
        <title>Contacts Database</title>
      </head>
      <body>
        <table border = "1">
          <xsl:for-each select = "address_book/record">
            <tr><td><xsl:value-of select = "name" /></td></tr>
            <tr><td><xsl:value-of select = "work_telephone" /></td></tr>
            <tr><td><xsl:value-of select = "fax_number" /></td></tr>
            <tr>
              <td><xsl:value-of select = "work_address/address" /></td>
              <td><xsl:value-of select = "work_address/city" /></td>
              <td><xsl:value-of select = "work_address/post_code" /></td>
              <td><xsl:value-of select = "work_address/country" /></td>
            </tr>
            <tr><td><xsl:value-of select = "electronic_mail" /></td></tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

The above process is referred to as transformation (or transformative presentation), since the XML dataset is being transformed from an XML dataset into an HTML document for

presentation on a web browser. The XML dataset could have just as easily been transformed from an XML dataset into a WML (i.e., Wireless Markup Language) document for presentation on a mobile telephone or from an XML dataset into an SGML (i.e., Standard Generalised Markup Language) document for presentation on a Desktop Publishing (DTP) application. It is important to notice that nothing ever happened to the XML dataset itself; which means that it may be reused in other ways over and over again. The language that was utilised above is known as the eXtensible Stylesheet Language Transformations (XSLT) language. The XML database structure, and the data structures thereof, should be described to a computer system with the use of an XML schema.

## **2.4 XML SCHEMAS**

Every database system, whether generated in XML or not, requires some sort of a schema. An XML schema is simply a set of predefined rules that describe a given XML document. An XML schema defines the elements that can appear within a given XML document, along with the attributes that can be associated with a given element (Microsoft, 2002). An XML Schema also defines structural information about an XML document, such as which elements are child elements of others, the sequence in which the child elements may appear, and the number of child elements.

XML requires that the programming logic be valid according to the supplied XML Schema, since, as was previously mentioned, the data (i.e., contents) present within a database is meaningless to a computer system; it is the structure of the database that is of meaning. In order for a computer system to distinguish a newer version of a database system from an older version, the database system should be accompanied by metadata.

## **2.5 METADATA**

Metadata simply refers to data about data. Metadata is methodology that is utilised to describe data. Metadata has been around since antiquity. Publishers and librarians have utilised metadata for the purposes of categorising and cataloguing, in terms of titles, authors, International Standard Book Numbers (ISBN), publishers, publication dates, and so on.

Metadata is utilised in the context of databases in order to indicate the underlying properties of the database, such as the version number of the database, the date and time of the previous update of the database, the date and time of previous update of a particular record, and so on.

Apart from the above usage of metadata, Microsoft makes extensive use of metadata within its new .NET framework. Metadata, utilised in this development context, poses several advantages including the fact that two files with exactly the same names and extensions may exist side-by-side on the same computer system and within the same folder without causing

conflict and confusion within the system. How is this possible? In essence, each resource (such as a file, module, or component) is known as self-describing through its unique set of metadata (such as, date and time of creation, date and time of last modification, version, and so on) that describes the resource, which is written in XML.

The Resource Description Framework (RDF) is a metadata specification that is based upon the XML specification and was released by the W3C in 1999. RDF allows for the interoperability of metadata on the web. The following code snippet clearly displays an example of RDF:

```
<rdf:RDF
  xmlns:dc = "http://purl.org/metadata/dublin_core#"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
  <rdf:Description about = ""
    dc:creator = "Theophanous, John Zacharias; Cosijn, Erica"
    dc:description = "This is a conference paper about the Semantic Web"
    dc:language = "en-GB"
    dc:format = "application/msword"
    dc:date = "2003"
    dc:type = "Conference paper"
    dc:title = "Representing Information for Knowledge Management"
    dc:subject = "Conference paper"
    dc:rights = "Copyright: the authors"
  />
</rdf:RDF>
```

If the above code snippet is examined, the “dc:” namespace is prevalent throughout it. The “dc:” namespace indicate that the Dublin Core Metadata initiative (DCMI) is utilised within the RDF implementation. The DCMI specification is a metadata element set that is composed of fifteen elements: Title, Creator, Subject, Description, Publisher, Contributors, Date, Type, Format, Identifier, Source, Language, Relation, Coverage, and Rights.

## 2.6 TAXONOMIES & ONTOLOGIES

A philosophical ontology refers to the study that deals with issues of being and existence - the constituents of reality. In Information Systems (IS), however ontologies deal rather with specifying and clarifying objects in a particular domain and structuring them within a framework of some formal theory with a well-understood logical structure - on levels of syntactic and semantic structure both. Ontology in the IS sense can therefore be defined as a specification of a conceptualisation, but in a very pragmatic sense.

Noy and McGuinness (2001) define an ontology as a formal explicit description of concepts in a domain of discourse, which are known as classes or concepts. The primary reasons for developing an ontology include (Noy & McGuinness, 2001):

- To share common understanding of the structure of information between software agents;
- To enable the reuse of domain knowledge;
- To make domain assumptions explicit; and

- To analyse domain knowledge.

There are certain basic rules when ontologies are constructed (Smith, 2000), amongst others:

- It should take the form of a tree in the mathematical sense;
- There should be only one top (or maximal) node, which should include all the lower categories;
- There should be defined minimal nodes which do not include sub-categories; and
- There should be a finite number of steps between the maximal category and each minimal category; and there should ideally be the same number of steps between the topmost node and all the lowest level node.

According to Berners-Lee, Hendler, and Lassila (2001), the most typical kind of ontology for the Web possesses a taxonomy and a set of inference rules. A taxonomy defines classes of objects and the relations between them. Classes are described by properties, which are also known as slots or roles, in terms of the features and attributes of the class and by restrictions on the properties, known as facets or role restrictions. Inference rules supply an ontology with extended power by means of making use of simple “if-else” type statements and the like. An ontology may express an inference rule in terms of the following example: “If a residential address or postal address originates from the United Kingdom then it uses a postal code; however, if a residential address or postal address originates from the United States then it utilises a zip code”. It is important to notice that a postal code and a zip code are simply synonyms of each other, but a computer system does not make this connection. An ontology combined with inference rules is able to make the connection between an postal code and a zip code clear to a computer system.

A computer system makes use of data without transforming it into information. By means of utilising ontologies, a computer system is able to manipulate concepts (i.e., data) much more effectively in ways that are useful and meaningful to the human user. In order to merge two or more ontologies together, topic maps are required.

## **2.7 TOPIC MAPS**

According to Pepper (2000), “topic maps are an ISO standard for describing knowledge structures and associating them with information resources. As such they constitute an enabling technology for knowledge management”. The purpose of topic maps is to provide ways of navigating large and interconnected corpora. The article by Pepper (2000) explains the basic concepts of topic maps as *Topics* (“a thing, ... about which anything whatsoever may be asserted by any means whatsoever”), *Associations* (which describes relationships between topics) and *Occurrences* (information resources to which topics are linked) - *the TAO of Topic Maps*.

Topic maps started as a solution to managing representation of information in complex document situations. Later the scope was broadened to also cover other navigational functions, for example glossaries, thesauri and cross-references. However, topic maps do not simply replicate the features of a printed index, it generalises these features. It is important to realize that links within topic maps are independent of actual documents, and can be valuable information sources in themselves.

Topic Maps are well suited to representing ontologies. According to Wrightson (2001), “the key role of ontologies in many real-world Knowledge Representation applications, the ability of Topic Maps to link resources anywhere in the Semantic Web, and then organize these resources according to a single ontology, will make Topic Maps a key component of the new generation of Web-aware knowledge management solutions”.

Furthermore, emerging techniques for simplifying and merging ontologies can be used to combine or articulate Topic Maps representing different ontologies, enabling disparate sets of information resources to be used together in a controlled way. This may be the key capability for realizing the vision of the Semantic Web (Wrightson, 2001).

### **3. XML WEB SERVICE FRAMEWORKS**

An XML Web Service refers to a programmable application component that is accessible by means of using standard Internet protocols. XML Web Services components may be reused over and over without concern for the implementation of the service. XML Web Services may appear on web pages in the form of currency converters, weather reports, and so on. XML Web Services may be considered as being early prototypes of the Semantic Web

XML Web Services have been made a reality with the advent of the Microsoft .NET framework, the Sun Microsystems' Java 2 Enterprise Edition (J2EE) framework, and the Sun Microsystems' Java 2 Mobile Edition (J2ME) framework, all of which simply refer to platforms for building, deploying, and executing XML Web Services and applications. The key to both of the above frameworks refers to the fact that they promote applications that are platform independent.

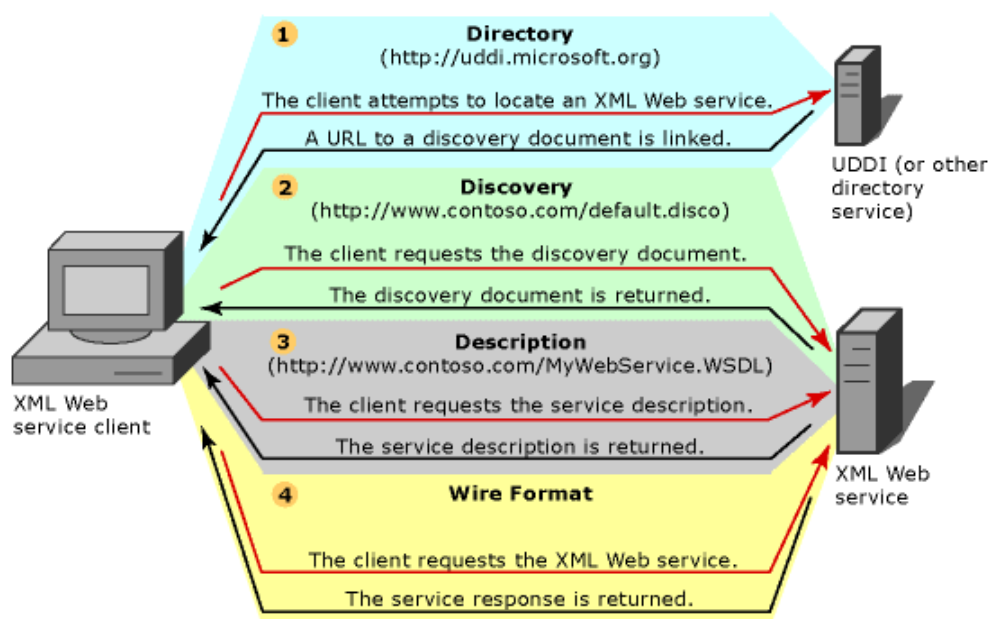
The Microsoft .NET framework is based primarily on the Visual Basic .NET, on the Visual C# .NET, and on the Visual J# .NET programming languages. The Sun Microsystems' J2EE and J2ME framework is based primarily on the Java programming language.

The Simple Object Access Protocol (SOAP) refers to a lightweight protocol that is intended for exchange of information in a decentralized and distributed environment (Microsoft, 2002). SOAP is an XML-based protocol that allows the distribution and operation of XML Web Services.

The Universal Description, Discovery, and Integration (UDDI) mechanism refers to a mechanism whereby XML Web Service providers are able to promote the existence of their XML Web Services (Microsoft, 2002). In addition, UDDI is a mechanism whereby XML Web service consumers are able to locate XML Web Services of interest.

The Web Services Description Language (WSDL) is an XML-based contract language that is utilised for describing the XML Web Services that are offered by a server (Microsoft, 2002).

Microsoft (2002) conveniently summarises XML Web Services, and their related technologies, with the aid of the following diagram (Taken from Microsoft):



#### 4. ARTIFICIAL INTELLIGENCE & AI AGENTS

Artificial intelligence (AI) and AI agents are the core of the Semantic Web. AI agents, or robots, refer to the software components that make the Semantic Web a reality. AI agents are multitasking components, in terms of the fact that AI agents merge databases seamlessly, search multiple databases simultaneously, create semantic links between resources, and so on.

#### 5. CONCLUSION

This paper has discussed the revolution of the Semantic Web, in terms of the structures provided for communication in representing the information for the purposes of knowledge management. The core of this paper focused on the various technologies that are required in order to make the Semantic Web a reality.

## 6. BIBLIOGRAPHY

Berners-Lee, T., Hendler, J. & Lassila, O. 2001. *The Semantic Web*. [Online]. Available: [http://www.sciam.com/print\\_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21](http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21).

Microsoft Corporation. 2002. *The Microsoft Developers Network (MSDN)*. [Online]. Available: <http://www.microsoft.com>.

Noy, N.F. & McGuinness, D.L. 2001. *Ontology Development 101: A Guide to Creating Your First Ontology*. [Online]. Available: [http://protege.stanford.edu/publications/ontology\\_development/ontology101-noy-mcguinness.html](http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html).

Pepper, S. 2000. *The TAO of Topic Maps: Finding the way in the age of infoglut*. Paper presented at XML Europe 2000, Paris, France, 12-16 June 2000. [Online]. Available: <http://www.gca.org/papers/xml europe2000/papers/s11-01.html>.

Smith, B. 2000. *Ontology and Information Systems* (Draft paper: 11 December 2001). [Online]. Available: [http://ontology.buffalo.edu/ontology\(PIC\).pdf](http://ontology.buffalo.edu/ontology(PIC).pdf).

Wrightson, A. 2001. *Topic Maps and Knowledge Representation*. [Online]. Available: <http://www.ontopia.net/topicmaps/materials/kr-tm.html>.