A Mechanism for Detecting Dead URLs in XTM-based Ontology Repository

Enesi Femi Aminu
Department of Computer Science
Federal University of Technology, Minna.
Nigeria.

Olumide Sunday Adewale
Department of Computer Science
Federal University of Technology, Akure.
Nigeria.

ABSTRACT

It is a common knowledge in this current age that the challenge is no more the availability of information but accessing the appropriate information. This challenge is attributed to the fact that information is highly unstructured. An innovative technology called XML Topic Map (XTM) is used to address the challenge. However, the technology is equally associated with an issue known as dead links as a result of lack of maintenance which causes a lot of time wastage thereby making the technology not very effective. Therefore, this paper presents an XTM-based system using News domain as case study to solve the issue of dead links. The system is a three-tier architecture implemented with Java Server Pages (JSP). The study offers a time saving system that would assist user to identify URLs that are dead in XTM-based applications as soon as user access topic.

General Terms

Ontology, Semantic Web, XTM, RDF, LTM, Ontopia, URLs, Repository

Keywords

Dead URLs, News, NEWS.net, News Ontology

1. INTRODUCTION

The availability of information and its related traffic on the Internet is always on the rise. For examples; book vendors, news agencies and organizations, or business enterprise in general own by individuals and organizations have been rapidly shifted to the Internet courtesy of web applications. It is often said that in human history the web is possibly the richest information depository. Seeking information is a familiar and important human activity. Most often, we involved in the use of websites and networked information management systems to surf for information. Whatever information users decide to surf for, the tendency that it can be found somewhere in the digital globe is high. But in most cases, surfing for information is not always effective or efficient.

When a user surfs for information, the system frequently returns a huge chunk of unrelated results. Hence, in this current age, the challenge is not the availability he information but getting the relevance information [1]. This challenge is attributed to the fact that information is highly unstructured. Thus, alternative information-organization approaches are required to more effectively and efficiently navigate and retrieve information from systems, web included through an innovative technology-XTM.

1.1 Semantic Web and Ontology

It is pertinent to know that the terms ontology and semantic web are closely related and intervolving as ontology is the stronghold and prime element for building up of semantic web contents. Ontology is a known term that computer science hijacked from other known disciplines like Philosophy and given it a precise technical meaning. Ontology according to [2] is defined as an explicit and formal specification of a conceptualization. And according to topic maps terminology [3], it is defined as a precise description of the kinds of things which are found in the domain covered by the topic map: in other words, the set of topics that are used to define classes of topics, associations, roles, and occurrences. In general, ontology depicts formally a domain of discourse. That is, it defines the terms used to describe and represent an area of knowledge. For example: News media, Medicine, Photography, Shakespeare work, and so on.

The development of ontology will continue to offer a platform for the building of the semantic part of Web. It is ascertained that there are several markup languages that are in existence and using XML (a well structured markup language) to denote ontologies is not unfamiliar. Today, XML Topic Maps (XTM) is the main topic maps syntax and is supported by nearly all topic map tools for developing ontologies. Thus just like RDF, RDFS and OWL of W3C are used for developing ontologies, XTM of ISO equally serves in the same capacity.

While XTM have been identified as a capable technology for ontology and semantic web, in [4] three principal constructs of XTM are Topics, Associations, and Occurrences. Other known constructs are scope, baseName, resourceData, resourceRef, topicRef, member, and the like.

First, the complexity and size of XTM is determined by the number of topics. Topics are the main building blocks of XTM [4]. In its most general view, topics can be anything: a human being, a concept, actually anything regardless of whether it exists or posses any other explicit feature. It forms the foundation for the (XML) topic maps creation. It can equally be seen as a “multi-headed link that points to all its occurrences”. This “link” collects information about a given subject [5].

A topic is expected to have a topic type or probably several topic types. Topic type can be described as that of class-instance relationship. Types stand for the classes in which topics belong to, that is, the group of one topic instance. Topic types are also topics on their own [5].

Second, an association defines as a relationship between one or more topics. For all the classes of topics, associations and the roles involved in associations of Topic Maps are user-definable task, and of course, they are all topics (for instance, they are the subjects of topics that characterize them). Associations are composed of members – for every topic that involve in an association plays a definite member role in the association [6][12].

Finally, in XTM, information resource(s) may be linked with a topic, and vice versa. These resources are termed as occurrences of the topic. Occurrences can be described performing a related function like the page numbers in a back-of-book index [7]. The resources for occurrences are either: resource reference (a URI - Uniform resource identifier) or resource data (expressing a piece of information about a subject) [8]. The occurrences are, usually, external to the XTM document itself, though it might equally be in it, and they are designed at using, normally, URIs [9].

2. RELATED WORK

In the work of [10], they presented a coherent and practical understanding of creating a very large XTM-based ontology by combining the strengths of Ontopia and Wandora which are two different topic maps creation tools with the aims of achieving a very large scale domain. They considered the existence of some interoperability between the tools in order to realize the objective of multiple topic maps tools integration. The whole essence of integration is that the strengths of one tool can complement the other’s weaknesses. However, the authors admitted that it was time-consuming experience to understand both tools and to integrate them. Furthermore, since both XTM creation tools used in the work are suite of java tools, the authors could have as well design a program that can complement or solve the weakness of a tool preferably ontopia, and thereby integrate the program to it for efficiency in time consuming.

Also [11] described the process of (XML) Topic Maps’s application creation for the purpose of Observatory and Planetarium in Hradec Králové, Czech Republic. Two tools were used, TM4L editor for developing the ontology, and Ontopia Navigator Framework for creating the web presentation. The pilot web presentation contains information and links to knowledge resources via the Topic Maps document (ontology) which helped to access web pages related to astronomy and to ensure better navigation. However, they admitted that in a later future, effort should be focused on procedures for maintaining the Topic Maps for consistence utilization of the application.

According to [12], the weakness associated with Universal Description, Discovery and Integration (UDDI) registry of web services is as a result of lack of semantic. An exact keyword for searching item and data need to be provided for the UDDI interface. He addressed the problem with the aid of XTM technology as a good representative of semantic web.

Specifically, the aim of the work was presenting XTM as a powerful agent for knowledge organization and representation. However, there exists the issue of dead links and they are attributed to the lack of XTM maintenance because the creation tool of XTM is still evolving. Therefore, the ONF is explored and exploited to provide remedy to the issues of dead links.

3. MAINTENANCE PROGRAM FOR XTM-BASED ONTOLOGIES

In every system design, maintenance is very critical. Therefore, as a result of lack of maintenance, some existing XTM-based ontologies are associated with dead links. Dead links occurs when URLs (external occurrences) of a topic is wrongly typed by author of ontology or not longer exist in the internet at the time ontology is designed. The tendencies of dead links when developing XTM-based applications are always high since the ontologies (or specifically the occurrences) are hand-coded. Whenever it occurs the system cannot precisely reports that URLs or links are dead. Thus, design a maintenance program on the platform of Ontopia Navigator Framework (ONF) technology which is based on the J2EE (Java 2, Enterprise Edition) platform, it works in any compliant J2EE application server, for example, Apache-Tomcat. It supports JSP (JavaServer Pages), the system can out-rightly report any incident of dead links without allowing the user to do much of the work by continuous searching, which ends with no expected results; and the issue of time wastage is thereby addressed.

4. RANKED SEMANTIC SIMILARITY MODEL

According to [13] Semantic similarity models are specific types of mathematical techniques used to measure the strength of the semantic relationship between concepts (topics) or instances, via a numerical description obtained according to the comparison of information formally or implicitly supporting their meaning or describing their nature. Practically, semantic similarity can be measured for instance by defining topological similarity, by using ontologies to define a distance between topics.

Semantic similarity can be measured by Ranked, Binary Cosine, Jaccard or SemRel models. However, the results of Binary Cosine, Jaccard and SemRel are not always accurate and reliable. In that Binary Cosine and Jaccard do not consider the numbers of occurrences of a topic in an entity [14]. Also, they do not consider the topics related to the ones found in an entity. Therefore, Ranked Semantic Similarity model which measures the XTM-based ontology and behave in the manner of the proposed system described is adopted and used for the case study (News Domain) and its requirement data.

Developing ontology is similar to database design. The difference is, in (relational) database design, tables are required that is composition, of rows and columns with keys and data must be added to these tables strictly according to their design specifications [15]. In ontology design according to [16] we make topics, occurrence and relationships, called associations, individually and then add these associations to the topics. An association (part-of) might be present between more than two topics. As soon as ontology is being built then instances (of these topics otherwise called classes) can be instantiated that store in a model. The model could be in database or file.

The Figure 1 depicts the schematic design of our case study that we refer to as “News Ontology” [15]. The design consists of “nodes” and “edges” which correspond to “topics” and “association” respectively in XTM.

The nodes (for example, Information Technology news, Education news, Business news) at the first level of the design represent the subclasses news of the super class (topic type) - NewsDomain. A subclass node or topic can equally be a super class. Sports News and ForeignScene News are typical examples. A topic can have relationship (that is, association) with one another.

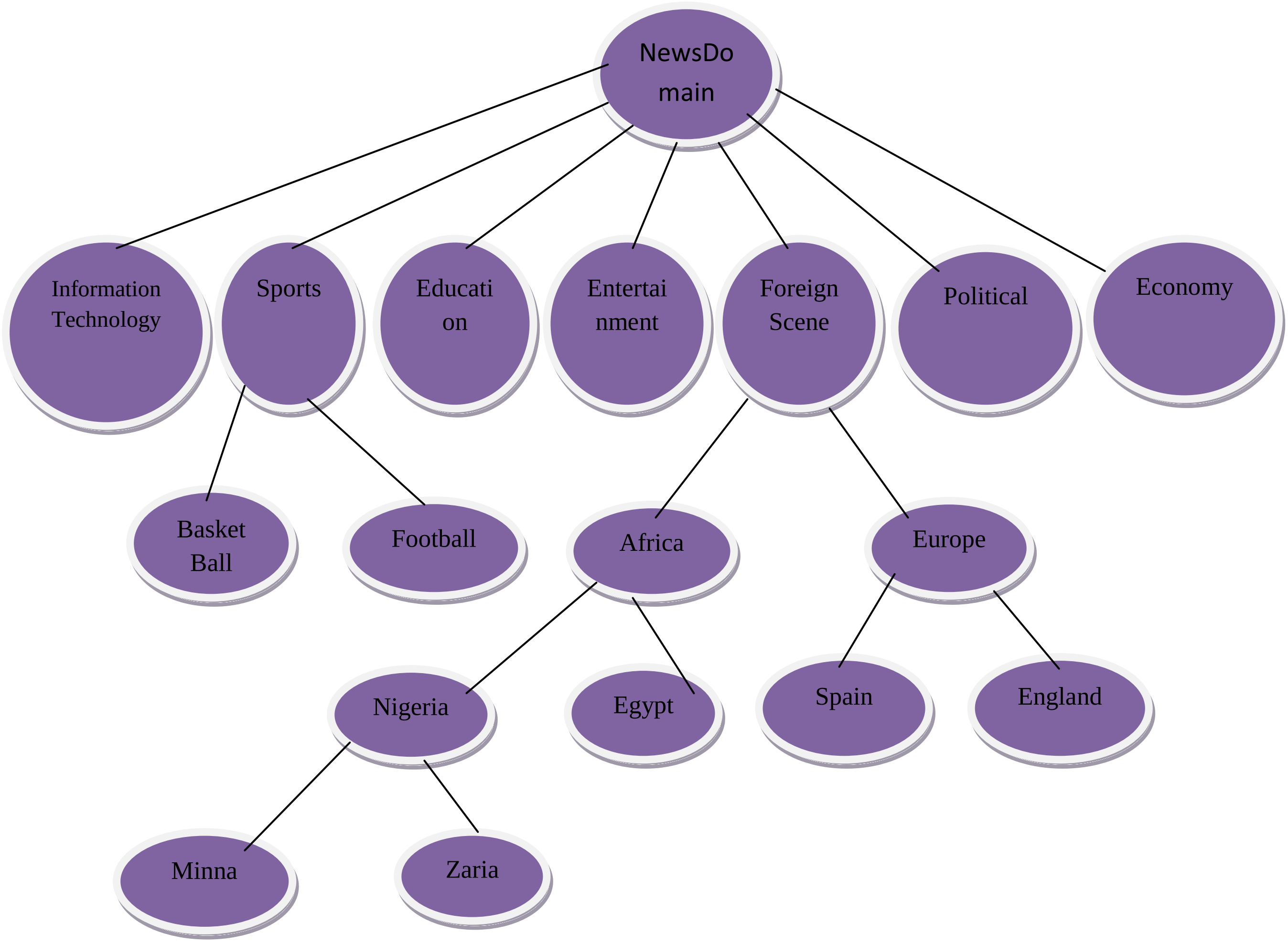


Figure 1: News Ontology

Considering the work of [17] as an idea, we therefore model and define Figure 1 as follows: An NO-model (News-Ontology-model) structure is a 3-tuple structure. That is, NO-model: = (R; N; T) where

R is the root topic of NO-model

N is the set of Non Terminal topics of NO-model.

T is the set of Terminal topics of NO-model.

Root NO-model (RNO-model) structure is defined as RNO := $(\{ROOT\}, \emptyset)$. ROOT is the root topic, each other topic must subclass ROOT (it may do so indirectly). Each level of NO-model must include RNO in order to identify the root topic. Generally, each element or node in NO-model is refers to as topic (t) weather R, N or T. Thus, the ontology contains a set of n topics.

The semantic similarity measures of the XTM-based News ontology is based on *is a, part of, casted by, created by* relationships and checking the semantic relatedness by way of measuring the similarity between news article being read and user profile created by articles read. This measure of [14] is adopted in this work and rewritten as follows.

The user profile is represented as the following set:

$$U = \{t_{1,}^{u}\ \ t_{2,}^{u}\ \ t_{3,}^{u}\ \ \ldots, t_{y}^{u}\ \} \text{ where } t_{i}^{u}\ \in \text{NO.}$$

The user profile consists of y topics identified by NO in the news previously read by the user.

A news article can also be formulated as a set of q topics that appears in the article:

$$N = \{t_{1,}^{n}\ \ t_{2,}^{n}\ \ t_{3,}^{n} \ldots, t_{q}^{n}\ \} \text{ where } t_{j,}^{n}\ \in \text{NO.}$$

When you read news about topic t_1 which is related to topic t_2 and topic t_3 you enhance not only your knowledge in topic t_1 but also in the other two topics. Every topic is assigned a value called rank. For instance, when a user reads about SportingsNews, same user might also be concerned in others, like BSkyBNews, but also in news is about Football. Both are considered to be in direct relation to the topic SportingsNews. Thus we increase the rank for SportingsNews, BSkyBNews, and Football. Topics that are not directly connected to the current topic, also need to be addressed. This means, if a user profile consists of topics $t_{1,}^{u}$ and $t_{2,}^{u}$ and the next news the user reads, contains topic $t_{3,}^{u}$ which is directly related to $t_{1,}^{u}$ but not related to $t_{2,}^{u}$ we increase the rank of t_1, and decrease the rank of t_2. By decreasing the rank for such a topic we make the user profile adaptive to the user's main interest.

The set of related keywords to topic t_i is defined as:

$$r(t_i) = \{\ t_{1,}^{i}\ \ t_{2,}^{i}\ \ t_{3,}^{i} \ldots, t_{p}^{i}\ \ \} \qquad \ldots 3.1$$

R is defined as the union of all related topics to the topics in the user profile:

R = $\bigcup_{t_i^u \in U} r(t_i^u)$ . . . 3.2

And finally U_X is described as the set of all topics and corresponding related topics, it is referred to as the extended user profile:

U_X = U $\cup$ R. . . . 3.3

The use of extended user profile is significant because it will raise the interest of user in certain topics that are not in the user profile, but are related to the topics in the user profile. To compute the final ranks for each topic, we organize the topics in a matrix. This is done because we have to assign a rank to each topic in the extended user profile for each topic the user has read about. Reading about topic t_1 increases its value with 1.0. If topic t_2 is directly related to topic t_1, then its value is increased with 0.5. If there exist a topic, say topic t_3, in the extended profile which is not equal to topic t_1 nor related to topic t_1, its value is decreased with 0.1. These constants were determined by experimenting with values ranging from 0 to 1 with a step size of 0.1. Thus, the matrix's columns contain the items from the extended user profile (U_X) and the rows contain the items from the user profile (U).

Table1 shows a rank matrix, where $t_i^e \in U_X$ and $t_i^u \in$ U. Adding the values of the cells in a column of the matrix, and repeating this process for each column, results in a vector with the final ranks for each topic, in the extended user profile.

Table1: Ranked Matrix

	t_1^e	t_2^e	t_3^e	…	t_x^e
t_1^u	r_{11}	r_{12}	r_{13}	…	r_{1x}
t_2^u	r_{21}	r_{22}	r_{23}	…	r_{2x}
t_3^u	r_{31}	r_{32}	r_{33}	…	r_{3x}
.	.	.	.	…	.
t_y^u	r_{y1}	r_{y2}	r_{y3}	…	r_{yx}

The user might have read one or more news about a topic. Rationally, the user is assumed to be more interested in topics that are found in some news. The number of news the user has read about topic t_i^u, is referred to as the weight w_i^u.

W = { $w_1^u, w_2^u, w_3^u \ldots; \; w_y^u$ }. … 3.4

Thus, the value for each cell in the matrix of Table3.1 is computed using:

$$r_{i,j} = w_i^u * \begin{cases} +1.0 & if \;\; t_j^e = t_i^u \\ +0.5 & if \;\; t_j^e \neq t_i^u, \; t_i^e \in r(\, t_i^u) \\ -0.1 & otherwise \end{cases}$$

. . . 3.5

The final rank for each topic from the extended user profile, can be calculated by adding the values of the corresponding column in the matrix:

Rank (t_j^e) $= \sum_{i=1}^{y} r_{i,j}$ . . . 3.6

The additions are kept in a vector V_U. Every topic in the extended user profile now has a rank. We need to ensure that the range of the ranks is [0,1] before we can compare the user profile with an unread news item. The normalization is done as follows:

$V_U[t_i^e] = \dfrac{t_i^e - \min(t_i^u)}{\max(t_i^u) - \min(t_i^u)}$ . . . 3.7

$t_i^e \in V_U$ and $t_i^u \in V_U$. With this normalization the extended user profile can be compared to a new item that needs to be classified. The new item consists of a set of topics, designated as I:

I = {$i_1, i_2, i_3 \ldots, i_h$} . . . 3.8

The vector containing the ranks for this item is described as V_I:

$V_I =$ ($b_1, b_2, b_3, \ldots, b_c$), . . . 3.9

.

Generally, b_i

$$= \begin{cases} rank(t_i^e) & if \; t_i^e \in \text{I} \\ 0 & if \; t_i^e \; ! \in \text{I} \end{cases}$$

. . . 3.10

Each topic from the extended user profile that appears in the item is assigned the same rank as the one in V_U. The remaining topics are assigned zero. Topics appearing in the item but not in the user profile are ignored.

In this research work it is assumed that all topics found in a news item are equally important. To compare the news item with the user profile we proposed to calculate the level to which the item fits the profile by dividing the sum of the ranks of topics in the item by the sum of the ranks of the topics in the user profile

Similarity (V_I, V_U) = $\dfrac{\sum_{b_i \in V_I} b_i}{\sum_{t_i^u \in V_U} t_i^u}$ . . . 3.11

Having experimenting with values ranges from 0 – 1 with a step size of 0.1, the news item with the highest similarity measure fits best with the user profile.

5. SYSTEM ARCHITECTURE

The architectural design of the system is a 3-tier (Data tier, Server/Maintenance tier and Users Tier) architecture represented in Figure 2.

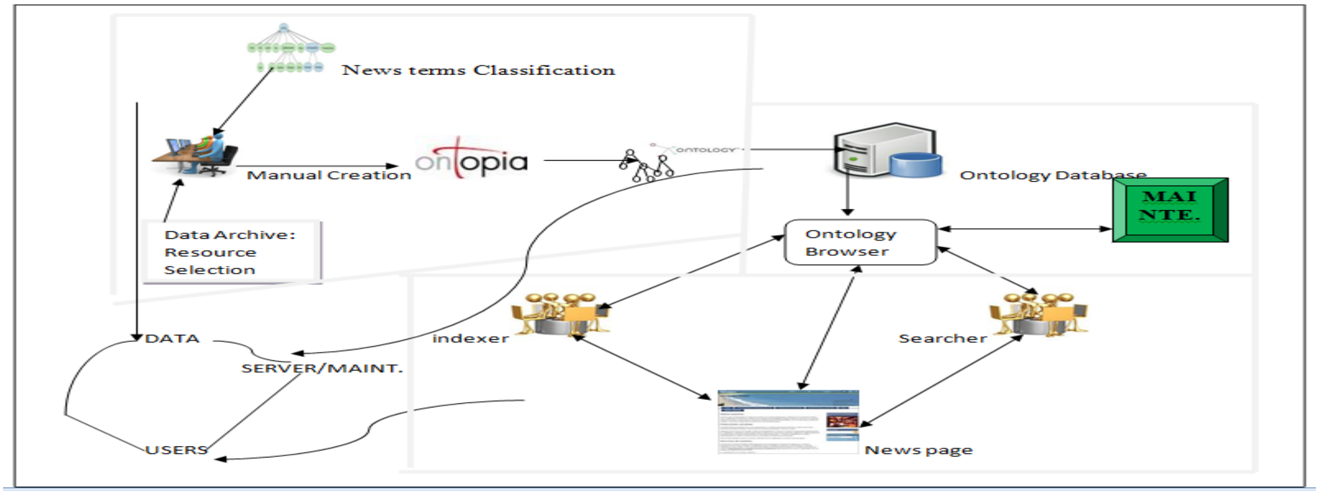


Figure 2: System Architecture

5.1 Data Tier

Data tier consists of developing XTM model manually or what could be described as building an XTM-based ontology. This tier is accomplished by converting a certain domain of (e.g. the news) ontology model and data selection of information resources required for every topics involved in the domain into XTM using the Ontopia-5.1.3 application. The XTM-based ontology is stored at the topicMaps (tomcat) server of the ontopia which acts as database. The tier considered and analyzes all the required data for developing the XTM.

5.2 Server/Maintenance Tier

The Ontopia software is a very robust (J2EE) application. Its runs on Tomcat Web Server and the server equally acts as database for the XTM developed in the first tier. The server is like a container where you retrieve every stored data accordingly "garbage in, garbage out". The XTM-driven ontology is browsed by a built-in application of Ontopia called Omnigator which enable users to browse for information in the domain.

XTM at this point needs to be maintained to resolves topic(s) that does not has semantic similarity with the topics that form user profile, in other words dead links among selection of resources. The maintenance is done by a platform called Ontopia Navigator Framework (ONF). In this framework, the ontology can be browsed via any client-side web browsers (Internet Explorer, Firefox, Google Chrome etc).

Therefore, the proposed system was implemented by connecting existing XTM-based ontologies with the system thereby resulting into a user web application in the third tier of the architecture. It is important to note that web pages written using the navigator framework are written as JSP, and JSP is a Java technology that lets you write web pages as HTML with embedded Java code.

5.3 Users Tier

The Users formed the third (final) tier of the architecture. At this tier, the system would be accessed and navigated by users which are categorized into two: indexers and information searchers (as represented in Figure2); representing various levels of knowledge and different ways for accessing the system. Besides, the system would first check user's computer network connectivity status – weather is connected or not and finally reports any dead link to user.

6. RESULTS AND DISCUSSION

The proposed system once incorporated with XTM-based applications has the capability to identify dead links. To this end, Table 2 presents the results of the XTM-based News ontology developed as case study and existing XTM-based bio-informatic systems with dead links as a result of the maintenance program incorporated.

Table 2: Comparative Results of XTM-based Systems and the maintenance program incorporated.

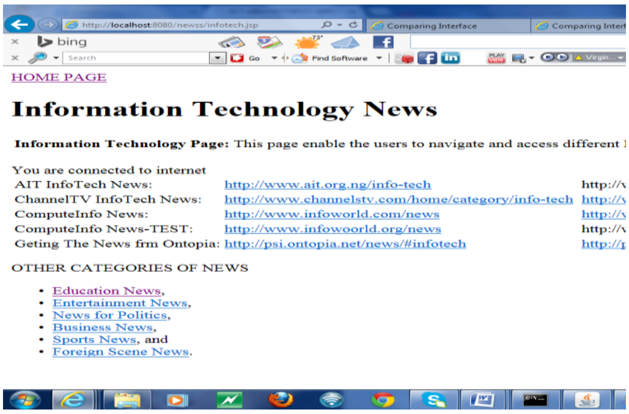
	Output without using Maintenance			Output using Maintenance		
	Dead links	Total Links		Dead links	Connecting Links	Total Links
Bio-informatic XTM-based System	0	56		54	2	56
Proposed NEWS.net (just a topic: infotech)	---	---		2	3	5

The existing XTM-based system had no maintenance program for XTM developed. Since the XTM are manually created and information resources that users access cannot be said to be stable at all time thus, the possibility of dead links is always high. And that is the essence of maintenance program - to identify dead links. For example; in Table 2, the existing XTM-based system has fifty six (56) links and none is identify to be dead until user discovers it while accessing the system. However, same system is executed with the maintenance program, it was discovered that only two (2) links are readily available for use while fifty four (54) are dead.

The same mechanism was also applied to the information technology (infotech) topic of the proposed News.net system, but since the application has the maintenance program incorporated, the output without using maintenance would, of course, automatically be nil while that of output with maintenance would definitely generate result. It has a total of five (5) links but instantly identifies to user that two (2) are dead, thereby left with three (3) other links to use. This saves the user's time from accessing the two dead links.

To this end, the proposed system has greatly mitigated the time taken by user who intends to navigate and access information on XTM-based systems.

Figure 3 shows the result of navigating Infotech news of the News application. The system instantly reports the dead links. Figure 4 represents the dead links output of bio-informatics XTM-based system.



HOME PAGE

Information Technology News

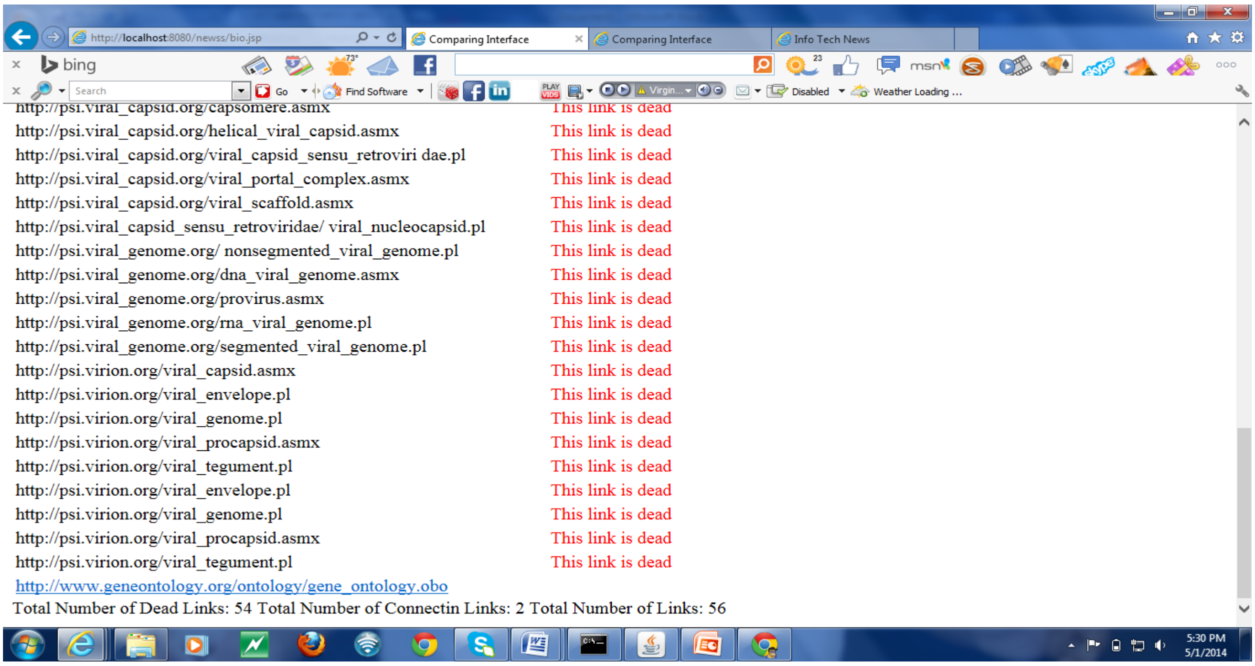
Information Technology Page: This page enable the users to navigate and access different

```
You are connected to internet
 AIT InfoTech News:              http://www.ait.org.ng/info-tech                        http://v
 ChannelTV InfoTech News:        http://www.channelstv.com/home/category/info-tech      http://v
 ComputeInfo News:               http://www.infoworld.com/news                          http://v
 ComputeInfo News-TEST:          http://www.infowoorld.org/news                         http://v
 Geting The News frm Ontopia:    http://psi.ontopia.net/news/#infotech                  http://p
```

OTHER CATEGORIES OF NEWS

- Education News,
- Entertainment News,
- News for Politics,
- Business News,
- Sports News, and
- Foreign Scene News.

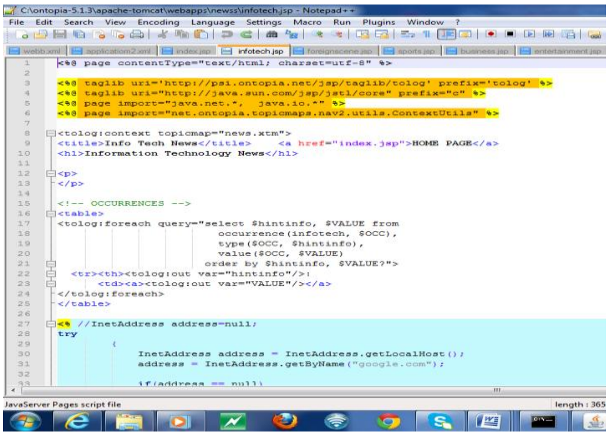
Figure 3: Result of (Dead Links) URLs in the infotech News of NEWS.net System



```
http://psi.viral_capsid.org/capsomere.asmx                          This link is dead
http://psi.viral_capsid.org/helical_viral_capsid.asmx               This link is dead
http://psi.viral_capsid.org/viral_capsid_sensu_retroviri dae.pl     This link is dead
http://psi.viral_capsid.org/viral_portal_complex.asmx               This link is dead
http://psi.viral_capsid.org/viral_scaffold.asmx                     This link is dead
http://psi.viral_capsid_sensu_retroviridae/ viral_nucleocapsid.pl   This link is dead
http://psi.viral_genome.org/ nonsegmented_viral_genome.pl           This link is dead
http://psi.viral_genome.org/dna_viral_genome.asmx                   This link is dead
http://psi.viral_genome.org/provirus.asmx                           This link is dead
http://psi.viral_genome.org/rna_viral_genome.pl                     This link is dead
http://psi.viral_genome.org/segmented_viral_genome.pl               This link is dead
http://psi.virion.org/viral_capsid.asmx                             This link is dead
http://psi.virion.org/viral_envelope.pl                             This link is dead
http://psi.virion.org/viral_genome.pl                               This link is dead
http://psi.virion.org/viral_procapsid.asmx                          This link is dead
http://psi.virion.org/viral_tegument.pl                             This link is dead
http://psi.virion.org/viral_envelope.pl                             This link is dead
http://psi.virion.org/viral_genome.pl                               This link is dead
http://psi.virion.org/viral_procapsid.asmx                          This link is dead
http://psi.virion.org/viral_tegument.pl                             This link is dead
http://www.geneontology.org/ontology/gene_ontology.obo
Total Number of Dead Links: 54 Total Number of Connectin Links: 2 Total Number of Links: 56
```

Figure 4: Screenshot of Bio-informatics XTM-based System's Dead Links Output

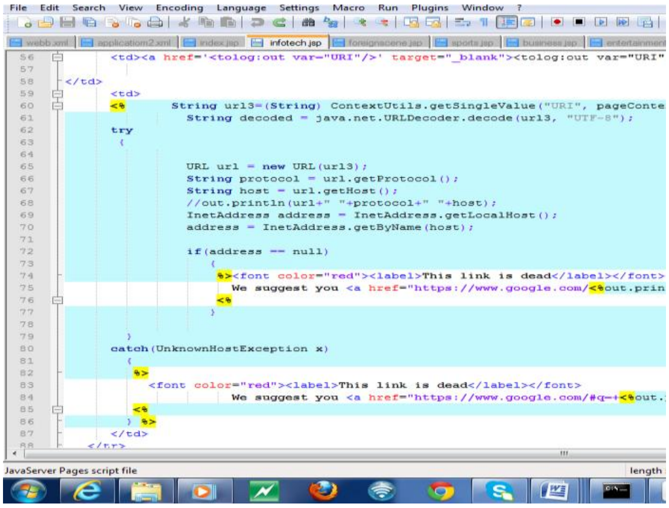
Figure 5 represent the code fragment to check if the user's computer system is connected to Internet. The first four lines of the codes highlighted are the set of the framework tag library and that of java to import classes.


C:\ontopia-5.1.3\apache-tomcat\webapps\newss\infotech.jsp - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
webb.xml
applicatiom2.xml
index.jsp
infotech.jsp
foreignscene.jsp
sports.jsp
business.jsp
entertainment.jsp
JavaServer Pages script file
length : 365

```
<%@ page contentType="text/html; charset=utf-8" %>

<%@ taglib uri='http://psi.ontopia.net/jsp/taglib/tolog' prefix='tolog' %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page import="java.net.*,  java.io.*" %>
<%@ page import="net.ontopia.topicmaps.nav2.utils.ContextUtils" %>

<tolog:context topicmap="news.xtm">
<title>Info Tech News</title>    <a href="index.jsp">HOME PAGE</a>
<h1>Information Technology News</h1>

<p>
</p>

<!-- OCCURRENCES -->
<table>
<tolog:foreach query="select $hintinfo, $VALUE from
                        occurrence(infotech, $OCC),
                        type($OCC, $hintinfo),
                        value($OCC, $VALUE)
                      order by $hintinfo, $VALUE?">
  <tr><th><tolog:out var="hintinfo"/>:
      <td><a><tolog:out var="VALUE"/></a>
</tolog:foreach>
</table>

<% //InetAddress address=null;
try
        {
            InetAddress address = InetAddress.getLocalHost();
            address = InetAddress.getByName("google.com");

            if(address == null)
```

Figure 5: JSP codes to check Computer Network Connectivity.

Figure 6 represents the codes fragment that do the job. That is, the maintenance program that identifies dead links.


File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
webb.xml
applicatiom2.xml
index.jsp
infotech.jsp
foreignscene.jsp
sports.jsp
business.jsp
entertainment
JavaServer Pages script file
length

```
      <td><a href='<tolog:out var="URI"/>' target="_blank"><tolog:out var="URI"

</td>
      <td>
      <%        String url13=(String) ContextUtils.getSingleValue("URI", pageConte
                  String decoded = java.net.URLDecoder.decode(url13, "UTF-8");
      try
       {


                  URL url = new URL(url13);
                  String protocol = url.getProtocol();
                  String host = url.getHost();
                  //out.println(url+" "+protocol+" "+host);
                  InetAddress address = InetAddress.getLocalHost();
                  address = InetAddress.getByName(host);

                  if(address == null)
                      {
                       %><font color="red"><label>This link is dead</label></font>
                         We suggest you <a href="https://www.google.com/<%out.prin
                       <%
                      }


          }
      catch(UnknownHostException x)
          {
           %>
              <font color="red"><label>This link is dead</label></font>
                         We suggest you <a href="https://www.google.com/#q=+<%out.
           <%
          } %>
      </td>
    </tr>
```

Figure 6: JSP codes to check Dead links

7. CONCLUSIONS

This paper focuses on how to identified dead links (URLs) in an XTM-based systems. To detect or identify dead URLs in the system, XTM as a medium for developing the ontology in this paper needs to be maintained. The XTM-based ontology system was designed using Ontopia; an ontology management application. The choice of ontopia in this research work is greatly influenced as a result of its abilities to save ontology of other technologies (RDF/LTM), good graphical visualization and above all, the built-in (ONF) framework. Hence, the proposed maintenance program on this research work was designed using the ONF mechanism. The mechanism is J2EE-based that in essence supported JSP.

The proposed mechanism is incorporated to some XTM-based systems and results were shown to prove its effectiveness. However, before the proposed system detects any presence of dead links; the system has the capability to check if the computer systems are connected to the internet. Instead of users detecting URLs that are longer available to access, the proposed system would automatically detect and reports them on its own. Therefore, the system has greatly mitigated the time taken by user who wants to navigate and access the systems.

8. RECOMMENDATIONS AND FURTHER STUDY

The significance of this mechanism to resolves dead links which may arise from unmaintained XTM cannot be overstressed, thus it is recommended for XTM, LTM or RDF developers in any area of knowledge and to a great extent, the system can identify dead links by itself.

In addition, since the NEWS.net is an XTM-based application and XTM is a topic map defined in XML syntax of W3C, then XTM supports structuring organization's documents or even make an organization's application talk to other applications. Also, in line with the versatility feature of the Ontopia Software Application, XTM document can be exported to RDF or LTM format and vice versa thereby, making it easier for RDF/S developers to use.

XTM's sensitive element (e.g., occurrence) is largely created by domain expert and consumes a lot of efforts. Future research should focus on automatically creating and maintaining such element of the knowledge base by interfacing the knowledge base to support a semantic search engine. Besides, the improvement of the knowledge base, the ranked semantic similarity model can be improved as well.

9. REFERENCES

[1] Yi, M. (2008). "Information Organization and Retrieval Using a Topic Maps-Based Ontology: Result of a Task-Based Evaluation", Journal of the American Society for Information Science and Technology 15(12): 1898 – 1911.

[2] Gruber, T. R. (1993). "A Translation Approach to Portable Ontology Specifications" Knowledge Acquisition, 5(2): 199-220.

[3] Ontopia, (2010). Omnigator: The Topic Map Browser, Retrieved September, 27, 2013. http://www.ontopia.net/omnigator/docs/navigator/userguide.html

[4] Park, J. and Hunting, S. (2002). XML Topic Maps: Creating and Using Topic Maps for the Web. Addison Wesley, Boston, U.S.A, 79 – 95.

[5] Librelloto, G. R., Ramalho, J. C., and Henriques, P. R., (2004). TM Builder: An Ontology Builder based on XML Topic Maps. CLEI Electronic Journal, 7(2).

[6] Biezunski, M. and Newcomb, S. R., (2001). XML Topic Maps: Finding Aids for the Web. In: Leu, P. (Ed) Siemens Corporate Research. IEEE Multimedia, 104 – 108.

[7] Garshol, L. M. (2004). Metadata? Thesauri? Taxonomies? Topic maps! Making sense of it all. Journal of Information Science, 30(4), 378–391.

[8] Pepper, S. and Moore, G. (2001). XML topic maps (XTM) 1.0. Retrieved July, 14, 2012, from http://www.topicmaps.org/xtm/1.0/

[9] Pepper, S. (2006). The case for Published Subjects. Retrieved September, 27, 2013, from http://www.ontopia.net/topicmaps

[10] Wu, Y. and Dunaway, D. J. (2013). Creating a large topic maps by integrating Wandora and Ontopia. Library Hi Tech/Emerald 31(1) 64-75

[11] Husáková, M. and Olševičová, K. (2008). Creating Web Presentation for Observatory and Planetarium with Topic Maps. In: Maicher, L., Garshol, L. M. (Eds) Subject-centric computing. International Conference held at Liezpig, Germany, 16 – 17 October on Topic Maps Research and Applications, 237 – 246.

[12] Gregoras, D. (2005). Using XTM for navigating UDDI. M.Sc Thesis. Department of Mathematics and Computer Science University of Osnabrück, Germany and Department of Computer Science, University of Twente, The Netherlands.

[13] Wikipedia, (2014). Semantic Similarity. http://en.wikipedia.org/wiki/Semantic_similarity

[14] Ijntema, W., Goossen, F., Frasincar, F., and Hogenboom, F., (2010). Ontology-Based News Recommendation. In Proceedings of the ACM, EDBT held at Lausanne, Switzerland.

[15] Rubbani, H. H. (2007). Semantic Web Solution. IT University of Copenhagen. M.Sc Thesis.

[16] Barta, R. (2005). Topic Maps: Fundamentalism meets Pragmatism. In: Meyer T., Orgun M. (Eds) Australasian Ontology Workshop (AOW), Sydney, Australia. Conference in Research and Practice in Information Technology Vol. 58.

[17] Lauser, B. (2003). Semi-automatic Ontology Engineering and Ontology Supported Document Indexing in a Multilingual Environment.

10. AUTHOR'S PROFILE

Aminu Enesi Femi lectures at Federal University of Technology Minna, Nigeria, Department of Computer Science. He obtains both his B.Sc and M.Sc degrees in Computer Science from University of Jos, Jos and Ahmadu Bello University, Zaria respectively. His current research interest is on ontology languages and semantic web contents.

Adewale Olumide Sunday (Ph.D) is a professor in the Department of Computer Science, Federal university of Technology Akure, Nigeria.