

# Development of a Prototype Public File Repository for XBRL Documents: Challenges and Opportunities<sup>1</sup>

**Janet J. Prichard.** Bryant College. USA.  
prichard@bryant.edu

**Saeed Roohani.** Bryant College. USA.  
sroohani@bryant.edu

**Abstract.** Within the past few years, companies have begun publishing financial reports in an XML-based open-standard file format called XBRL. Providing easy public access to XBRL formatted data of company disclosures would provide investors, analysts, and accounting researchers the data in a highly flexible and usable format. EDGAR currently provides public access to the U.S. Securities and Exchange Commission's required filings for publicly traded companies. Currently, EDGAR filings must be in either ASCII, HTML, and in some select cases, XML format. Though EDGAR has recently started to provide XBRL formatted data to the public for a limited number of companies, their primary distribution method remains in traditional file formats including plain text files, PDF, and GIF. Regardless of difficulties with XBRL specification versioning and customized taxonomies, XBRL data aggregators have yet to address any simple mechanism to retrieve XBRL documents. The purpose of this paper is to discuss a prototype public file repository for XBRL documents and present challenges and opportunities of having this public file repository. Also, in response to an anticipated demand for XBRL documents for the purpose of more frequent reporting and auditing of financial information, we compare two approaches of public access to XBRL documents on the web: file repository and file registry.

**Key words:** XBRL, File Repository, File Registry, Continuous Reporting.

## 1. INTRODUCTION

Within the past few years, companies have begun publishing financial reports in an XML-based open-standard file format called XBRL (eXtensible Business

---

<sup>1</sup> We would like to thank Eric E. Cohen, Walter Hamscher, and Mike Willis for their helpful comments on earlier version of this paper.

Reporting Language). Providing easy public access to XBRL formatted data of company disclosures would provide investors, analysts, and accounting researchers the data in a highly flexible and usable format. EDGAR (Electronic Data Gathering, Analysis, and Retrieval) currently provides public access to the U.S. Securities and Exchange Commission's required filings for publicly traded companies. Currently, EDGAR filings must be in either ASCII, HTML, and in some select cases, XML format (Osheroff *et al.*, 2003). Though EDGAR has recently started to provide XBRL formatted data to the public for a limited number of companies, its primary distribution method remains in traditional file formats including plain text files, PDF, and GIF. Regardless of difficulties with XBRL specification versioning and customized taxonomies, XBRL data aggregators have yet to address any simple mechanism to retrieve XBRL documents. While companies are under no obligation to submit data to EDGAR in an XBRL format, it is anticipated that firms motivated by investor relations will adopt XBRL standards for reporting to compete in the capital markets for resources. The purpose of this paper is to discuss a prototype public file repository to XBRL documents and present challenges and opportunities of having this public file repository.

This paper begins with an introduction to XBRL and file repositories. This is followed by a discussion of the development of the XBRL file repository prototype as an academic project. The goals and user requirements of the system are discussed, as well as the system architecture and implementation. The next section discusses the issues and challenges of providing public access to XBRL data. This includes some discussion of the trade offs between a file repository and a file registry, with a conclusion that a combined approach may be best. The development of the prototype file repository system enabled us to observe opportunities and challenges of providing public access to XBRL data.

## **1.1. Overview of XBRL**

XBRL is an XML-based open-standard currently owned by XBRL International Inc., a non-profit consortium (McNamar, 2003). XBRL is a standard data format based on XML. XBRL taxonomies can be defined to describe financial statements, journal entries, bank call reports, and other information for corporate reporting. This provides management, creditors, investors, accountants, and analysts, access to company financial data in a format that can be utilized by a wide

variety of applications. Bonsón (2001) states that XBRL will also facilitate the communication of information in a homogeneous way, allow the interchange of data between software applications, and allow for the automatic analysis of financial information.

The XBRL Consortium is an international entity consisting of accounting firms, software developers, government regulators, and professional organizations around the world. The Consortium is the authority for approving XBRL specifications and standards. Software developers producing XBRL tools and applications follow such standards. For example, Fujitsu LTD makes an XBRL taxonomy editor, XBRL instance creator, and XBRL validator – these tools get updated as new XBRL specifications are released.

Companies such as Microsoft have developed applications such as “Microsoft Office 2003 Solution Accelerator for XBRL” that takes XBRL formatted data and facilitates analysis of company and financial performance (Hannon, 2004). In conjunction with this Microsoft release, EDGAR (2003) offered sample XBRL files so users could try out the software. The product also facilitated production of XBRL formatted data by users.

XBRL interest groups are now active in jurisdictions such as Japan, United Kingdom, Australia, and Canada. A new byproduct of XBRL is XBRL GL (general ledger) aimed at financial information reporting at the transactional level. A query in XBRL GL can produce information on when and where a business transaction took place. In the future, XBRL GL may simplify and expedite business information exchange among business partners.

One of the most attractive features of XBRL is that it enables interoperability of information (Willis, 2003). XBRL formatted data is easily reusable by multiple applications. This is due to the fact that XBRL is an XML application, and hence XBRL files are easy to transform into standard file formats such as HTML, XML, ASCII, and PDF. XBRL files can also be transformed into application specific formats like Microsoft’s Excel spreadsheet format. These transformations may be done within the application producing and consuming XBRL data, or can be done using external XSLT (eXtensible Stylesheet Language Transformation) files.

Other advantages of XBRL relative to the current business information exchange process have been noted by Willis (2003). These include lower production costs

and more efficient reporting, lower consumption costs and greater transparency, and enhanced information sharing.

It has also been suggested by McNamar (2003) that use of XBRL filings would allow “the SEC to be able to identify deviations from industry norms or standards”. He examines the Enron case, and believes that it would have been easier to compare Enron’s reported revenues, cash flows from operations, and profits against industry standards and that SEC could have flagged for further investigation those values that had high discrepancies from the norms. In essence, XBRL formatted data facilitates such analysis.

## **1.2. File repositories**

Internet-accessible public file repositories are not a new concept. An electronic file repository can be defined as any system that provides users a mechanism for downloading files for later consumption. Public electronic file repositories have been accessible since the early 1980s using FTP. These early sites used public or “anonymous” logins for users to primarily download files to their own system; relatively few sites allowed users to upload files for public distribution. Public repositories that provided users the ability to submit files would typically screen the content before making it available in the repository. These sites often reflected a common theme, for example – a collection of programs or a library of images and icons. The biggest problem with these early systems is that they provided no mechanism for the user to do keyword searching. As a result, programs such as Archie were developed to provide this capability.

To facilitate users browsing and retrieval of information, a menu-based system called Gopher was developed. But as the amount of information available began to increase, the menu navigation provided by Gopher became laborious. And as with FTP, Gopher did not directly support searching, programs such as Veronica and Jughead were later developed to provide the needed searching capabilities.

The advent of the browser revolutionized the way users accessed information on the Internet. The browser provided users an easy to use graphical display that was able to handle text and images. The addition of hypertext links allowed users to more easily navigate between related documents. Web-based systems soon

became available to provide access to public repositories that were previously only available through FTP. This allowed greater availability of information to the average user.

One use of file repositories that was recognized fairly early was the ability for a group of users to share files while working on a common project. But to avoid conflicts, some controls must be in place to allow users to track and merge changes. These systems are commonly referred to version control systems. In essence, they help to track the history of files.

More recent developments include Digital Libraries and Content Management Systems (Roth *et al.*, 2002). Digital Libraries are similar to a traditional library in that they provide access to some collection of information – usually in digitized form. Content Management systems add the ability to annotate data in ways that facilitate the management of the data. For example, in addition to storing a photograph digitally, information about the photograph is also stored. Information about data is often referred to as metadata. Metadata for a photograph may include a description of the photograph, the names of any persons in the photograph, the photographer, and the date it was taken. A Content Manager System not only provides access to the photograph, but it also provides mechanisms that allow a user to search through the metadata to help locate the photograph. Content Management Systems also tend to manage large stores of digital media. Some of these systems require a specialized client application to access the information while others use a standard browser.

Today's file repositories typically provide a varied range of capabilities including:

- Storage and retrieval of files
- Authentication and access control
- File check-out/check-in
- File versioning (often with ability to see file differences)
- Storing file metadata (file properties, source, history, etc.)
- Searching
- Integration with desktop applications such as e-mail
- Encryption
- Secure transmission of files

An XBRL file repository also has the potential to offer other benefits. First, given the ease that XBRL data can be transformed to other formats, the file repository could provide users the ability to download XBRL files in a variety of formats automatically generated by the system through the use of XSLT files. Standard views of the data could be provided, for example, an individual investor may be interested in only seeing a summary of a company's financial statement. The XSLT file can extract content suitable for an investor and allow their application – for example, an Excel spreadsheet – to read it. A second benefit of creating a file repository is that it provides researchers XBRL formatted data in a single location for applications such as data mining. An interesting area of research that is emerging is data mining of XML formatted data.

During our development of an XBRL file repository, it became clear that another mechanism should also be considered in providing public access to XBRL documents – a file registry. Our view of the file repository was that in addition to storing all of the metadata associated with an XBRL file (such as its source, date of capture, etc.), a copy of the actual file would also be maintained. A file registry contains all of the same metadata as a file repository, but makes no local copy of the actual XBRL file. A further discussion of file registries will be presented later in this paper.

### **1.3. Current availability of XBRL files on the web**

The primary motivation for creating a public XBRL repository was to provide a centralized location for accessing available XBRL files. Currently, though many companies are starting to adopt XBRL standards for financial reporting (Weisel, 2002), only a few of these XBRL files are readily available. Another challenge is that there is no standard location for these files on a company's web site. Thus, it is still somewhat difficult to locate XBRL formatted data on the web. A recent search for XBRL documents on the web using the Google search engine produced the results shown in Table 1.

The first column of Table 1 contains the search phrase as it was entered into Google. Quoted strings allow searching for phrases. The OR keyword allows the search to match on either of the words used in the OR clause. The second column shows the total number of hits the search produced. The third column represents

the number of sites within the top thirty sites for each search that have XBRL documents directly referenced on the page. It is interesting to note that only four different companies appear in the top thirty listings, with three of them providing access to recent XBRL documents. Table 1 also demonstrates that there are significant writings and discussions about XBRL in the last several years. While there seems to be an increase in interest in XBRL, there are limited sites posting XBRL documents.

<b>Search phrase</b>	<b>Total returned</b>	<b># Sites with XBRL</b>
XBRL files OR file	12,400	2
“XBRL files”	101	2
“XBRL file”	66	2
XBRL data	30,400	2
“XBRL data”	672	2
XBRL instance document OR documents	2,450	0
XBRL “instance document”	502	0
XBRL-Formatted Financial Information	133	1
XBRL Formatted Financial Information	587	1
XBRL financial statement	6050	0
XBRL “financial statement”	1,750	1
XBRL financial report	15,000	1
annual report XBRL	10,300	2
“annual Report” XBRL	7,910	2

Table 1. Results of google searches for XBRL phrases

The XBRL Express website, powered by EDGAR Online, also provides access to XBRL documents for a limited number of companies. Users must initially select an industry category from a list, then select a company within that category. The user is then presented with a standardized list of text files and XBRL files including a balance sheet, cash flow statement, income statement, and company taxonomy. Upon viewing the source code of these documents, it becomes evident that EDGAR has automatically generated these files from required filings. For a fee, users can get a premium subscription from EDGAR Online that will provide them access to over 12,000 U.S. publicly traded company filings in XBRL.

Financial Reporting and Auditing Agent with Net Knowledge (FRAANK) is a prototype that fetches information from the SEC (Securities and Exchange Commission) EDGAR site and converts it into XBRL (Bovee and Srivastava,

2001). At this time, there are two programs available on the FRAANK website for converting portions of text-based 10Q or 10K financial statements. Each program allows users to select the company of interest, and have returned hypertext links to the raw filings or a collection of XBRL documents. However, this prototype was developed based on an earlier version of the XBRL specifications.

Recently, Microsoft (2004) released the “Microsoft Office Tool for XBRL™ Prototype”. This software makes it possible to use Microsoft Office Word 2003 and Microsoft Office Excel 2003 to directly create and analyze documents in XBRL format. It also provides users a built-in connection to EDGAR Online. With the use of a demonstration username and password, users can download free financial statements for a sample of the S&P 500. If users wish to have access to a larger number of companies, they must pay for the premium subscription to EDGAR Online mentioned above.

## **2. DEVELOPMENT OF THE XBRL FILE REPOSITORY PROTOTYPE**

### **2.1. Functional goals and user requirements for public XBRL file repository**

A prototype public XBRL file repository was created as a project in a graduate web development course by one of the authors. The initial functional goals for this system were as follows:

1. Provide a centralized location that could be used to gather information on the location of XBRL files on the web.
2. The system locates the XBRL files from information provided by the user and downloads a copy of the XBRL file to the repository (creating the centralized store of the files on the server)
3. The system should also maintain metadata (information about file properties, source, history, etc.) related to each XBRL file stored in the repository

In addition to the above goals for building a system, the following user requirements were considered in the initial system development:

1. Provide a simple to use interface for users to report the location of XBRL files on the web.
2. Provide an interface that allows users to download XBRL files from the repository.
3. Provide users with a way to request data using one or two standard XSLT files. This capability can help to demonstrate the flexibility of XBRL formatted data.

A user's view of the repository is shown in Figure 1. Users either submit XBRL file locations to the repository or retrieve files from the repository. When a user submits an XBRL file location, the system attempts to locate the XBRL documents and extract them from the company website. Metadata associated with the XBRL files is also stored. A user may also request files from the repository, if the files are present, they are returned as linked documents to the user.

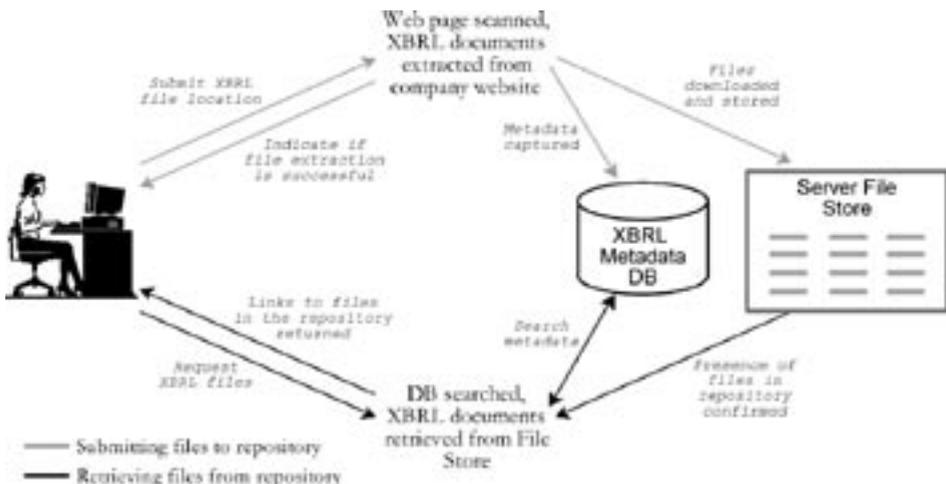


Figure 1. User's view of XBRL repository

## 2.2. System architecture

The XBRL repository system utilizes a simple three-layer architecture, including an application layer, business logic layer, and a storage layer, each with an Application Program Interface (API) to facilitate future enhancements (see Figure 2). The storage layer is implemented using a relational database for the metadata associated with each XBRL document, and the local file storage system

is used to store a copy of the XBRL document in the repository. The business layer is where the XBRL repository-specific functionality is located, including the “Submit” and “Browse” modules. Future modules (written in gray italic in Figure 2) are also specified here for functionality such as archiving and versioning, administrative tools, authentication, and enhanced searching. Each module also has an API specified to facilitate replacement or enhancement of a function as desired. Finally, the application layer covers the interfaces to the system: a web interface (browser) and web services (e.g., to support interoperation with other systems). Willis (2003) has mentioned the use of web services as another mechanism for providing access to XBRL documents.

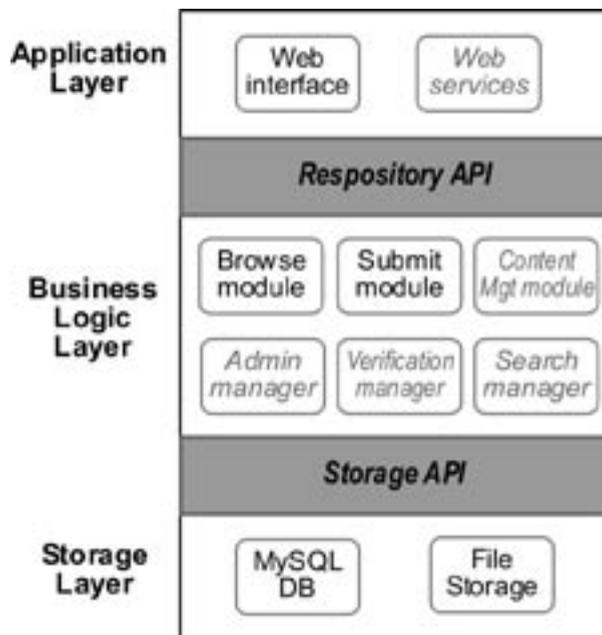


Figure 2. System Architecture

The term web services is often used in many different ways. One interpretation is for services that are web enabled. Another, more technical interpretation, is new web applications that use a combination of technologies, including XML, to provide communication and data services between two seemingly disconnected systems. In web services environment, Universal Description, Discovery, and Integration (UDDI) is the specification that enables information customers to find services in its registry and return results on how to contact the XBRL data service (source). Web Services Definition Language (WSDL) is the language with which various

data services are registered with the UDDI. Once the service has been located, the information customer's application uses Simple Object Access Protocol (SOAP) to request the company's XBRL document. SOAP is necessary for exchange of XML formatted data. In addition to responding to the information customer with XBRL documents, web services can provide additional services such as authentication, fees, etc. To be effective and successful, web services rely on universally accepted rules, specifications, and standards. It will be interesting to see how this evolves in making XBRL documents available to the public.

### 2.3. Implementation

The first task was to design and build the submit module that can be used to gather information about the location of XBRL files on various company web sites. We developed a simple web interface to the submit module to facilitate the gathering of information from users (anyone – public access) about the location of XBRL documents on the web.

The primary task of the submit module is to process a web page containing the links to the XBRL documents on the company website, and then to extract all XBRL document linked to that page. To require a user to provide links for each of the XBRL documents from a single source is cumbersome. Our experience has been that most of the web sites that provide access to XBRL files do so by having the documents listed on a single page. Figure 3 shows the form that users use to submit the URL of the page that summarizes the XBRL links.

The screenshot shows a web form titled "XBRL Public File Repository" with a subtitle "Collect XBRL File Location Information". Below the title is a grey bar with the subtitle. The form contains the following elements:

- A heading: "Please enter the company information and the location of the XBRL files in the form below."
- Four input fields:
  - Company: [text input]
  - Stock Code: [text input]
  - Company Website: [text input]
  - XBRL Links Page: [text input]
- Two buttons at the bottom: "Submit Information" and "Reset Form".

Figure 3. Web interface for submit module

The submit module parses the “XBRL Links Page” for URLs to XBRL documents that are assumed to adhere to XBRL standards. These URLs can usually be identified by the fact that they have the .xml filename extension. For example, a typical link to an XBRL document on the Microsoft web page may look as follows in the HTML source code:

```
<a href="/msft/content/XBRL/MSFTv1_10Q02_3.xml">
  Q02-3 10-Q
</a>
```

From this link, the following information is extracted:

*XBRL document name:* Q02-3 10-Q

*XBRL document location:* /msft/content/XBRL/MSFTv1\_10Q02\_3.xml

In this case, resolving this relative URL yields the actual location of the XBRL file at

```
http://www.microsoft.com/msft/content/XBRL/MSFTv1_10Q02_3.xml
```

Once this data has been collected, the metadata about the XBRL file (collected from the form submitted by the user and the system itself) is stored in the relational database. The actual XBRL file is then downloaded and placed in the server’s file system.

The second task was to design a browse module. Currently this portion of the repository is not very sophisticated; it only allows users to search for files in the repository using the company stock code or to choose the company from a pull down menu of companies in the repository as shown in Figure 4.

**XBRL**  
**Public File Repository**

**Retrieve XBRL Files from Repository**

To view the XBRL documents available in the repository for a particular company, please enter the stock code or use the pull down menu.

Stock Code  or

Figure 4. Web interface for browse module

If files are present in the repository, the user is presented with an opportunity to download the XBRL files as shown in Figure 5.



Figure 5. XBRL file download page

The last task, providing users with the results of using XSLT file (this file allows users to view all or part of XBRL data in HTML, Excel, or text) format located on the server, is currently under development. This effort will demonstrate the feasibility of implementing such a feature as part of the repository.

### 3. CHALLENGES AND ISSUES

#### 3.1. Issues directly related to prototype development

##### 3.1.1 Submit module

In developing the submit module, a few issues came to our attention. First, a mechanism must be provided to allow more than one web page to be submitted for each company. As noted, most of the sites that we found had posted these files through a single page, but clearly as more XBRL data becomes available, it is likely that this information will be spread across numerous pages.

Second, sometimes XBRL data is embedded within other file types such as HTML. In these cases, the submit module may overlook these files since it is only designed to search for files that end with the `.xml` suffix. It would be helpful to have `.xbrl` or `.xbr` as file extensions for XBRL documents. Third, files are not

always statically linked to a site. Sometimes they are the result of a dynamic query generated from a user request. This particular situation was not encountered during our development, but must be recognized as a distinct possibility.

Lastly, because no standard naming scheme exists for XBRL files (e.g., financial statements, 10-K Form, 10-Q Form), the system will utilize the name of the document as provided by the company on its web site. Clearly, a standard naming scheme would be helpful. Given that each XBRL document contains information about what it represents, it should be fairly easy to have the system parse through the document and look for a standard name for the document from this information.

### **3.1.2 Browse module and file storage**

The Browse Module and File Storage components provide the main functionality of the repository – providing access to a collection of XBRL documents. The obvious deficiency with the current browse module is that access to the files is limited to searching by stock code. Clearly a more sophisticated interface for file retrieval should be developed. But part of the difficulty here is deciding what criteria users would like to use in retrieving files from the system. An empirical study of users' preferences could help to determine and establish a set of likely criteria to retrieve files.

In the current implementation, the files are currently stored in the local file system on the server. This means that the data stored within the files is basically unavailable for use as search criteria. The alternative would be to store the contents of the XBRL files within the database itself. The data can be stored in a relational database, where it is usually indexed in some manner to make searching the data faster and more efficient. Typically, there are two solutions for storing XML files (XBRL in this case) to a relational database system. The first solution stores the entire document as a single large character object. But this complicates searching of the data (it is treated as one long string), because some of the meaning gained from the structure of the data is no longer available. The second solution attempts to map schema elements to rows. But this in itself is a difficult problem, because some XML files have very vertical hierarchical structures, which do not lend themselves well to row and column translations.

With the emergence of XML, native XML databases (NXD) have become more commonplace. These database systems use an internal model based on XML, not the traditional relational model (Software AG, 2003). In addition to the ability to store data, this feature also allows the database to maintain information related to the structure of the data. The standard query language SQL was designed generally to work with the relational model, and thus has limitations in dealing with XML data (Liotta and Preimesberger, 2003). Alternative query languages such as XQuery exist specifically for XML databases, but this language is procedural in nature (SQL is declarative), and as such, is considered more difficult to learn.

### **3.1.3 Providing file transformations**

The browse module is also under revision to support a user request for a XSLT transformation. Using this as a method of providing users different file formats is under study. Developing XSLT that supports needs of various users and type of information requested is not an easy task, particularly if different XBRL specifications are employed, and there are various customized taxonomies. XSLT files are being integrated into the system and we are experimenting with the user having the ability to specify that a certain XSLT file be applied before the data is made available for download.

One of the open issues in this portion of the development is whether or not users would find this feature useful. Willis (2003) mentions an interesting demonstration of XBRL-enabled reporting can be viewed at <http://www.nasdaq.com/xbrl>. In this demonstration, the user selects an industry (only two similar industries available) and one or more companies from 21 firms available to view the data in Excel Spreadsheet and perform financial and/or ratio analysis. The original data on these firms has been re-formatted to fit a single XSLT style sheet. This demonstration benefits from having two similar industries and similar companies, however, it would be difficult to repeat this experiment with customized taxonomies such as those for banks and healthcare industries. The new trend towards creation and recognition of customized taxonomies might complicate cross-industry data analysis; for example, if a user is interested in liquidity data of firms in a certain region or jurisdiction.

In another demonstration, the site <http://www.xbrlforearningsreleases.com> allows users to view XBRL data of three companies in Microsoft Word and Excel. Alternatively, users can download instance documents and extension taxonomies

to build their own style sheet to view XBRL data for analysis. Thus, assuming that finding XBRL documents is facilitated by the proposed system architecture in this paper, it seems public users expect some kind of file transformation facilities.

### **3.2. Other challenges – data integrity**

One major concern in a system such as this is the integrity of the data – in this case, the XBRL files. There are at least three major issues:

1. Is the data provided by the user accurate?
2. Do the XBRL files conform to the various standards?
3. What is the integrity of the data maintained by the system?

The first issue addresses the need to verify that the information about the location of XBRL files submitted by the user is really representative of an actual company. At the moment, any user may submit this information, but none of the data becomes publicly available in the repository until the source has been verified manually. Other methods doing this verification are currently under study. One solution would be to require companies themselves to furnish this information, but it is not clear what incentive they would have at this time except that it is a show of good faith to the investment community. In theory, to benefit from opportunities offered in the capital markets, firms have incentives to provide timely and accurate financial information to potential investors and that XBRL may facilitate the supply and demand for financial reporting on web (Hunton *et al.*, 2003). Also, it is possible that auditing firms to participate in providing measured assurances on XBRL documents - this type of activities opens new opportunity for auditors to provide assurances over entire financial statements or at the account level using XBRL documents (Cohen *et al.*, 2003). As a result, a resource such as a data aggregator firm, another third party or a joint effort activity that includes auditing firms should be interested in becoming a repository to help users.

The second issue addresses the conformance requirements of XBRL files. Two levels of conformance can be discussed here. First, is to ensure that the file conforms to the XBRL schema. The ability to provide such a service has been demonstrated by Tam (2003). This conformance test verifies that the file is using the syntactic elements of XBRL in a proper fashion. The submit module is currently under revision to verify the XBRL document against the XBRL Schema to make certain

that the data is in the proper XBRL format. If it is not, a message is returned to both the user and the XBRL repository administrator indicating the problem status of the document. The second level of conformance is to check if the file conforms to an authorized taxonomy, particularly given the recent proliferation of customized taxonomies. The taxonomy contains several sections that correspond to typical financial statement components (Weisel, 2002). If an XBRL file is identified and located but does not validate against authoritative taxonomy and its definitions, is no longer reliable for data analysis or use because users may be rolling up (or adding) incompatible elements of financial statements, for example, adding Intangibles to Accounts Receivable. Thus, the system architecture should inform users of such difficulties about the XBRL document as found on the web.

The last issue and perhaps most important issue is related to the integrity of the metadata and the locally stored copy of the XBRL file. There are two major issues that come into play here:

### **Is the XBRL data “fresh”?**

Recall that when a user submits a location for XBRL documents, the system scans the site for XBRL files and gathers data. What happens if the site modifies the names of the XBRL files, changes the contents of the XBRL files, or add new XBRL files to the site after this scan? Also, when a copy of the file is downloaded to the repository, how can we assure that this file represents the latest version of that file at the time it is requested by a user?

One solution to this problem would be to crawl the corporate web sites in a fashion similar to that done by search engines. This would be done on a periodic basis, with information provided to the user as to the last date that updates were applied from the corporate web site. This would also simplify the process of submitting locations to the system and perhaps find other sources of XBRL files that might not have been reported to the system otherwise. The disadvantage would be the amount of processing time required to keep the repository current, especially if large websites, such as Microsoft, have to be frequently scanned.

### **Is the file a current representation by the company?**

This is clearly an issue that differentiates a file repository from a file registry. In the case of a file registry, the user is directed to the corporate website to download

the file. If the file has been cached locally, as in the file repository, the end user should have a mechanism for verifying that the file has not been modified in any way – for example, remains untouched by third parties. Currently this has not been an issue for data aggregators such as EDGAR Online, Inc. This is where techniques such as digital signatures and encryption (private key/public key) could be useful in verifying the authenticity of the data.

There are also issues with respect to the authenticity of the financial data in electronic format (Weisel, 2002). Has the information been audited by independent accountants? Is there any way for the user to distinguish audited from unaudited data? These are issues that are beyond the scope of this work; yet need to be addressed by the XBRL community. One solution is that XBRL documents, as posted by a company, can be required by law, or auditors to be an authenticated copy that contains all vital and audited financial information. If the repository provides users with the information about the source of the documents, users will always have the option of retrieving the file from the original location on the company website using XBRL file registry mechanism. At this time, there is no clear mandate by law or auditors whether XBRL documents or financial information on the web site should be authenticated.

### **3.3. Tradeoffs of a file repository versus a file registry**

As mentioned earlier, once the effort to develop a repository was underway, it became clear that a file registry should also be considered as a solution. The primary difference between a file registry and a file repository is that a file registry makes no attempt to store the XBRL documents locally. A file registry presents users with links to the XBRL documents as located on the original company web site. Users can then download the documents directly from the source. Hence, the only changes to the functional goals presented earlier would be that goal #2 would no longer be supported - the XBRL files would not be downloaded to the repository.

Thus, a file registry appears to be a good way to address some of the logistical problems of maintaining a repository of XBRL files. Below is a brief summary of what we see as the tradeoffs between the two systems:

**File availability.** A file repository allows the files to be stored in local file systems, so the files are readily available. As long as the file repository is available, its content can be access and viewed. Using a file registry, you are dependent on the company web site for the file availability. Also, if file validation capability is required in the file registry, the XBRL documents would have to be downloaded temporarily to perform the various validations (schema and/or taxonomy) on the files.

**Searching capabilities.** When using a file repository, you have the option of storing the data in a database system. A richer set of searching constructs can be employed and applied over a wide range of files. Even if the files are stored in a file system, primitive searching capability can be provided. It's not clear at this time if a file registry could support any type of searching capability given that the XBRL documents are not readily available (not stored locally).

**File transformations.** The process of providing users with data that is a result of an XSLT transformation will be easier and faster when a file repository is used. The XBRL file that the transformation is applied to is already resident in the system. In a file registry, providing users with transformed data would require the file to be downloaded temporarily so that the file could be transformed and then made available to the user.

**File versioning.** If a company provides multiple versions of the same file, the repository can maintain the various versions of the file to accommodate historical data analysis. A file registry could not accomplish this unless the company web site provides access to the previous versions of the XBRL files. However, if only recent postings are available, this is still suitable for analysis of current financial and market data about the company. Obviously, data aggregators most likely will follow the repository approach. It should be noted that even with repository approach, commercial data aggregators do customize the XBRL data, as they have done to company data, for general public use. If one day government regulators such as U.S. Securities and Exchange Commission (SEC) mandate XBRL reporting, then there will be less need for customization of aggregated data by commercial data aggregators.

**Data integrity.** A file repository must provide users a mechanism that assures the integrity of the files in the repository. This is also an issue with a file registry,

but a less substantial one, since the user accesses the XBRL file from a known corporate web site directly, and has direct access to the source of the XBRL data. One may also question whether a user should worry about the authenticity of the XBRL files from the company site – e.g., whether it is audited or not.

**Data freshness.** A file repository must provide a mechanism to keep track of the “freshness” of the data and report it to the user. Assuming XBRL file properties exist, when using a file registry, the user is getting the data from the source, and can make his/her own judgments regarding the freshness of the data.

Of course both a file repository and file registry suffer the problem of files being moved around the company web site, and determining which files have already been registered and made available to users of the system. Also, if a user want to download fourth quarter 2003 earnings statements from two different companies, we can't rely on the filenames alone to identify these documents. As mentioned previously, further scanning of the XBRL document for information such as company name, date, type of report, etc. will be necessary to determine what data is actually contained in the document. This paper recommends a set of standards reflecting file properties and naming to be developed by the XBRL community and observed in both file registry and file repository systems.

Perhaps the best solution is in essence a combination of the two strategies - to mimic the behavior of the popular search engine Google. Like a file registry, Google present users with links to the original web pages returned by a search, but the search results are based upon cached versions of the web pages – that are made available to the user through a “Cached” link. This particular combined strategy allows the repository to deliver a fresh copy of the XBRL file from the corporate web site, yet allows searching and other applications such as data mining to take advantage of storing the files on the local server. The modular system architecture design of the file repository allows for this modification with minimal effort.

The file repository and file registry approaches have implication for continuous reporting of financial information and continuous auditing. In recent years, investors and government regulators emphasize the importance of more frequent disclosure of financial information to the public. XBRL has the capability of offering real time, frequent reporting of financial information. Even if we assume that firms

are prepared and willing to provide such information on a more frequent basis to the public, there have been limited mechanisms available to provide investors and creditors a standard way of identifying, locating, and retrieving such information on a company web site. Users of financial information should be able to retrieve XBRL documents in a timely fashion, with ease and confidence. This was the primary motivating force for the development of the file repository system developed in this work.

## **4. CONCLUSIONS**

Thus far, we have created a working prototype of the XBRL repository and an XBRL registry. These systems focus on sites that are driven by public interest. In time, websites such as EDGAR Online and EDGAR XBRL Express will eventually offer XBRL files to end users for analysis. In absence of the SEC requiring XBRL reporting, commercial data aggregators will continue to customize (harmonize) data, whether submitted in XBRL or otherwise, to achieve certain degree of uniformity to assist a wide range of users. In the mean time, given obstacles to easily identify, locate, validate, and authenticate XBRL files on the web, it seems logical to assume that individual investors and creditors have a long way before they can use XBRL data by do-it-yourself approach; and the investing community may gravitate, at least initially, towards XBRL data and format provided by data aggregators.

This project has provided an opportunity to study the issues and difficulties of providing timely financial information in a public venue. In the near future, when we have sufficient XBRL disclosures on the web, the XBRL repository can be used for other research projects, especially those, such as data mining, that would benefit from the availability of a large number of XBRL files.

## **5. REFERENCES**

BONSÓN, E. (2001): "The Role of XBRL in Europe", *The International Journal of Digital Accounting Research*, 1, 2: 101-110.

BOVEE, M.; SRIVASTAVA, R.P. (2001): "FRAANK-XBRL: An Intelligent Tool for Converting Financial Statements in Text to XBRL-Tagged Documents",

*First International Conference on XBRL Taxonomies: Current Issues and Future Directions*, University of Kansas, October.

COHEN, E.E.; LAMBERTON, B.; ROOHANI, S. (2003): *The implications of Economic Theories and Data Level Assurance Services: Research Opportunities in Trust and Data Assurances in Capital Markets: The Role of Technology Solutions*, Research Monograph (Ed: Saeed J. Roohani): 51-61

EDGAR (2003): *Microsoft to Team With EDGAR Online to Facilitate Financial Analysis On the Desktop*. Edgar Online Press Release. <http://www.edgar-online.com/investor/news/110503.asp>

HANNON, NEAL (2004): "XBRL and Metcalfe's Rule of Technological Change", *Strategic Finance* (85:7): 57-58.

HUNTON, J; WRIGHT A.; WRIGHT S. (2003): *The Supply and Demand for Continuous Reporting in Trust and Data Assurances in Capital Markets: The Role of Technology Solutions*, Research Monograph (Ed: Saeed J. Roohani): 7-16.

LIOTTA, M.; PREIMESBERGER, C. (2003): "Native XML Databases Resolve XML Document Retrieval Issues", CNET Networks, Inc. <http://builder.com.com/5102-6388-1051795.html>

MCNAMAR, R.T. (2003): "New Technology can Help Avoid a Second Enron", *Regulation*, 26, 3: 62-67.

MICROSOFT (2004): *Microsoft Office Tool for XBRL™ Prototype*. Microsoft Corporation. <http://www.microsoft.com/office/solutions/xbri/datasheet.mspx>

OSHEROFF, M.L.; GREEN, M.W.; SANDERS, R.A.; DAS, S.K. (2003): "Electronic Filing and the EDGAR System: A Regulatory Overview", Securities and Exchange Commission, Divisions of Corporation Finance and Investment Management. <http://www.sec.gov/info/edgar/regoverview.htm>.

ROTH, M.A.; WOLFSON, D.C.; KLEWEIN, J.C.; NELIN, C.J. (2002): "Information Integration: A New Generation of Information Technology", *IBM Systems Journal*, 41, 4: 563-577.

SOFTWARE AG (2003): *XML Databases. A Tamino XML Server White Paper*/ Software AG. [http://www.softwareag.com/tamino/download/WP\\_XML\\_Databases\\_E.pdf](http://www.softwareag.com/tamino/download/WP_XML_Databases_E.pdf)

TAM, K. (2003): “An XML Schema Design Framework to Simplify, Financial Statement Validation”, *The International Journal of Digital Accounting Research*, 3, 6: 133-164.

WEISEL, J.A. (2002): “The Financial Reporting Revolution and XBRL: Another Fad or is this Really Going to Work?”, *Journal of Accounting and Finance Research*, 10, 3: 26-34.

WILLIS, M. (2003): “Corporate Reporting Enters the Information Age”, *Regulation*, 26, 3: 56-60.

XBRL (2004): XBRL International, Inc. <http://www.xbrl.org>