

December 2001

Examining a New Approach: XBRL Applications - A Case Study

Terry Fox
Baylor University

M. George Durler
Emporia State University

Zane Swanson
Emporia State University

Nitham Hindi
Emporia State University

Follow this and additional works at: <http://aisel.aisnet.org/amcis2001>

Recommended Citation

Fox, Terry; Durler, M. George; Swanson, Zane; and Hindi, Nitham, "Examining a New Approach: XBRL Applications - A Case Study" (2001). *AMCIS 2001 Proceedings*. 102.
<http://aisel.aisnet.org/amcis2001/102>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISEL). It has been accepted for inclusion in AMCIS 2001 Proceedings by an authorized administrator of AIS Electronic Library (AISEL). For more information, please contact elibrary@aisnet.org.

EXAMINING A NEW APPROACH: XBRL APPLICATIONS — A CASE STUDY

Terry L. Fox
Baylor University
Terry_Fox@baylor.edu

M. George Durler
Emporia State University
durlerge@emporia.edu

Zane Swanson
Emporia State University
swansonz@emporia.edu

Nitham Hindi
Emporia State University
hindinit@emporia.edu

William S. Remington
Emporia State University
remingtw@emporia.edu

Abstract

eXtensible Business Reporting Language (XBRL) is a new approach that offers great promise for rethinking the way in which financial information is maintained and reported. This paper presents a case study of three student projects that resulted in the development of XBRL applications. A discussion of the teams, the projects, and most important lessons learned is presented.

Introduction

XBRL – one of the latest acronyms in a technological ocean of new concepts, new techniques, new tools, and new approaches. XBRL – eXtensible Business Reporting Language – is a new approach to making financial data easily accessible and reportable. In a nutshell, this approach consists of a generic (XBRL) taxonomy that standardizes the variety of terms by which various organizations refer to their financial information, the ease of XML (eXtensible Markup Language) to act as a data file, and the flexibility of XSLT (eXtensible Stylesheet Language Transformations) that defines what format a user wishes to view selected data. Combined, these elements offer the opportunity of causing us to rethink the method in which financial data is reported and compared within and between organizations. Consider that using a generically defined approach, an investor, banker, or other interested party could easily download multiple companies' published financial information, each one in a standard format, pull the data into a spreadsheet for the purpose of running various analyses and comparisons, save the results as an XML "data" file, and "publish" all or part of the information on an intranet or the Internet using a standardized format. The implications of the types of applications that could be developed to take advantage of this approach are countless.

This paper reports on the work of three student groups, each developing an XBRL application. The primary catalyst behind these projects was the first annual XBRL Academic Competition 2000-2001 sponsored by Bryant College. An obvious secondary motivator was the challenge of learning and applying a very new approach with relatively few references and resources from which to refer.

The Teams

Five faculty members, three from the accounting discipline and two from the computer information systems (CIS) discipline, sponsored the three student groups. One group was comprised of graduate students; the other two groups consisted of undergraduate students. Each group had 5 students, with a mix of both accounting and CIS majors. Some students received

independent study credit for their work. Regular meetings were scheduled approximately every two weeks where the faculty would present material pertaining either to XBRL, XML, XSLT, or coverage of application knowledge.

The Projects

Each of the three teams was assigned a different application to develop. One team was assigned the task of developing a foreign currency translation application. The second team was assigned the task of developing a parent-subsiary consolidation application. The third team was assigned the task of developing an application that would calculate a materiality threshold for a set of financial statements. All the teams were required to adhere to the XBRL taxonomy, maintain data in XML format, and produce results that could ultimately be placed on web site. Current XBRL style sheet formats can best (only) be viewed using Microsoft's Internet Explorer 5.0. Each team also incorporated the use of MS Excel that would read the initial XML data, perform necessary calculations, and, as an option, generate the results as an updated XML file. The resulting applications, with an explanation of each, are available for review at www.emporia.edu/business/acis/xbrlproject.

Example documentation for the project included a project report, organized as follows:

- Executive Summary
- Table of Contents
 - Introduction
 - Explanation of the accounting topic (e.g., consolidations)
 - Operating Instructions
 - Installation Instructions
 - Future Implications
- Appendix
- Visual Basic Code

Lessons Learned

Perhaps the two most important aspects of this project were 1) the knowledge acquired of this new approach to maintaining financial information, and 2) the importance of working together in a cross-functional team.

From the outset, neither the students nor faculty sponsors knew much, if anything, about the components required to develop the application. Very little has been written to date on the "how" of XBRL. We did locate, however, several good sources on XML, most of which included a section on developing style sheets. The faculty went through a series of "try this and see if it works", often going back to square one but occasionally enjoying the fruits of our labors when a particular aspect actually made sense and worked. The faculty then took the responsibility of sharing this new knowledge with the students via seminar classes, who were also charged with discovering what they could on their own. All in all, a very collaborative effort was achieved. A breakthrough was achieved with the realization that the three components, XBRL, XML, and XSL were not so much alternatives to doing the same thing, but actually need to exist in a symbiotic relationship. The key to XBRL is the *taxonomy* which defines a standard set of element names and descriptive labels (currently 1,880 elements) for commercial and industrial organizations (ci) that when adhered to, makes every financial application generic across all organizations. For example, in referring to Cash, the XBRL taxonomy has defined Cash with the element name of "cashAndCashEquivalents.cash", and a descriptive label of "Cash". Likewise, Property, Plant and Equipment net of Accumulated Depreciation has the element name of "noncurrentAssets.propertyPlantAndEquipmentNet", and a descriptive label of "Property, Plant and Equipment, Net". A discussion of the precise syntax and hierarchy rules of XBRL is beyond the scope of this paper, however the reader is encouraged to examine www.xbrl.org for additional information.

XBRL is a derivative of XML, which can simply be regarded as the data file. This data is not stored in relational format, but rather a hierarchical or tree-structured format with the ability to store a variety of attributes with each element or data record. For example, a record pertaining to "Cash" could be included in the XML file, along with its' element name, its label, the balance sheet date, and the amount. Not offering a relational format is not necessarily a problem, and actually makes generic data retrieval simpler. An XML file can be created and edited using Microsoft NotePad or any other editor which creates text files. There are also a variety of new utilities available to automatically generate an XML file based on a minimal amount of user input. One such XML generator, written in Microsoft Access 2000, was created by one of the authors of this paper and can be located on the web address for the projects (downloadable soon). XML makes use of HTML-like tags, and the above example could be created as follows:

```
<group type='ci: cashAndCashEquivalents.cash '>
  <label>Cash</label>
  <item period='12-31-2000'>100000</balance>
</group>
```

Once the data is in XML format using the generic XBRL taxonomy, it can be read (utilizing Visual Basic) into an Excel spreadsheet for whatever manipulation and review the user needs. We preface the ability to open the XML file into Excel by stating that a bit of Visual Basic code is necessary to make it work seamlessly. Visual Basic can read the text file, parse it into records, identify the attributes (element name, label, amount, etc.) and place it into a defined cell in an already formatted spreadsheet. One of our students made the revolutionary suggestion that by assigning the XBRL element name to be a specific spreadsheet cell name, the placement of the data can be very simple and generalizable across any spreadsheet application. The Visual Basic code that actually does the importing of the XML data into a spreadsheet, and assigns it to pre-named cells is as follows:

```
Set oNodeList = xmlDocument.selectNodes("/group/group/*")
Dim eachNode As Integer
For eachNode = 0 To oNodeList.Length - 1
  xElement = oNodeList.Item(eachNode).selectSingleNode("@type").Text
  xLabel = oNodeList.Item(eachNode).selectSingleNode("label").Text
  xItem = oNodeList.Item(eachNode).selectSingleNode("item").Text
  rangename = Right(xElement, Len(xElement) - 3)
  Range(rangename).Select
  ActiveCell.Value = xItem
Next eachNode
```

Please note, there is additional code required to declare variables and to initially open the XML file, but there is not room in this paper to include all the code. Suffice it to say, the user needs a little familiarity with Visual Basic in order to create a similar spreadsheet application. The reader is again encouraged to visit our project's website and examine the various applications.

Once the XML file is created, using the XBRL taxonomy, the final step is to allow for the ability to display the XML data in a web format. This is where the concept of style sheets comes in. We chose to use XSLT (eXtensible Stylesheet Language Transformations) to accomplish this purpose. While a discussion of all the things that XSLT can do with an XML file are well beyond the scope of this paper, what this boils down to is essentially formatting the XML file to look good. This includes defining tables, setting fonts and colors, etc. Style sheets are also capable of sorting XML data, selecting various elements and attributes from an XML file, and even performing mathematical calculation on the data.

As mentioned above, the second major benefit of this undertaking was the cross-functional nature of the projects. Not only did the faculty have to team up and contribute their referential knowledge, even more importantly the students realized that given a project of this nature, requiring both systems development skills from the CIS students, as well as accounting functional knowledge from the accounting students, that no one individual, group, or organization holds all the answers. Only by working together, sharing information, and contributing as a team, can a project be successful.

XBRL, XML, and XSL Sources

XBRL Homepage	www.xbrl.org
Bryant College XBRL Resources	web.bryant.edu/~xbrl/index.html
XML Homepage	www.xml.org
XSL Frequently Asked Questions	www.dpawson.co.uk/xsl/xslfaq.html
XBRL Taxonomy	www.xbrl.org/us/gaap/ci/2000-07-31
XML Developer's Guide	http://msdn.microsoft.com/xml/xmlguide