An Analysis Of Student XBRL Project Teams: Value-Adding Educational Experiences Using Emerging Technology

Terry L. Fox, (E-mail: terry_fox@baylor.edu), Baylor University
Zane Swanson, (E-mail: swansonz@emporia.edu), Emporia State University
William S. Remington, (E-mail: remingtw@emporia.edu), Emporia State University
M. George Durler, (E-mail: durlerge@emporia.edu), Emporia State University
Nitham Hindi, (E-mail: hindinit@emporia.edu), Emporia State University

Abstract

Extensible Business Reporting Language (XBRL) is a new approach that offers great promise for a fundamental improvement to the way in which financial information is maintained and reported. With every innovation, comes the need to educate. Because of the overarching aspects of XBRL and its implications, a team learning approach is appropriate. This paper presents a case study of three student projects that resulted in the development of XBRL applications. An analysis is provided of the teams, the projects, and important lessons learned.

Introduction

Current industry practices rely heavily on a team approach, suggesting that universities should be using team-based activities. The purpose of this paper is to describe a case example of how student teams created software applications utilizing a project design approach, which has been lightly addressed in previous educational literature. The projects described herein brought an emerging technology - eXtensible Business Reporting Language (XBRL) - into our university experience. The leading-edge nature of the technology required that we employ a "learn as you go" approach, but the methods are applicable to often-used software platforms as well. These projects also took a cross-functional approach, which may be of interest to other universities who want enrich the educational experience by crossing the boundaries between academic disciplines.

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. A secondary motivator was the challenge of learning and application of an emerging technology with relatively few references and available resources.

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-subsidiary 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 eXtensible Markup Language (XML) format, and produce results that could ultimately be placed on a web site. Currently XBRL style sheet formats can best (only) be viewed using *Microsoft Internet Explorer* browser, version 5.0 or higher. Each team also incorporated the use of *Microsoft Excel 2000* 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. Documentation for the project included a project report, organized as

Readers with comments or questions are encouraged to contact the authors via email.

follows: Executive Summary, Table of Contents, Introduction, Explanation of the accounting topic (e.g., consolidations), Operating Instructions, Installation Instructions, Future Implications, Appendix, and Visual Basic Code. Each of the three projects was completed in a timely fashion, and each with a workable software solution.

The remainder of the paper is organized as follows: the concept and brief explanation of XBRL as an emerging technology is introduced, followed by a discussion describing issues in managing student projects. Lessons learned from this experience and a summary are then provided.

XBRL as an Emerging Technology

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 defined taxonomy – XBRL – 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 structure, and the flexibility of XSLT (eXtensible Stylesheet Language Transformations) that defines how a user can 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. 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 types of applications that could be developed to take advantage of this approach are countless.

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 financial data generic across all organizations and applications. 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". Similarly, 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 the entire XBRL taxonomy is beyond the scope of this paper, however the reader is encouraged to visit www.xbrl.org for additional information.

XBRL is a derivative of XML, which can simply be regarded as the data file organization. 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 that 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. XML uses HTML-like tags, and the above example of the cash account could be created as follows:

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

Once the data is in XML format using the generic XBRL taxonomy, it can be read into a *Microsoft Excel* spreadsheet for whatever manipulation and review the user needs. We preface the ability to open the XML file into *Microsoft Excel* by stating that, with *Excel 2000*, it is necessary to write a short routine using *Visual Basic* code to make it work seamlessly. (*Microsoft Excel XP* is said to be able to open an XML file directly, but in our experience this function is dependent on various factors, such as the XML hierarchy used and whether a style sheet is incorporated in the file.) The routine written using *Visual Basic* can read the text file, parse it into records, identify the attributes (element name, label, amount, etc.) and place each data element into a defined cell in an already formatted spreadsheet. One of the 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 greatly simplified and generalized across any spreadsheet application.

A subset of the *Visual Basic* code that actually imports 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 using *Microsoft Excel 2000*. The reader is again encouraged to visit our project's website and examine the various applications for further details.

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. While a discussion of what formatting capabilities are possible using XSLT with an XML file are well beyond the scope of this paper, style sheets essentially provide the formatting that makes the XML file look good, including 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. The reader is encouraged to refer to Table 1 for selected web sites that can assist in the understanding of these concepts.

Table 1 – XML, XBRL, and XSL Reference 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

Managing Student Software Development Projects

Our school advisory council continues to stress the need to expose students to team projects, and a majority of the courses in our curriculum include just such a component. It is well known that managing and coordinating a student team project can be considerably different from a "real-world" project. For example, a variety of factors, ranging from motivation – "I just need a 'C' to graduate" – to scheduling – "The only time I can meet is between 2:00 am and 4:00 am", that are present in an academic setting are removed in real life – if you don't meet expectations you lose your job, and generally there is a requirement that you will be available during certain normal work hours. However, many of the elements of a real world project are found in a student team project. The most notable of these are the schedule, including a deadline after which severe penalties can be assessed, and a set of requirements that must be achieved in order for the project to be considered a success.

From the outset, all three teams were provided with a schedule that included various milestones dates by which time things such as the analysis, preliminary design, final testing, and documentation, were to be completed. The setting of milestones encouraged the students to stay on task and helped reduce the tendency to procrastinate. The students were also provided with a complete set of requirements for their respective project – not so much the *how*, but the *what*. The measure of success was not to win the contest, but rather to have a good, functional application ready to submit by the time the contest deadline arrived.

Relatively little literature is available which examines the factors at work in student software project teams. Group dynamics, interpersonal and team skills as they apply to technology have been addressed several times in the literature (Rooney, 2000; Jessup, 1992, Jessup, 1991). Recent studies related to team projects have examined issues such as managing interpersonal conflict (Barki and Hardwick, 2001) and cross-functional teams (Sethi, Smith, and Whan, 2000; McDonough, 2000), but these studies addressed professionals rather than students. One recent study of graduate students involved in a software project examined perceptions of their contribution to the project (Rajlich, Syed, and Martinez, 2000). Watson, Johnson, and Merritt (1998) examined various issues pertaining to diversity in student teams, but not with a systems development project. Very little, if anything, has recently been published related to the cross-discipline systems development effort undertaken here.

As mentioned above, it was imperative to make expectations known early, and to manage the schedule closely. Many issues needed to be managed as the teams progressed through their projects. Thomas (2000) provides a good checklist that the faculty sponsors generally followed when working with their teams. Thomas (2000) divides his checklist into two areas – "Selecting the right people" and "Getting the most out of your team."

When "selecting the right people", the first order of business is *qualifications*. The faculty met and selected a group of students who were invited to participate in the project. These students were considered based on their academic standing, leadership, and interpersonal skills. 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 composed of graduate students; the other two groups consisted of undergraduate students. A specific mix of both accounting and CIS students was necessary to provide the needed application area and technical knowledge. 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. Thomas (2000) also suggests that you consider attitude, attendance, proactive behavior, and personality types when selecting project teams.

The second step is to consider *availability*. The undergraduate students were given the option of receiving independent study credit for the project, and therefore invested the time spent on this project the same (or more) as they would have a regular lecture course. The third factor mentioned by Thomas (2000) is *interest level*. Each student who was invited to participate was given ample opportunity to decline, and a few, in fact, did. The remaining students all showed a great deal of interest in the project, due to the challenge of working with new technology and the non-traditional classroom environment.

Thomas' (2000) fourth factor is *chemistry* – can the team members work together. Many of the students already knew each other before the project began; some had worked on other projects together in the past, while some had not ever met before. No evidence of a lack of chemistry was seen during the project, and faculty observations noted that the students tended to work quite well together. The fifth point is that the team members need *balance*. This does not mean like-mindedness, but rather more of a Ying-Yang environment. Strengths of one member overcome the weakness of another, and the perspective of one individual, while quite possibly different from the others, is respected and considered. The cross-functional nature of this project required the recognition of the student's own weaknesses and the strengths of their teammates. The accounting students generally yielded to the technical knowledge of the CIS students, and the CIS students gained a much better appreciation of the need for application area knowledge as provided by the accounting students. The quality of the students participating, and the respect shown to them by the faculty, also contributed to the cooperation that existed across disciplinary boundaries.

Thomas (2000) then goes on to address how to "get the most out of your team" in three areas. First, *create buy-in* for the project, partly by getting input from the project team members. Throughout this project, both the faculty and students were learning as we went. Often the students would discover something that their sponsor had not considered. Also, the friendly competitive environment of the contest provided a rallying point for the teams.

Second is the need for *team-building* functions. The initial meeting of all three groups was in the form of a luncheon, where all participants were able to meet and be introduced to their teammates. On a number of occasions, teams held informal pizza parties to relax and talk about things other than "XXX" (XML, XSL, and XBRL). Working on the project away from a traditional classroom environment also contributed to the "building" of the teams.

The third point that Thomas (2000) makes is to *create an urgency* for results. With the contest deadline growing nearer each day, the students seemed to build up more and more momentum. As it was, each of the teams was essentially complete with their project at least one day before the deadline.

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 had been written at this point 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. After struggling with the various acronyms and what each really meant, a breakthrough occurred 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 second major benefit of this undertaking was the cross-functional nature of the projects. It was necessary for the faculty to team up and contribute their referential knowledge, but 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 has all the answers. Only by working together, sharing information, and contributing as a team, can a project be successful.

This XBRL software application project sheds light on a few additional issues. For example, consideration needs to be given as to the instructor's choice between a typical class project and an independent study project when examining unstructured topics, particularly those addressing emerging issues. A key factor in this choice is the educational value added for the students. Our experience indicates that in these situations, an independent, or in this case a small group, study can offer the best learning experience. Student team membership can also more easily be controlled in a small group project. In our case particularly, the selection of superior students provided highly motivated group members. These students are more likely to work just for the knowledge gained from this experience. In our project the reward was in the learning.

In the event that institutional factors limit the offering of independent or small group studies, an alternative technique for projects such as this is to assign them to cross-functional teams consisting of students from Systems Analysis and Design and Accounting Information Systems courses. In many universities and colleges, these courses emphasize teamwork and the applications as well as the implications of emerging technology. Such projects could be very valuable tools when teaching teamwork and/or the impact of emerging technology (e.g., improving the company's reporting system using XML).

Summary

Our experience with exploring emerging technologies (e.g., XBRL) using a cross-functional student team environment indicates that this can be an effective means of educating students. This proved to be a very educational experience not only for the students, but the faculty, as well. We also discovered that good project management is a key component in this type of educational activity. Working as a team, and recognizing the unique strengths of each team member, contributed greatly to the successful completion of each application. Although the teams did not end up winning the contest, they were each successful in their endeavor.

Suggestions for Further Study

As mentioned in the discussion of lessons learned, the question of whether to offer independent/small group projects or whether to make these types of projects part of a regular class need to be considered. There are positive and negative factors associated with either approach. Certainly, when examining emerging technological issues, projects of this type provide a very good learning experience, not only for students but instructors, as well. Providing a collaborative environment in which all parties are learning from each other can be very beneficial and needs to be further explored. Lastly, the potential of XBRL is just beginning to be examined and considerable study needs to undertaken to determine the full impact that this new technology can have on financial reporting applications. Studies of the perceived impact and use of XBRL applications are also warranted.

References

1. Barki, Henri, and Jon Hartwick, "Interpersonal Conflict and its Management in Information System Development," *MIS Quarterly*, Vol. 25, No. 2, pp. 195-228, 2001.
2. Jessup, Harlan, "A Model for Workteam Success," *Journal for Quality & Participation*, Vol. 14, No. 3, pp. 70-74, 1991.
3. Jessup, Harlan, "The Road to Results for Teams," *Training & Development*, Vol. 46, No. 9, pp. 65-68, 1992.
4. McDonough, Edward F. III, "Investigation of Factors Contributing to the Success of Cross-Functional Teams," *Journal of Product Innovation Management*, Vol. 17, No. 3, pp. 221-235, 2000.
5. Rajlich, Vaclav, Waseed Abdul Syed, and James Martinez, "Perceptions of Contribution in Software Teams," *Journal of Systems and Software*, Vol. 54, No. 1, pp. 61-63, 2000.
6. Rooney, Pamela, "Constructive Controversy: A New Approach to Designing Team Projects," *Business Communication Quarterly*, Vol. 63, No. 1, pp. 53-61, ???? .
7. Sethi, Rajesh, Daniel C. Smith, and C. Whan, "Cross-Functional Product Development Teams, Creativity, and the Innovativeness of New Consumer Products," *Journal of Marketing Research*, Vol. 38, No. 1, pp. 73-85, 2000.
8. Thomas, Michael. "Building and Managing a Winning Project Team," *Manage*, Vol. 52, No. 1, pp. 4-5, 2000.
9. Watson, Warren E., Lynn Johnson, and Deanna Merritt, "Team Orientation, Self Orientation, and Diversity in Task Groups," *Group & Organization Management*, Vol. 23, No. 2, pp. 161-188, 1998.