2011

Xml Beyond The Tags

Christopher Adam Meloy
University of Central Florida

XML: BEYOND THE TAGS

by

CRISTOPHER ADAM MELOY
B.A. University of South Florida, 2008

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Arts
in the Department of English
in the College of Arts and Humanities
at the University of Central Florida
Orlando, Florida

Fall Term

2011

ABSTRACT

XML is quickly being utilized in the field of technical communication to transfer information from database to person and company to company. Often communicators will structure information without a second thought of how or why certain tags are used to mark up the information. Because the company or a manual says to use those tags, the communicator does so. However, if professionals want to unlock the true potential of XML for better sharing of information across platforms, they need to understand the effects the technology using XML as well as political and cultural factors have on the tags being used.

This thesis reviewed literature from multiple fields utilizing XML to find how tag choices can be influenced. XML allows for the sharing of information across multiple platforms and databases. Because of this efficiency, XML is utilized by many technologies. Often communicators must tag information so that the technologies can find the marked up information; therefore, technologies like single sourcing, data mining, and knowledge management influence the types of tags created. Additionally, cultural and political influences are analyzed to see how they play a role in determining what tags are used and created for specific documents. The thesis concludes with predictions on the future of XML and the technological, political, and cultural influences associated with XML tag sets based on information found within the thesis.

This thesis is dedicated to those who kept believing in me and supported me during tough times while finishing my degree. It’s because of you that I was able to keep motivated.

ACKNOWLEDGMENTS

I would like to thank Dr. Applen, Dr. Kamrath, and Dr. McDaniel for the time they put in to help me through this thesis. They provided invaluable insight in how to create a stronger document and to improve my writing in the future.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

AI	Artificial Intelligence
AOM	Agent Oriented Modeling
AOML	Agent Oriented Modeling Language
CCO	Cataloguing Cultural Objects
CD	Compact Disk
CEO	Chief Executive Officer
CIO	Chief Information Officer
CMS	Content Management System
CMWS	Common Missile Warning System
DBMS	Database Management System
DITA	Darwin Information Typing Architect
DOM	Document Object Model
DTD	Document Type Definition
EAD	Encoded Archival Description
EPSS	Electronic Performance Support System
HTML	Hyper Text Markup Language
IRS	Internal Revenue Service
IT	Information Technology
MDDBMS	Multidimensional Database Management System
NDIPP	National Digital Information Infrastructure and Preservation Program

NINES	Nineteenth-century Electronic Scholarship
OASIS	Organization for the Advancement of Structured Information Standards
ODF	OpenDocument Format
OLAP	Online Analytical Processing
OMB	Office of Management and Budget
OOXML	Open Office XML
OS	Operating System
PC	Personal Computer
PCD	Performance Centered Design
PDF	Portable Document Format
POC	Point of Contact
RDF	Resource Description Framework
RNG	RELAX NG
SAX	Sample API for XML
SGML	Standard Generalized Markup Language
SME	Subject Matter Expert
SOAP	Simple Object Access Protocol
SQL	Search and Query Language
TC	Technical Communication
TEI	Text Encoding Initiative
URI	Uniform Resource Indicator
URL	Uniform Resource Locator

USAF	United States Air Force
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XSD	XML Schema Definition

CHAPTER ONE: INTRODUCTION

Extensible Markup Language (XML) refers to a tagging language that focuses on the content over the aesthetics of information. XML has been used in the business and financial industries for many years but is just now being incorporated into the technical communication (TC) field. Because XML is a relatively new technology being used in the TC field, many professionals will be hesitant to use it or fall into the regurgitation process of repeating the tags they already saw in the document. For instance, most corporations tag information in XML and want their technical writing department to duplicate the tagging pattern the company currently uses. This process influences the regurgitation of tags instead of taking time to understand them and develop better tags. Both refusal of technology and regurgitation sell XML short.

XML is more than just tags; it has a rhetorical property to it. Why does a writer create a tag `<given_name>` for first name instead of `<first_name>` or `<birth_name>`? What determines the tree structure of the tagging being used over a different structure? These are just some simple examples of how influences on the tagging can affect the choices technical communicators make. The influences can be rhetorical, political, and cultural or come from a technology like data mining and single sourcing. In either case, there is an influence on the XML tags being used. If technical communicators could acquire the knowledge of what goes into deciding why a certain tag is used, then technical communicators could create tagging structures that would increase the efficiency of information transfer and create the greatly desired Semantic Web.

For this paper, writer will refer to the technical communicator. Markup and tags will also be used interchangeably.

Purpose

Current XML technology allows for any company and/or writer to create their own tagging for the information they are marking up. Because XML is a very user friendly language, many technical communicators will follow the company's Document Type Definition (DTD) or schema without understanding why the tags they are using are being used. XML tagging is greatly influenced by outside factors. While these factors can be near infinite, this paper looks to analyze some of the most common ones within the TC field. These influences include data mining, single sourcing, knowledge management, and rhetorical, cultural and political factors. Most technical writers will not know what factors helped choose their current tagging structures and some do not care to know, but the more information the writers have about the tags they are using, the more likely they are to help create structures that are more uniform and accessible for people looking for the information.

Overview of Chapters

Chapter 2 reviews XML basics for the uninformed reader. The chapter will start with basic XML structures that writer might use within their field. An example will be provided to help the reader understand why tagging and structure is important for XML to operate correctly.

Next DTD and schemas will be discussed. While many technical communicators will not have to construct or edit the DTD or schema, the DTD and schema play an important role in XML structure. The third area of this chapter will look at Darwin Information Typing Architecture (DITA). DITA uses a system of topics and tagging to formalize tagging structures within the TC field. While DITA seems to be a good choice with regards to making tagging more formalized, not all companies use DITA. Analyzing why some companies do and do not use DITA will help in understanding the decisions behind tagging in XML. Text Encoding Initiative (TEI) is also briefly discussed; however, TEI is limited to the fields of humanities and social sciences. Using all of the information discussed earlier in the chapter, we can then analyze why XML is a great tool for technical communicators in their field.

Chapter 3 brings together XML and TC. Technical communicators have mixed feelings when it comes to XML. Some professionals see XML as a chance to advance TC while others sees XML as hindering the writer’s abilities. Both of these arguments will be analyzed in this chapter. Next the chapter will look at how XML is incorporated in the field. Several examples will be used to show how XML is used by technical communicators, which will lead to the brief introduction of rhetorical choices and influences on XML.

Chapter 4 begins with a discussion on data mining. Data mining uses XML in order to locate patterns within information and locate specific information for companies’ needs. While technical communicators will typically not create the data mining programs, some will use them to locate information in XML. Data mining has certain criteria by which to search out information, therefore influencing the specific tagging being used to markup the information being searched. For example, some data mining programs only search for certain sibling tags while other will completely ignore duplicate tags in a tree structure. Depending on the information needed, the tagging will be created for that need. I will then analyze benefits and ethical dilemmas of data mining. While XML allows data mining to be efficient in its search capabilities, the flexibility of XML can lead to misuse of data mining. Finally, the data mining software and the use of the software in the TC field will be discussed.

Chapter 5 focuses on another technology that can be based on XML, which is single sourcing. Single sourcing relies on XML structure to locate modules of information or metadata to be reused for multiple applications. The purpose of single sourcing, its tools, and use in the TC field all play a role in what tagging structure is used in XML. This information will then be used to see how single sourcing and XML work together and how this teamwork plays a role in the creation of tags in XML.

Chapter 6 analyzes the importance of knowledge management. Knowledge management affects the setup of information within databases, the processes of organizing information for transfer to other people, and XML tag structures. I will discuss several examples of the tools used in knowledge management and how they are used in the field of TC. The chapter will then show the incorporation of XML into knowledge management and how knowledge management influences the tagging decisions in the TC field.

Chapter 7 looks at the more human-based rhetorical influences of XML tagging, which include cultural and political influences. Rhetorical, cultural, and political influences can have a great influence on what tags will be used to markup information. This chapter will break down each of these factors that influence XML tagging and show how they shape XML, if they hinder or encourage XML development, and their effect on tagging in the TC field.

Chapter 8 is more of an educated opinion based on all the information from the previous chapters. All of the factors mentioned previously will play some role in the evolution of XML tagging. While nobody can say for sure what the future holds for XML tagging years down the road, this chapter will make plausible predictions about DTDs and schemas, data mining, single sourcing, knowledge management, and other rhetorical factors that influence XML tags. The chapter will then make a prediction on what these future factors will have on XML tagging. The final section of Chapter 8 is the conclusion to the thesis. This section will sum up the concept discussed and provide any final thoughts on the subject of influences on XML tagging.

CHAPTER TWO: WHAT IS XML?

XML Basics

XML is a term that has been thrown around in the TC community for some time. In many ways it is quite similar to Hyper Text Markup Language (HTML) yet completely different at the same time. Unlike HTML, XML focuses on the content over the presentation of information. XML does this by allowing authors to create tags through DTDs or XML schemas to mark up information for storing in databases, converting information to multiple formats, and sharing information across platforms like Windows, Linux, and Macintosh. Since XML is also a text file, it can be read by any application with the capabilities of reading text files. Any document that uses XML markup is considered an XML document. However, technical communicators must understand XML is a meta-language, which means XML is a technology that provides rules and structure which any markup language must follow (Morrison 9). What does this mean for the technical communicator? No matter what the document is, communicators can create their custom tagging system with the proper DTD or XML schema associated with the markup. This concept allows for many variations in tagging from corporation to corporation, making the XML technology very powerful. For the scope of this paper, only the tagging structure and tagging rules will be analyzed.

XML is made up of specific tags created through the DTD or XML schema. DTDs and XML schemas will be discussed more in the next section. These tags are created by a structure where a single root tag has all other tags fall under. The tags under the root tag are known as parent tags and child tags. The structure of the tagging is in a hierarchical format and usually associated with a tree structure. Figure 1 provides an example of a simple tree structure for an XML document. As the figure shows, parent elements fall under the root element and child elements fall under the parent elements. Child elements of the same level are known as siblings and all siblings in an XML document may only have one parent. All elements or leafs within the tree structure are also known as nodes. Nodes are very important to XML documents because they associate the child element with the parent element, which is crucial for XML documents to be parsed correctly. Parsing is where an application takes the XML document and analyzes its structure which allows for the information to be manipulated into the desired need of the technical communicator.

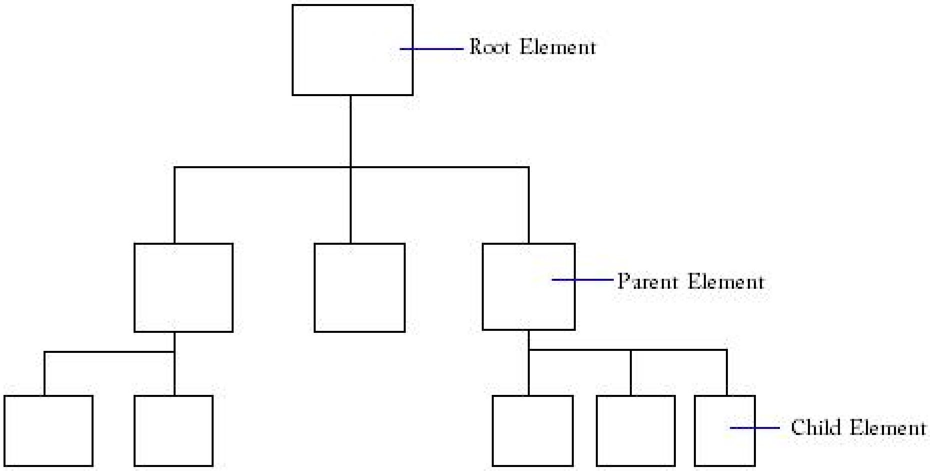
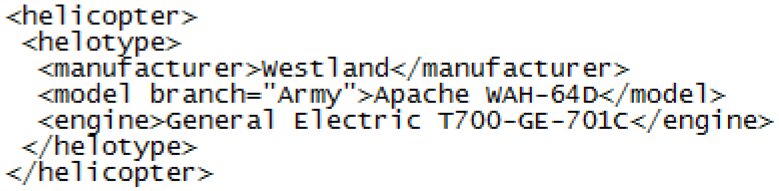


Figure 1: Simple Tree Structure

Anytime a technical communicator will tag a document, this tree structure will be followed. An example of the tagging in an XML document can be seen below. Figure 2 shows an example of a tagging structure that a communicator might see in a defense contractor document. This is an example tagging I made up and would be used in a large document about this specific helicopter or a document listing a number of helicopters currently used by the military. This XML document is very simple but displays many of the characteristics of a longer XML document. The root element is `<helicopter>`, followed by the parent element `<helotype>`, and three child elements `<manufacturer>`, `<model>`, and `<engine>`. The XML document follows the hierarchical structure of XML and even includes an attribute with its value in quotes. Technical communicators must make sure they are allowed to use attributes when marking up the company's documents. This is because some companies take the view that attributes are for metadata and elements are for informational data while other companies believe there is no obvious difference between metadata and informational data (Harold and Means 17). If you are a communicator that falls into the first category, you should use attributes sparingly and try to markup the information with elements for better organization of the data. If you are a communicator that falls into the second category, try to create elements that cover the data an attribute would normally be used for instead. In the example below, the attribute provides the military branch information on the model of helicopter in the inventory list.



```
<helicopter>
 <helotype>
  <manufacturer>Westland</manufacturer>
  <model branch="Army">Apache WAH-64D</model>
  <engine>General Electric T700-GE-701C</engine>
 </helotype>
</helicopter>
```

Figure 2: Simple Tagging Example

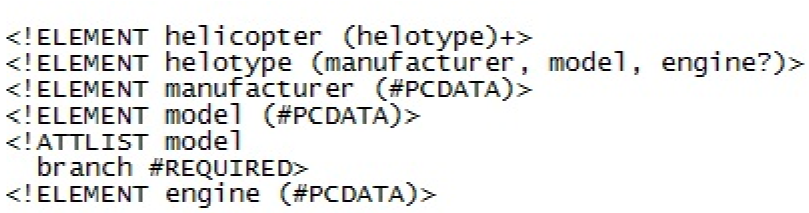
The key to tagging a document is to make sure it is well formed. A well-formed document follows rules like no overlap in the tagging structure, case sensitivity, elements are spelled right, and unquoted attribute values to name a few. If a document is not well formed, it cannot be parsed by any application. Once the technical writer discovers a well formed error, he or she must correct it in order to allow the document to be parsed by an application. The result may be an error displayed in a screen below the document or displays the entire document with markup.

Technical communicators also want their document to be valid. While validity is not necessary for a document to be parsed, being valid is necessary for a document to be used with a DTD or XML schema. In order for a document to be valid, it must abide by the rules of the DTD and XML schema. If not valid, the parsers will not recognize the document and the results will show an error or possibly a fatal error depending on the application. While an error may be ignored by a parser, a fatal error will not allow the document to be parsed. This is equivalent to a well formed error discussed earlier. The only reason a technical communicator would not validate a document is if the document is being transferred to another database with a different DTD or XML schema.

DTD and XML schema

DTD and XML schema were mentioned a number of times in the previous section with regards to tagging an XML document. From here on, XML schema will be referred to as schema. Without the DTD or schema, documents would not be able to be parsed and the information could not be accessed in the databases. DTDs and schemas make the rules for the XML tags to follow. Technical communicators use DTDs and schemas for the creation of the markup language to be used, which means the tags are also made according to the technical communicator’s needs. In other words, the DTD and schema describe the elements and attributes used in a markup language, how these elements and attributes are associated with each other, and relationships between elements (Morrison 45). Another way to think about it is to look at Figure 1 from earlier in this chapter. The DTD or schema sets the rules for the XML document by listing what elements are to be used and how they relate to each other in a hierarchical way—root, parent, child—like the tree example. Elements are listed in the DTD or schema along with their attributes that will control how the data appears in a document, and how the data will be parsed for other uses. The relationship of elements used to markup data will then have an effect on the results from queries on the XML document. However, before any tags can be developed, the technical communicator must decide what to use, DTD or schema. Both work just fine, but DTDs have been around longer with more support and incorporation and schemas are designed with XML, therefore giving schemas more powerful capabilities when used with XML documents.

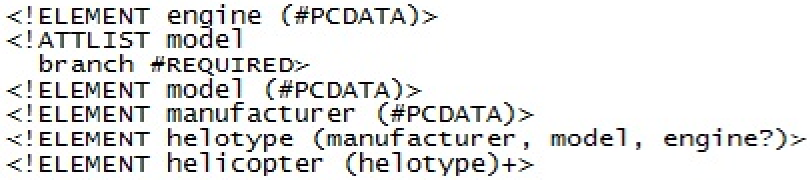
DTDs have been around longer than schemas and have a wide range of support. Typically a technical communicator would have to use a DTD with an XML document. Companies that have been using XML for a while will already have a master DTD for their documents to follow. Figure 3 shows an example of a DTD for the tagging from the last section.



```
<!ELEMENT helicopter (helotype)+>
<!ELEMENT helotype (manufacturer, model, engine?)>
<!ELEMENT manufacturer (#PCDATA)>
<!ELEMENT model (#PCDATA)>
<!ATTLIST model
  branch #REQUIRED>
<!ELEMENT engine (#PCDATA)>
```

Figure 3: Helicopter DTD

Figure 3 shows how the elements are created in lines 1-4 and 7. In line 1 the root element `helicopter` is listed followed by its child element, `helotype`. Line 2 shows the `helotype` element as a parent to the three child elements `manufacturer`, `model`, and `engine`. The `engine` element is followed by a `?` which means it is an optional element. This element does not have to be included if there is no information in the document for it. All three of these sibling elements are followed by `#PCDATA`. This statement indicates that this element may only contain text and no elements. Line 5 displays the attribute that goes with model. The attribute `branch` is followed by the `#REQUIRED`. The `#REQUIRED` statement indicates that a military branch must be provided with this attribute. If no branch is provided, than the document will be labeled as not valid. A DTD can be written like the example above with the highest level element (root) first or in opposite order with the highest level element listed last like Figure 4.



```
<!ELEMENT engine (#PCDATA)>
<!ATTLIST model
  branch #REQUIRED>
<!ELEMENT model (#PCDATA)>
<!ELEMENT manufacturer (#PCDATA)>
<!ELEMENT helotype (manufacturer, model, engine?)>
<!ELEMENT helicopter (helotype)+>
```

Figure 4: DTD Reverse Order

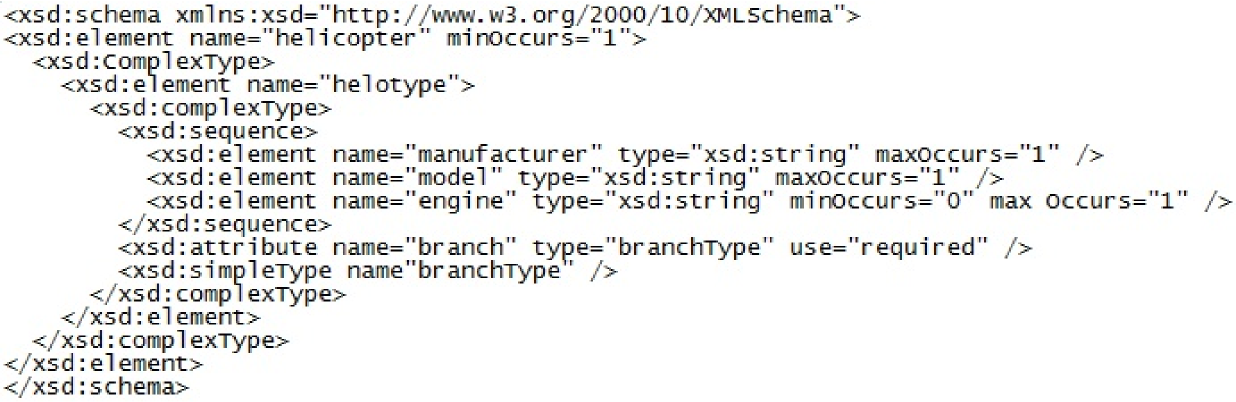
The DTD is not based off the XML language, but the more complex SGML. DTDs only care about the general information of tagging including elements used, attributes used, and the order in which they must follow. DTDs do not focus on the data between the tags in an XML document, only the structure of the XML document. If a technical communicator works with a DTD, it will either be internal or external. Internal DTDs are placed within the marked up document. The DTD coding would come after the declaration of the XML document. Many companies do not set up their DTDs this way because the size some DTDs and documents would reach. Also it is much easier to reference a DTD from a location to multiple documents.

External DTDs are referenced in an XML document. The DTDs can be on a server or a file within the computer the technical communicator is using like the example listed earlier. The reference tag in the XML document links the document to the corresponding DTD. If the DTD is saved on a server, than the declaration will look like this, `<!DOCTYPE helicopter SYSTEM “http://www.helicoptermanual.com/dtds/IML.dtd”>`. The `DOCTYPE` indicates a DTD and then the root element is listed followed by the location of the DTD. If the DTD is saved in a folder on the hard drive, it would look the same except instead of the website there would be `“/dtds/helo.dtd”>`. The backslash lets the technical communicator know the file path of the DTD for the document. For DTDs saved in the same folder as the document, all a technical communicator would need is `“helo.dtd”>` instead of the website. External DTDs will be the most commonly used since they can be easily applied to multiple files and documents with little effort. Whether internal or external, DTDs give the technical communicator the power to create their custom tags for any imaginable document they need to mark up. For example, Lockheed Martin can create a custom tagging language to keep track of military helicopters in a marked up reference guide. Instead of having thousands of pages of documents with outdated aircraft, the information can be marked up and updated for anyone with access to view.

Schemas accomplish the same task as the DTD except they provide much more complexity to XML. Schemas are created with XML and provide a more powerful and expressive method of creating elements and attributes (Harold and Means 278). Schemas created with XML are coded in the XML Schema Definition (XSD). If there are multiple XML vocabularies being used then the technical communicator can incorporate multiple schemas. Schemas also provide restrictions on the number and sequence of child elements, create rules for the contents of elements and attributes, and uses namespaces to exchange tagging information from one person to another. All of these traits cannot be performed by DTDs. I must also note that XSDs are becoming more widely used in the TC field because of their capabilities. While there are still many legacy DTDs in the workplace, they are slowly being replaced by XSDs.

As mentioned before, schemas use namespaces. Because elements can have different meanings from company to company, namespace help distinguish elements of the same name. For instance, Lockheed Martin merges with another defense contractor and decides to combine both companies' information into one database full of XML documents. While organizing and parsing the information, Lockheed runs into the problem of the same tags with different meanings. Lockheed sees that the tag `<CMWS>`, which refers to the entire Common Missile Warning System (CMWS) at Lockheed, means only the laser component at the other company. If Lockheed Martin wanted to publish the XML document with the CMWS element, it may provide the wrong CMWS element and become not valid or provide the wrong information when published. While Lockheed could go through all the XML documents and change the tags associated with the CMWS, it would be much easier to assign them namespaces where XML applications could differentiate between the two tags.

Figure 5 provides an example of a schema for the tagging structure from earlier.



```
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<xsd:element name="helicopter" minOccurs="1">
  <xsd:ComplexType>
    <xsd:element name="helotype">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="manufacturer" type="xsd:string" maxOccurs="1" />
          <xsd:element name="model" type="xsd:string" maxOccurs="1" />
          <xsd:element name="engine" type="xsd:string" minOccurs="0" max Occurs="1" />
        </xsd:sequence>
        <xsd:attribute name="branch" type="branchType" use="required" />
        <xsd:simpleType name"branchType" />
      </xsd:complexType>
    </xsd:element>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

Figure 5: Helicopter.xsd Example

Figure 5 shows how the `xsd` namespace is used to tell the technical communicator that the XSD language is being used. Another option would be to use the `xs` namespace; however, both are approved by the World Wide Web Consortium (W3C). The important part of the namespace is the Uniform Resource Indicator (URI). The URI can be found by the web address provided after the `xmlns:xsd=`. The declaration of the namespace uses a Uniform Resource Locator (URL) to describe the physical location of the resource being referenced (Morrison 92). This makes the `xsd:` namespace unique to this document and helps in deciphering documents in case of a database merger between companies. Looking through the XSD, a technical communicator can see some similarities between the DTD from earlier and the XSD example above. Elements are created with the `xsd:element`, attributes are created with the `xsd:attribute`, and the highest level element in the document starts at the top of the XSD.

There are also many differences as well. In a XSD, the `xsd:schema` is the root element of the XSD and the helicopter element is its child. The XSD also uses `complexType` and `simpleType` to describe elements and attributes. The `complexType` refers to elements that are nested while the `simpleType` are used when no nesting is involved. For example, a `complextype` would be the tag `<helotype>`, which contains the elements manufacturer, model, and engine within it. The elements manufacturer, model, and engine would be considered a `simpeltype`. Another great feature of XSD is the `minOccurs` and `maxOccurs`. These statements allow the technical communicator to control the number of times an element or attribute can be used. As the example shows, in each sequence there can be only one element of manufacturer, model, and engine. The technical communicator would know that something is wrong if he or she saw two models listed in the model tagging, making the document invalid. The engine element also states that there can be zero occurrences if the information is not available.

The simple example of a XSD gives an idea of just how powerful the XSD or schema can be when used on an XML document. Unfortunately there is the downside of the schema; it is very complex to create. While many technical communicators will not have to create a schema, the ones that do might choose the DTD for its broader support and less complicated syntax. It should also be mentioned that there are many types of schemas including RELAX NG (RNG). RNG is supposed to have the same benefits of XSD but is less complicated to create. However, the technical communicator will usually not have the choice in deciding what format to use when working in the field.

DITA

DITA is much like a compact disk (CD) burning studio. You upload the technology, enter in the information you want on the CD, and burn the CD for whatever use necessary. DITA was created by IBM in the wake of failed attempts to create the universal DTD or schema. There are too many variables between the rhetorical choices of tags and sequences to create one all-inclusive DTD or schema. DITA has "unifying features that serve to organize and integrate information," making DITA the closest technology to providing the universal template for document creation (Day et al. 2). DITA was designed this way to provide an end-to-end capability, from authoring to production. DITA uses both XML and Standard Generalized Markup Language (SGML) for content reuse and creation. Unlike DTDs, there is no nesting in the root element. Instead the root element, known as the topic, relies on sections to support the topic and provide organization (2). If a topic is to be reused, it is referenced within the document using the topic. The generic setup of topics, elements, and attributes allows multiple content management approaches to be used with DITA.

So how does DITA relate to XML? Besides the use of XML as a language, XML allows DITA to be broken into chunks of data for mass use. DITA uses the generic tags of XML to create data that can be applied to multiple scenarios. For technical communicators that use DITA in their field, XML plays a big role in how they create content. Table 1 displays the IBM design of the layers in DITA.

Table 1 Layers of DITA

Delivery contexts			
Helpset	Aggregate printing	Web site; information portal	
Typed topic structures			
Topic	Concept	Task	Reference
Specialized vocabularies (domains) across information types			
Typed topic:	Concept	Task	Reference
Included domains:	Highlighting software programming user interface		
Common Structures			
metadata	OASIS (CALS) table		

The first group, delivery contexts, lists the deliverables for the topic information. The deliverables include printing, help systems, and web sites. The second group, typed topic structures, represents the four main content categories in the TC field. Typically documents will fall into one of these four categories. Each of these domains will have their own vocabulary. The reason for narrowing the categories to four is to provide a simple construct for the content type, making a more uniform language across the field. While there lacks a certain customization of tags within an XML document, the value of DITA is to provide a vocabulary that can transfer information quicker without the problems of combining documents with multiple XML vocabularies. DITA uses XML to extend tags between the domains, allowing for a template that fits many document structures. This is where group three, specialized vocabulary, is applied to the DITA model. The final group, common structures, refers to the presentation of body-level content in a document.

What DITA does for the technical communicator is provide a means to take specialized topics and associate the proper content with that topic through generic elements and attributes. Although the term generic is used, these tags are the most commonly used in XML and can be specialized for a specific document. DITA has its own DTD that can further modify the design of the document. With XML, technical communicators can use DITA to set up a general template for a document while having specialized tags for the data in the document.

TEI

Another option in XML tagging is TEI. TEI was established when libraries and museums were having trouble sharing their documents within their databases. The goal of TEI is to create machine-readable encoded text for the humanities and social sciences (tei-c.org). There are over 500 elements and 200 attributes within TEI, which are used to describe the document and the content within the document. For instance, some elements include highlight and quotation, graphical, closing salutations, rhyme schemes, and gestures (tbe.kantle.be). TEI itself does not have an official schema, but instead, requires users to select their elements and attributes for documents from the TEI modules (tbe.kantle.be). Once the user selects the elements need for a document, he or she can apply them to items TEI does not cover. This means that the tag sets are vaguer, which allows for a broader range of application. While technical communicators working in the humanities or social sciences field can create new tags through TEI, the tag creation would not be suitable for documents requiring incredibly specific tag applications with only one possible use (i.e., a milspec element used to tag military specifications on weapons and/or vehicles).

Why XML?

In the field of TC, a communicator is more than a writer. They not only create content, but share it across multiple platforms, make it accessible by multiple applications, and build information systems to store data. The technical communicator needs a technology that will meet all of these demands. XML seems to be the answer for these demands at the moment. XML is flexible, fully customizable, and has a very large support base. With XML, the communicator also gets to choose the tagging, template, and output of the document or data through DTDs and schemas. While XML appears to be the obvious choice in content creation, I will discuss in the next chapter more detail about how the TC field and XML are coexisting.

CHAPTER THREE: XML AND TC

How Technical Communicators View XML

XML is a technology that will increase efficiency in the transfer and organization of information for years to come. Whether you are a new graduate just starting in the field of TC, or an experienced professional, there is a good chance you will use XML in some way. Acquiring skills in new technologies like XML allows the technical communicator to operate productively in the current job market (Albers 336). Those who only work with text will have to learn the XML technology and vice versa since XML may become a standard for all technical documents in the future (Stolley 291). While many technical communicators will not develop DTDs or schemas, they will have to create or transfer content into an XML language for use within the company and for clients. However, the incorporation of this technology will not be an easy one. Some technical communicators are either afraid or do not want to learn a new way to create content. Others will welcome XML into their repertoire and use it to their full advantage. The third group will be in the middle, waiting to see if the XML technology will stay and how it could improve or hinder their current processes. This section will look at all three and how this affects XML and its tagging.

The first group will typically be made up of experienced communicators who have been in the field for 20+ years. They have been with a company for a long time and are comfortable in their methods of how to create a document for multiple companies. Michael Albers would describe this group as the craftsman model because this group of professionals thinks “they must expect to do everything,” and if they do not, “the concept of writing a book loses meaning” (Albers 335). These professionals are very good at their job and have the skill set to create high quality documents. While they do use programs like Microsoft Office Suite and Adobe products, they are very hesitant about using XML. XML makes them think that they will be “left out or run over by the machinery of efficiency” or put too much pressure on the stable constructs of their writing (Carter 318). However, Rodney Dick’s study revealed that communicators can learn new technologies through experience with other interfaces with similar commands (Dick 212). This means that the use of Dreamweaver or a similar company-specific program can help them learn the XML technology quicker.

Technical communicators in the first group take pride in creating the content for one or multiple manuals and consider the writing to be *theirs*. With the use of XML, many of these professionals find their writing to not be *theirs* anymore because the modularity of XML will require them to create content for only a portion of a manual, or create a neutral format for use technologies like single sourcing. They have the understanding that accepting XML will “[cut] into their ability to examine their practices critically” (Applen and McDaniel 6). XML helps with modularizing information—making information into reusable chunks—which may take away from the idea of this is my piece of writing. Instead of the technical communicator creating an entire document, the communicator would create modules of content for multiple manuals worked on by multiple communicators. XML also allows the sharing of editing material easier through programs like SharePoint. This capability leads writers to believe that posting the edited material “cause[s] everyone involved to further scrutinize the change, often resulting in its revision” (Jones 459). These key traits of the XML technology goes against the craftsman model these professionals are accustomed to.

In my professional experience, I have worked with one of these professionals in the field. He was close to retirement and typically passed off the projects involving XML to the younger staff that were still in college or just graduated. He would verbally state how he disliked working with the technology and felt it hindered his process of creating content. While many experienced professionals do not fall into this category, there are enough in the field to slow the incorporation of XML into the TC field. Locke Carter believes “excessive change may lead to ‘initiative overload, organizational chaos, and resistance to change’” when a technology like XML is incorporated into the document creation process (318). The professional I worked with would be even more resistant to the XML technology when he was forced to stick with a project that included the use of XML. His resistance would lead him to not consider the purpose of tagging a document in a specific manner or make suggestions to improve the structure of the content. Fortunately as the technology becomes more commonplace, I feel there are fewer professionals who will fall into this first group.

The second group is made up of all types of professionals. There are the experienced professionals who took on the challenge of incorporating SGML and HTML into their work when those technologies first appeared; there are the mid-level technical communicators who were introduced to the technology when they first started, and the junior communicators who have seen XML but not used it in a professional environment. This group is what accelerates the use of XML within the TC field. These professionals see themselves moving from mere communicators to information architects (Battalio 212). Not only are they just creating content, they are designing the rhetorical structure that will house the information. They work closely with subject matter experts (SME) to create their documents instead of the traditional work alone structure of the craftsman model. Their interaction with the SMEs allows the technical communicators to create better tagging structures for their documents because they receive a better understanding of the product than stand alone research.

Johnson-Eilola would describe this second group as "symbolic-analytic." This group "mediates between the functional necessities of usability and efficiency while not losing sight of the larger rhetorical and social contexts in which users work and live" (Johnson-Eilola 246).   In other words, these professionals use the XML technology to increase usability and efficiency without taking away from the rhetorical and social side of creating documents for mass uses. XML provides the efficiency and usability these professionals need while also providing rhetorical and social choices in the tagging. With more incorporation of XML into documents, having a rhetorical knowledge of tagging will be just as important as being able to write content for the document. The new acceptance of the XML technology will also create different jobs for the technical writer like information architect, knowledge manager, and programmer.

The third group of professionals will fall into the middle of group one and two. Not much research has been done in the TC field on this group of professionals, but my professional experience has introduced me to some of them. While they do not accept technologies like XML with open arms, they will work with it without resistance. These professionals still use the craftsman model to some degree and take pride in completing a project on their own whether in XML or not. If they do interact with other professionals on the project, it will be with the client for review and the occasional technical communicator when they have questions regarding the XML application. Most of these professionals are in the mid to senior level in their career and have a wide knowledge of multiple technologies for document creation. Their middle ground stance on the XML technology prevents them from considering the tagging structure or how they could improve on it from a rhetorical standpoint.

No matter where a technical communicator works, he or she will find at least one of these groups in the department. If the company has a large TC department, he or she may find all three. Figure 6 breaks down the groups and uses a Venn diagram to show their relation to one and other. Group one is the craftsman model and group two is the symbolic-analytic communicators discussed earlier. Each circle describes the traits of each group with a slight overlap. This overlap is the group three technical communicators who are on middle ground with the use of XML technology. The diagram shows how group three has traits from both groups yet does not lean more towards group one or two. However, with the growing use of XML in document creation, there should be a lot more groups 2 professionals in the workplace.

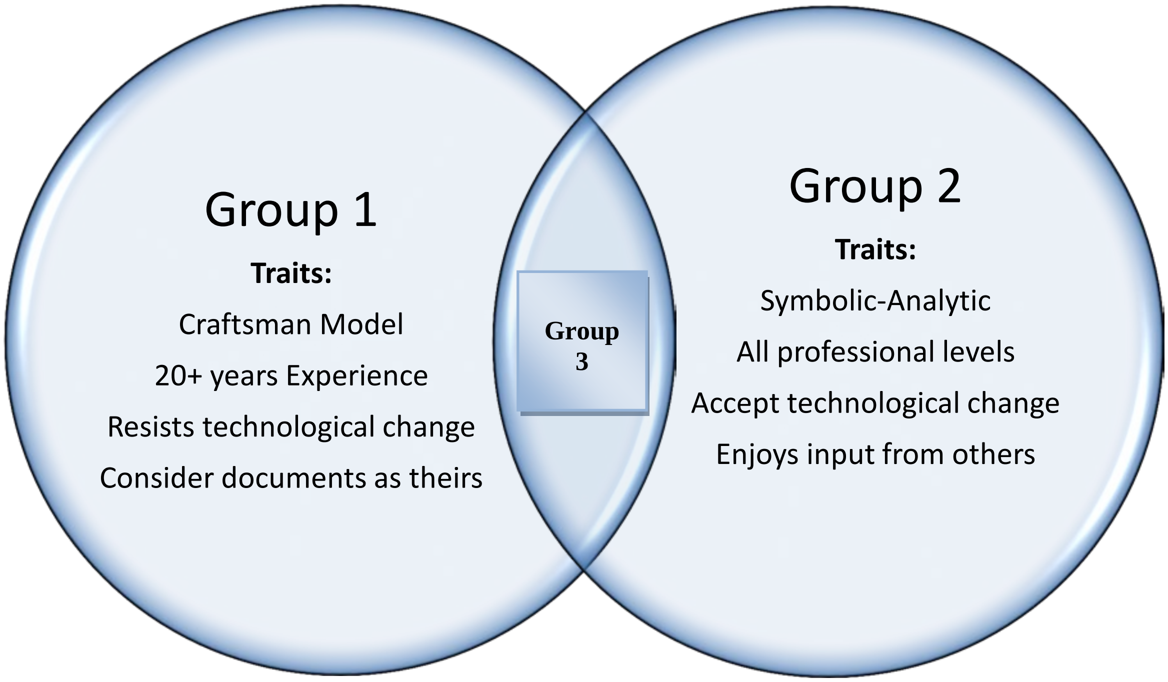


Figure 6: Three Group Traits of Technical Communicators in the Workforce

Incorporation in the TC field

With the introduction of XML by the W3C in 1996, XML has changed the transfer of information like no previous technology. XML provides the means to modularize data and transport the data across multiple platforms and devices. The one of the first industries to adopt XML were in financial institutions. Financial institutions share millions of pieces of information daily. This information is shared by investors, business owners, clients, and anyone else that needs to keep up with the fast-paced environment. XML help with sharing this information across different platforms and databases. While the coding language developed for the financial industry has been around for decades, XML help improve it with XMLs efficiency and versatility.

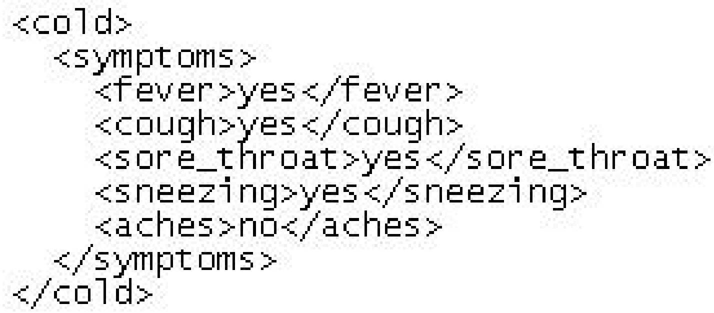
Lately the TC field has taken an interest in the XML technology because of the ability to modularize information and multiple publication outputs. Before XML, technical communicators were seen as regurgitating the information given to them by engineers and other SMEs. Technical communicators were not given any respect by their peers and often brought into a project late in the production of a document. With XML having a bigger role in document creation, technical communicators can use their abilities of "deal[ing] in *knowledge*" in a more efficient manner (Hughes 276). Technical communicators can take the *knowledge*—"presenting information in actionable terms and relating it to specific applications"—and combine it with XML to create more efficient information hierarchies, knowledge databases, and content management systems (CMS) (276).

One way XML is used in the field is with document preservation and re-use. XML is considered the top choice for document preservation with the goal of re-use, re-purpose, and re-presentation of content by communicators working with archives (Hodge and Anderson 59). While content creation is important for technical communicators, the ability to take content from one document and place it in another with little revision is just as important in today's company. There are also government agencies like the Office of Management and Budget (OMB) and the Chief Information Officers (CIO) Council who has technical communicators to preserve any documentation between the agencies in XML. If there are any questions with these preserved documents, they can be searched and brought up on whatever device is being used. The preservation also allows for re-use of specific parts of manuals to create updated versions like an OMB guide to creating schemas. Academia is also using XML to store and share information. Some schools are undertaking projects (University of Central Florida's Charles Brockden Brown Electronic Archive, Tufts University's Perseus Project, etc.) that use XML's capabilities to share with students and other users their archives of classic writings and other related information. With these projects in place, a student can now access and see where the information requested lies within manuscripts from Charles Brockden Brown.

Another task a technical communicator may have to do is create a documentation management system. XML is a good technology for sorting many documents and using them again like mentioned above. However, some companies also have documents that reach clients who do not speak the same language. Translation is typically very expensive, but XML allows the transfer of files with schemas that speed up the process and make it cheaper to do. XML also gives technical communicators access "to the many thousands of XML and image files included in the inventory of manuals" as well as "files for system maintenance, adaption of menus, and dialog boxes, style sheets for formatting the XML-tagged information into PDF" (Broberg 540). XML allows the technical communicator to control every aspect of the documentation process, not just the writing. Technical communicators who work with XML and documentation management systems may also have to code the SQL for the database; another skill gained by the technical communicator through the use of XML.

The last scenario illustrates how XML brings web development into the TC field with the Semantic Web. The Semantic Web is seen as a "highly interconnected network of data that could be easily accessed and understood by any desktop or handheld machine" (Feigenbaum et. keywords being used. Instead of 120,000 search pages to look through that may or may not be appropriate, the results would be a list of relevant pages based on the keywords used. XML tagging tells the machine this document is relevant and could have the information the user needs. With the introduction of XML and XHTML, technical communicators could make this a reality. First, technical communicators not only have to be able to write, but explore advances in structural aspects of digital writing (Stolley 291). Technical communicators will have to consider the tagging of modules for multiple output deliverables and multiple audiences when writing digitally. Digital writing also includes the tagging of information so that all devices can understand the information because the same document from a computer may not be viewable on a cell phone. XML's cross platform capabilities make it key in developing the Semantic Web and overcoming issues like multiple outputs and devices. The creation of these similar web pages will serve as important preparation for technical communicators learning XML (300).

While the Semantic Web is still in development, it could play an important role in the field of TC with regards to the transfer of information to companies and clients. Since some companies currently exchange documentation with clients and other technical communicators through the web and the intranet sites like SharePoint, technical communicators will need to know XML. The important part will be for technical communicators to understand not only the tagging of XML, but the rhetorical aspects of the tagging. An example of this would be the tagging structure in a medical database. There are more to symptoms than yes and no answers to whether or not a person has a symptom. A tagging structure like Figure 7 would not be efficient since the tagging does not provide a clear understanding of the information marked up.



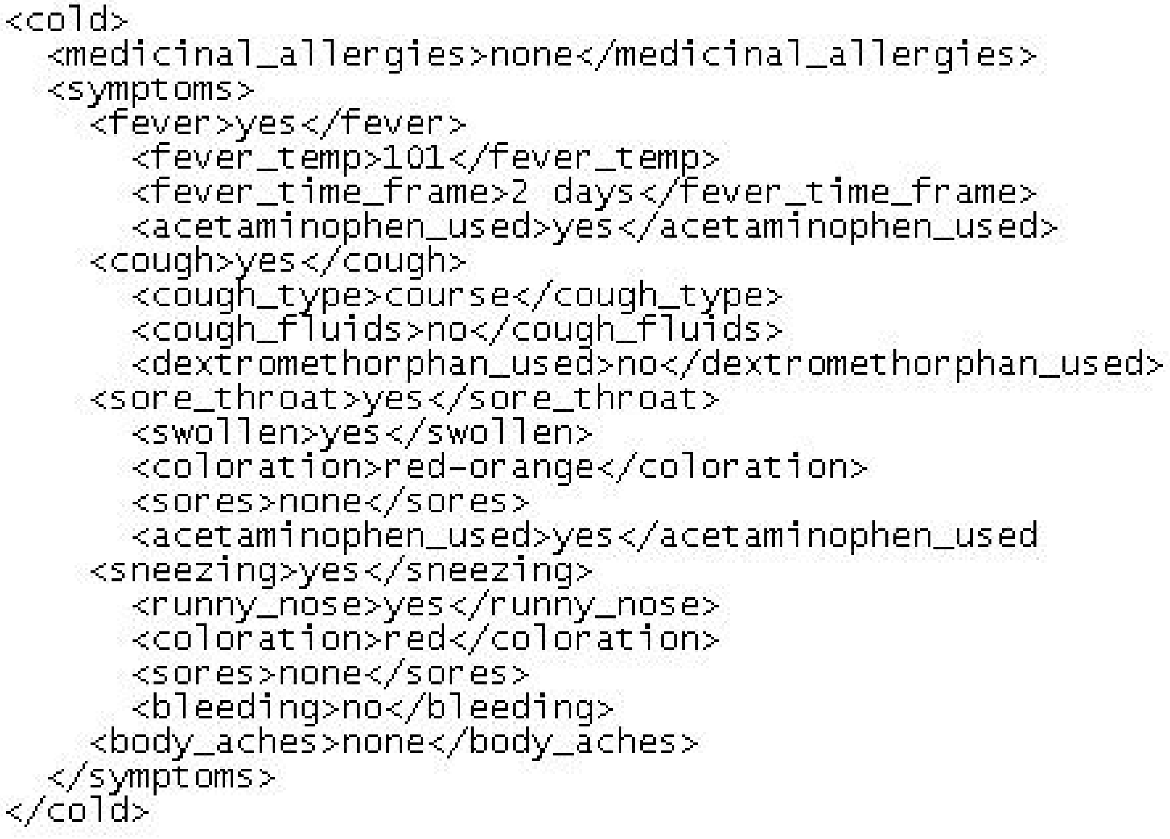
```
<cold>
  <symptoms>
    <fever>yes</fever>
    <cough>yes</cough>
    <sore_throat>yes</sore_throat>
    <sneezing>yes</sneezing>
    <aches>no</aches>
  </symptoms>
</cold>
```

Figure 7: Poor Tagging Structure

This tagging structure does not provide an in-depth understanding of the symptoms and would not be useful when searched in a database. A better rhetorical choice in structure would be Figure 8.

Unlike Figure 7, Figure 8 has a more rhetorical tagging choice because it provides the doctor with a tag set that can provide specific details and lead to a more accurate diagnoses. Figure 8 also provides a full description of the symptoms through the tagging structure and would be helpful to any doctor checking on the conditions of a patient or previous medical history.

These are just a few ways XML is incorporated in the TC field. XML has changed the technical communicator’s skill set from just creating content through knowledge to a number of skills including programming, archiving, and documentation system architect. Some important



```
<cold>
  <medicinal_allergies>none</medicinal_allergies>
  <symptoms>
    <fever>yes</fever>
      <fever_temp>101</fever_temp>
      <fever_time_frame>2 days</fever_time_frame>
      <acetaminophen_used>yes</acetaminophen_used>
    <cough>yes</cough>
      <cough_type>course</cough_type>
      <cough_fluids>no</cough_fluids>
      <dextromethorphan_used>no</dextromethorphan_used>
    <sore_throat>yes</sore_throat>
      <swollen>yes</swollen>
      <coloration>red-orange</coloration>
      <sores>none</sores>
      <acetaminophen_used>yes</acetaminophen_used
    <sneezing>yes</sneezing>
      <runny_nose>yes</runny_nose>
      <coloration>red</coloration>
      <sores>none</sores>
      <bleeding>no</bleeding>
    <body_aches>none</body_aches>
  </symptoms>
</cold>
```

Figure 8: Better Rhetorical Tagging Structure

XML based processes not mentioned above that technical communicators will be required to develop include data mining, single sourcing, and knowledge management. These three will be discussed in greater detail in the following chapters because they have a large impact on the social and rhetorical factors that go into their tagging structure.

CHAPTER FOUR: DATA MINING

What is Data Mining?

The term “data mining” is fairly new compared to the concept it describes. The technical definition of data mining is the “usage of classification, association rules, machine self-learning, sequential analysis, cluster analysis, and other statistical methods to seek out implicit, unknown, yet extremely useful information from massive and diverse databases” (Chang et al. 1434). What this definition means is data mining is the process of using programs and algorithms to locate patterns of data amongst information a company may be searching for in a database. The data can include sales patterns in the retail industry, purchase patterns in the credit card industry, or even the client responses to technical documents being used in out in the field. Data is mostly in written form, usually in XML markup, but there are also experimental technologies including verbal and visual formats found in the Agent-Oriented Modeling (AOM) algorithm. There are many options when choosing a data mining system and therefore only a brief overview of the workings of data mining will be discussed to keep in the scope of this chapter.

For years, companies have been trying to find an efficient way to locate information within their massive databases and many companies still use a hierarchy of folders to store their information. This makes it very difficult for employees or knowledge managers to locate the information they need because they spend a lot of time searching through folders with irrelevant information. Data mining transforms hidden knowledge into manifest knowledge which allows for the information to be transferred to the appropriate decision-making units (Chang et al. 1433). Chang states that most companies have their information stored in the traditional database made up of folders. Once a company decides to use a data mining system, a technical communicator or knowledge manager can mine the once hidden information. This will provide greater transfer of knowledge from the database to the professional and from the professional to their audience. Data mining technologies are typically algorithms that are programmed to search through XML documents (or other databases) for specific information. Depending on the parameters created by the programmer, the same data mining technology can yield different results. Technical communicators that work with data mining can create front-end interfaces that allow users to access databases as well as standardize that information from different systems (Le Vie 8).

TC is not the only field using data mining to increase productivity or make information more accessible. For instance, the financial industry uses data mining to determine customers' credit scores. They use an algorithm for benchmarking and credit scoring (Giudici 69). In the medical field, data mining allows for "the ability to store and access radiologic data (images and text reports)" which has "changed the practice of radiology during the last 30 years" (Erinjeri et al. 348). Data mining in the medical field "promotes the intercommunication of clinical documents between heterogeneous hospital information systems;" that is, the sharing of medical records between hospitals and physicians is more efficient because this information can be mined when needed instead of waiting for paper documents (Bond et al. 1). Finally, the humanities field takes advantage of data mining. The Networked Infrastructure for Nineteenth-century Electronic Scholarship (NINES) website relies on the ability to help users locate information. With "over 300,000 digital objects" in the database, users can mine for "peer-reviewed resources that had just begun to appear on the web" (Wheeles 145). They combined their data mining systems with highlighted search areas, popular search terms, and a prominent search blank to provide the user with a better experience (148). These are just a few examples of other industries that us data mining to their advantage.

Table 2 shows the four typical types of relationships (patterns) sought with data mining.

Table 2 Relationship Types

Type of Relationship	Definition	Example
Classes	Stored data is used to locate data in predetermined groups.	A technical communicator could mine customer complaints with a document and use that data to update the document.
Clusters	Data items are grouped according to logical relationships or consumer preferences.	Data can be mined to identify the client's needs for their documents.
Associations	Data can be mined to identify associations.	Technical communicators could find data that correlates the sentence length in procedures with comprehension.
Sequential Patterns	Data is mined to anticipate behavior patterns and trends.	A technical communicator can predict the likelihood of a document being understood based on the comments from a previous document.

The user of the program determines the type of mining. If the technical communicator wants to locate associations between documents, they will use the association relationship. An example of this would be finding data that correlates reading time and comprehension with print and online material. The sequential pattern relationship is used to acquire information on how an audience will react to a certain style of procedures. Data mining relationships are chosen based on the company's information needs.

Within these four types of relationships, data mining also consists of five major elements:

- Extract, transform, and load transaction data onto the warehouse system;
- Store and manage the data in a multidimensional database management system (MDDBMS)—a database that can answer direct questions from a user like "How many times has the warning section from Manual A been used in other manuals?" instead of a traditional SQL query;
- Provide access to content for use by technical communicators and data analysts;
- Analyze the data by application software; and
- Present the data in a useful format (Frand 3).

These five major elements break down the process of data mining. First, a professional may collect the information, mark up the information in XML, and then store it in a warehouse. The warehouse is an ideal vision of maintaining a central repository of all the organized data and allows for quicker analysis and retrieval of the data (2). Next, he or she will move the information/warehouse to a database management system (DBMS). The third element is where the professionals working with the information have access to the DBMS where the information is stored. Then a technical communicator will use a data mining algorithm/program to analyze the data and present the information in a format desired by the technical communicator and/or company. Figure 9 shows this process through a basic data mining diagram. The diagram shows how the data move to the warehouse, then the DBMSs, through the data mining software, and to the output. While this is a very basic data mining diagram, most data mining systems will be much more complicated like Figure 10.

As seen in the figures, the DBMS is the middleman of a data mining process. Without the DBMS to store the information, documents could get lost within folder hierarchies. Unlike XML, relational DBMSs need to have other languages (i.e., MySQL, C, etc.) to operate. These languages combined with linking between data help data mining programs locate information. This differs greatly when compared to XML and XML database models structure, which is an all in one package. XML is self-sustainable in the sense that the tags create a structure that can be searched and lead to other related tag sets. XML "makes available information about the structure of the data – as well as just the data – through the same interface" (Trotman xml.com). Creating an efficient and quality tag structure helps locate information with data mining algorithms even with mixed structures. Depending on the need of the company, either a relational or XML database model will be used.

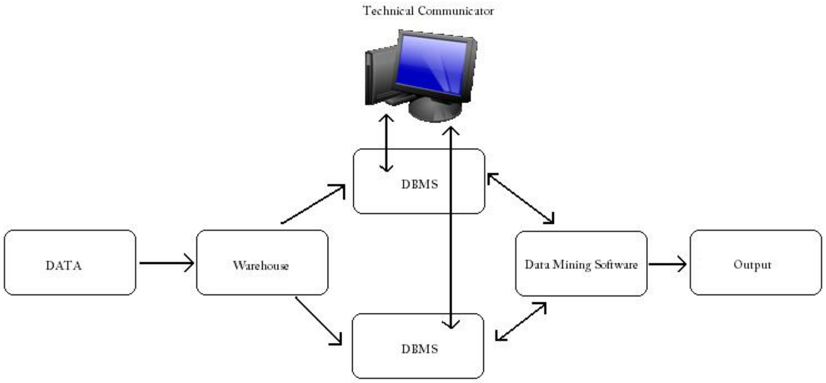


Figure 9: Basic Data Mining Diagram

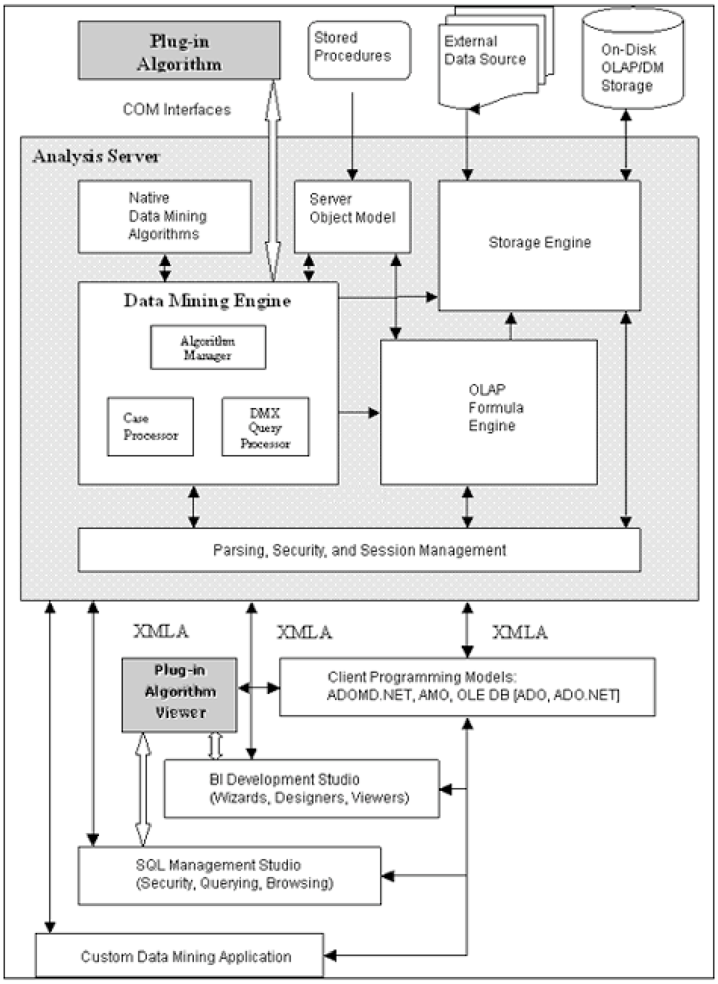

Plug-in Algorithm
Stored Procedures
External Data Source
On-Disk OLAP/DM Storage
COM Interfaces
Analysis Server
Native Data Mining Algorithms
Server Object Model
Storage Engine
Data Mining Engine
Algorithm Manager
Case Processor
DMX Query Processor
OLAP Formula Engine
Parsing, Security, and Session Management
XMLA
XMLA
XMLA
Plug-in Algorithm Viewer
Client Programming Models: ADOMD.NET, AMO, OLE DB [ADO, ADO.NET]
BI Development Studio (Wizards, Designers, Viewers)
SQL Management Studio (Security, Querying, Browsing)
Custom Data Mining Application

Figure 10: Complex Data Mining System (Microsoft Technet)

Unfortunately, data mining can be a very complex and expensive process for a company. Databases can range from 10 gigabytes to over 11 terabytes of storage space, and the more space a database needs, the more costly the data mining technology will be (4). Technical communicators also run into the problem of heterogeneities when working with multiple databases. Heterogeneities include differences in the operating system (OS), hardware, data models, access commands, and the way data relates or is similar (Pluempitiwiriyawej and

Hammer 301). Once these issues are solved, a company can take full advantage of a data mining technology.

Data Mining and XML

With the incorporation of XML, data mining has become a more powerful tool. XML's tagging structure allows for fine tuning of data mining algorithms to locate specific information. However, XML provides a new challenge for technical communicators and the data mining technology. Many times when XML tagging is created or deleted, the file is saved, overwriting the legacy of that document. It is not practical for companies to save every copy of every document due to cost and database restrictions; therefore, there is usually one historical document to refer to. Once a copy is saved, unless the technical communicator states what was changed in the tagging, nobody knows exactly what items were changed.

Ling Chen et al. named this complication as frequently and concurrently mutating structures (FRACTURE). FRACTURE is a set of substructures of an XML document that frequently change together (Chen et al. 320). FRACTUREs include structural deltas and content deltas. Structural deltas involve the insertion and deletions within the XML document, and content deltas are the changes of the values of nodes. These two deltas are the most common form of modification to XML documents. When a node is changed, the subtrees within that node are changed as well. Chen et al. states that the discovery of one FRACTURE within an XML document most likely will lead to other FRACTUREs within the ancestor relationships in the XML structure. This idea is like taking away a building's foundation; once the foundation is gone, it will affect the parts directly related to the foundation. When technical communicators use data mining to discover novel knowledge, they will typically run into FRACTUREs.

Data mining technologies will not only have to work with the tagging, but also the DTDs. DTDs determine the tags used within an XML document, which is what some data mining algorithms use to locate the information being requested. Some of the databases used by companies have a versatile class structure system to capture all meaning within a DTD (Tseng and Hwung 209). This means the tags created by the DTD are categorized within the DBMS. When the data mining algorithm looks for data, it will search the specific category the tag would be stored. Because the DTD and the data mining technology would have a one-to-one relationship, the data would be easily located for more efficient query result. Efficient data mining technologies are a company's main goal because it saves the company money. This means that a company who uses data mining will also use the technology to influence the tagging of the data.

Since the main goal of data mining is the efficiency in locating data, the tagging structure needs to be concise and logical. The tags should exactly describe an item and follow some sort of standardization within the TC department in a company. One communicator should not be tagging a part for a helicopter `<helo_part>` if all the parts are being tagged `<aircraft_part>`. This change in tagging could send the markup to a different category within the DBMS where a data mining algorithm might disregard the data. Since there are hundreds of data mining algorithms with specific commands, tagging associated with the data mining technology is very important. This is why data mining greatly influences how tagging is done within a company. The tags must be easy enough for employees to remember and understand, yet clear enough to provide the correct results when a user is mining for information. Technical communicators must also take into consideration that certain algorithms combine like tagging or may disregard tags that are too similar. While much of the algorithm is complex math and statistical equations, the tags and DTDs are still created by technical communicators. This puts pressure on them to create precise tagging for documents to provide the most efficient means for data mining.

They should also understand that XML documents are constantly changing, causing the FRACTUREs mentioned earlier. Unless the technical communicator frequently mines the data, some knowledge may never be discovered. For the professional mining data for structural changes, it will require them to parse document repeatedly which is the most expensive task in the whole mining process (Zhao et al. 644). The best way to cut costs is to create logical XML structures and tagging, where parsing can be done efficiently. Once again, the cost and efficiency play a big part in determining the tags and tagging structure of data in XML.

Data Mining Software

This section articulates a few options for data mining technologies. By no means is this an exhaustive list, but these three can cover the general aspects of data mining technologies available to the technical communicator. Because all three of these use algorithms to locate specific data, they all influence the tagging of the data so the queries can locate the information quickly and accurately like mentioned earlier in the chapter.

2PXMiner

This algorithm was designed to discover frequent query patterns by scanning the database at most twice (Yang et al. 375). Yang et al. also developed this algorithm to compete with the shortcomings of another algorithm known as FastXMiner. FastXMiner could not handle documents that contained siblings and scanned as many times as there are edges of the largest frequent pattern tree (376). What this means is that any XML documents that have siblings, which are most XML documents, would not be mined efficiently and the scanning capabilities of FastXMiner were slow due to complex pattern trees. 2PXMiner was developed to be able to accomplish these tasks, quick mining results and handle siblings efficiently.

2PXMiner uses three tools to accomplish this: a global tree that summarizes query patterns and stores them with an ID, a search tree that generates candidate subtrees, and index tree that tracks repeating siblings for quick tracing (376). While the specifics of the algorithm are out of the scope of this paper, choosing 2PXMiner over the FastXMiner results in the technical communicator being allowed to use sibling tagging and create complex trees without concern about performance. Even though concise tagging will be needed for quicker results, the tagging structure is not limited to what the data mining algorithm cannot do. This allows for more freedom in tag creation and structure.

XMine

XMine is slightly different than the traditional data mining algorithm in the sense that it takes into consideration the schema and how it relates to two similar elements names. XMine “determines the similarity between the heterogeneous XML schemas by considering the semantic, as well as the hierarchical structural similarity of elements” (Nayak and Iryadi 336). Analyzing both the semantic and hierarchical structures of schemas, XMine can locate high quality cluster results. These clusters of schemas help with the index structure, which help improve the speed and accuracy in retrieving data (337). An association rule can be applied to correlate relationships between the tags and schemas, leading to the discovery of the XML structures within the data.

XMine heavily relies on the schema and tags. If either are done incorrectly or have a bad choice for element tags, XMine could not operate to its full capabilities. For an individual using XMine, he or she would have to remember to tag items based on their association with other data tags. If two DBMSs merged together, the tagging would need to be similar to locate the necessary information in both databases. In either situation, the tagging structure would need to be accurate with the metadata the tags contain.

AOM

AOM was developed to break away from the tradition thought that all information is in the “Written Word” representation (Gu 434). AOM, designed for the linguistic field, provides a way of “conceptionally integrating multimodal data, while XML, TML, and RDF operationally integrate multimodal data (434). AOM focuses on what the sense can pick up, not just what is written down. For instance, if the technical communicator had to take notes in a meeting to be stored in a DBMS, AOM would allow for not only what was said, but also how people reacted to the statements. AOM provides a way for information like body actions and facial actions to be mined with the written data.

AOM also has an Agent-Oriented Modeling Language (AOML) that is used to mark up the data much like XML. This does not mean that AOM is not using XML. XML can be used with this algorithm to locate the data and structure needed by the user. Technical writers using AOM would tag items according to descriptions of the situation involving the expressions somebody makes or their tone. This would greatly help when looking for data on how a client reviewed a manual in a new format. Beyond what is stated, the technical communicator could also get a sense of how the client felt. Since AOM tags would need to be specific, all tagging is influenced by the situation of data. The mining algorithm may miss any data not tagged accurately.

Ethical Dilemmas

Data mining allows for technical communicators, business associates, doctors, and other professionals to locate information from large databases with great efficiency. However, with the promise of efficiency, data mining is particularly vulnerable to misuse (Evfimievski et al. 343). The technical communicator uses the data mining technology to locate information, which can include proprietary or confidential documentation. Any data that matches the search parameters in the data mining algorithm will be mined. This causes a serious security risk.

While the technical communicator will not intentionally seek out secret information within a company's DBMS, there is still a possibility of that information being mined by somebody else. There are two types of parties using data mining technologies, semi-honest and malicious. Semi-honest is not intentionally looking for private information while a malicious party is breaking protocol to find private information. Semi-honest is the party the technical communicator falls into. The technical communicator looks for information through data mining and may “extract any extra information from the messages that they see during the protocol execution” (Shah and Zhong 5468). The technical communicator is not looking for private information, but that information may appear within the search results.

A big problem with this private information turning up in search results is due to the XML tagging and schema. The goal of an XML document is to have accurate, simple tagging that will allow for efficient query when the data is needed. The schema determines the values of that tags and the technical communicator provides the information within the tags. This does not mean the technical communicator or tagging is wrong, but instead that the company needs to make the decision on what is more important: “the privacy of the individual data [or] the accuracy of the extracted results” (Magkos et al. 1224).

Since the protection of a company’s top secret information will outweigh the accuracy of query results, the company will have two categories for hiding their private data from data mining technologies. The first option is a “query restriction, which prohibits queries that would reveal confidential data,” and the second option is “data perturbation, which alters individual data” to the point where the data is relatively the same but not the exact data being hidden (Zhu et al. 133). The first option seems easy enough but does not work with all data mining technologies. The second works with more data mining technologies and can have a privacy setting that can “support various types of users with varying privacy needs without significant degradation in the quality of data mining results” (Liu et al. 5). Your clearance level would determine the amount of noise (distorted private data) your data mining result would present.

Another option that not many professionals have looked at is the technical writer and the tagging. While having great technologies to prevent a leak in private data sounds good, some companies may not have to go that far. A “common practice for protecting [data] disclosure is to remove [identifying] related attributes” from query data (Zhu et al. 134). This option leaves a great deal of responsibility on the tagging structures of XML documents. The tags themselves would still have to be efficient yet protect the identity of private data. The private data would also have to be tagged in a way that clustering and classification data mining technologies would ignore the data. Whether the company would want to leave the private data in notes within the XML document which could not be mined, or tag the data in coded tags that only top clearance personnel would know to program the data mining algorithm to locate the data would be up to the company. When reviewing the dilemmas of data mining, the technical communicator can see how they play just as big of a role in influencing XML tagging as data mining technologies do in the TC field.

Data mining is a powerful tool that spans across many industries including TC, medical, financial, and humanities to name a few. Its uses can be for locating information to determining a person’s credit score or provide knowledge to an employee researching a company’s business proposals. Data mining itself also comes in many forms depending on what information is needed to be mined and can lead to ethical issues if not used correctly. Overall, the success of data mining falls into the tagging structure of the information. Information tagged with quality XML markup will be more likely to be located from a data mining system. Technical communicators should be aware that their decisions on the tagging structure can mean the success or failure of a corporate data mining algorithm. The next chapter will look at single sourcing, where information is tagged in a database and mined by communicators for multiple uses later.

CHAPTER FIVE: SINGLE SOURCING

The Purpose of Single Sourcing

Single sourcing is often associated with the statement write once, use many times. J.D. Applen and Rudy McDaniel define single sourcing as “the practice of using one document for different outputs” (108). The idea of multi-use was formed when procedural manuals and other technical documents reused the same information, but created new content for each document. This leads to many inconsistencies on a topic that needs to be exactly the same for every document. An example of this is a warning section for an electronics company’s product line. Many of their products require a warning for not placing their equipment under water due to damage to the product and electrocution potential to the user. Because all of their products require this type of warning, all of the accompanying documents would have to have this warning in the appropriate document section. The company would want each warning to be exact from document to document; therefore, sourcing the same content for multiple documents guarantees the consistency the company needs. Single sourcing allows for using content in multiple documents by accessing a database and/or CMS as well as simply accessing spreadsheets and word processing documents.

Ann Rockley states there are four levels of single sourcing. The first level is called “identical content, multiple media.” Identical content, multiple media involves taking information from a single source and placing it in multiple outputs like a Word document and cell phone document reader. In the level one stage of single sourcing, “little attempt [is] made to differentiate the content or the presentation of the information to accommodate the differences in media and usage” (190). For instance, I described the Word document and a document program on a cell phone earlier. These two are common means for transferring information but have to be presented in different ways. While a full-page document with comments works for a personal computer (PC), the comments will not show on all cell phones. The only cell phones that will work are ones equipped with a mobile version of Microsoft Office, and even then the presentation needs to be altered for the differences in screen size and bandwidth limitations. Level one single sourcing may only be suitable for smaller companies where the output is limited to one or two similar outputs, like .docx and .PDF.

The second level of single sourcing is “static customized content.” Level two “moves away from traditional documentation (section, chapters, and files) to object-oriented information” (191). This type of single sourcing allows the technical communicator to customize the output based on the audience or client’s needs (help, PDF, webpage). The technical communicator will still use a legacy or core document for the information and the information itself can be customized (190). An example of this can be found with companies who use XML tagging to create their documents. Instead of creating documents by thinking in linear terms (Chapter, Heading, Subheading), the technical communicator creates the section modules independent of the other sections. The independence creates modules that can stand on their own and still make sense, allowing the communicator to pick and choose modules tailored to the specific needs of the end-user.

Level three single sourcing is called “dynamic customized content.” Dynamic customized content provides “on-the-fly” customized information for the user based on the user’s profile, content selection, and/or the combination of the two (191). In order to perform this type of single sourcing, technical communicators would need to have knowledge about the user accessing the information. The information provided would only be what the user needs to know. An example of this is if a worker needed to learn a function in FrameMaker. In a lower level single sourcing environment, worker would wade through pages of information in a manual, causing delays in completing the assignment. With level three single sourcing, the worker could just log into a database that knows what he or she is working on and provide the relevant information.

Level three single sourcing can also be found on the web. When customers of a website log on to look at products, the website might list possible items the customer may want to buy. The single sourcing, combined with data mining from Chapter 4, provides the user with only the information they may be interested in. The only way the level three single sourcing can work though is with a strong structuring of the data with XML and CMSs.

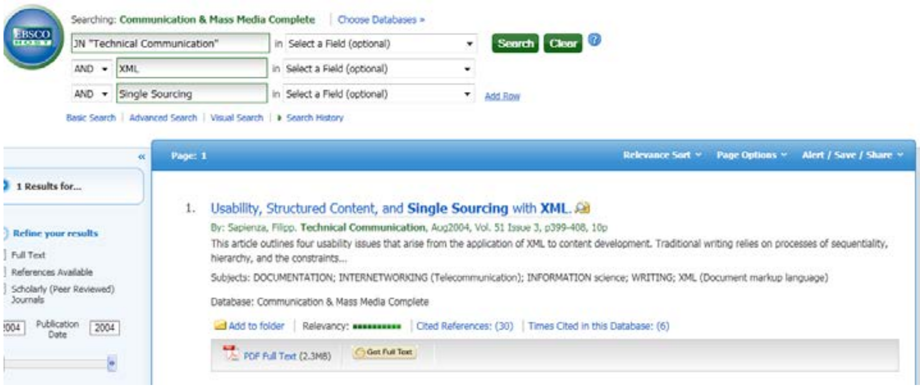
The final level of single sourcing is the “electronic performance support system” (EPSS). Level four single sourcing provides “‘just-in-time’ information based on the user needs” (191). What this means is that the information is provided to the user when they need it and sometimes “before they know they need it” (191). The example of the worker looking for a function in

FrameMaker from the level three single sourcing example works here also. The difference from the level three to the level four is that the information the worker needs would be presented right when they became lost or confused in the program. Level four single sourcing could detect that the worker has paused for a certain period of time without performing an action. A dialog box would pop up and suggest information based on the last actions of the worker, or have the worker select the information from the dialog box that would be relevant to him or her. While this level seems the best for any user, there would need to be a lot of research on the typical user, a very strong structure of information with appropriate tagging, and a database or CMS that could contain all the information needed.

CMS

When a company within the TC fields decides to use single sourcing for their documentation, it should consider developing a CMS. A CMS is a "system for transporting modular data from one location to another" and provides "search and access to the content stored within" (Applen and McDaniel 109). The CMS does this through the use of XML. XML allows for modular content to be stored, searched, and referenced in multiple documents, which is a "more sophisticated method of cutting and pasting content to generate coherent and consistent meaning across documents" (Robidoux 114). In order for the CMS to be efficient, the XML tagging must be done so that "you can use search routines based on metadata attributes, tag names (XML elements), and even full text search to find the appropriate pieces" to complete a document (Hackos 309). This means that the tagging should allow the technical communicator to locate the information he or she needs quickly without having to go through pages of information. Not only does the metadata become linked for document uses, but the hierarchies within a CMS are configured into modules and linked to other related hierarchies (Ament 23). Linking related hierarchies help with locating the metadata needed for a document and help with any data mining processes used by the technical communicator. An efficient structure will provide the best use out of a CMS for any single sourcing project.

A library's CMS provides a good example of how efficient structure and quality tagging help to locate information within hundreds of thousands of pages stored within a library CMS. A student who wants to locate a specific passage within the database will use a search query with keywords. The search engine will mine the data and provide a results list with the most appropriate showing at the top of the list. Figure 11 is an example of search results from a student searching within *Technical Communication* for an article that has both the keywords XML and Single Sourcing. The database returned one result which can be seen in this example. According to the library's CMS, only one article fits these parameters entered in the search boxes.



Searching: Communication & Mass Media Complete

JN "Technical Communication" in Select a Field (optional)

AND XML in Select a Field (optional)

AND Single Sourcing in Select a Field (optional)

1 Results for...

Refine your results

□ Full Text

□ References Available

□ Scholarly (Peer Reviewed) Journals

2004 Publication Date 2004

Page: 1

1. Usability, Structured Content, and Single Sourcing with XML.

By: Sapienza, Filipp. Technical Communication, Aug2004, Vol. 51 Issue 3, p399-408, 10p

This article outlines four usability issues that arise from the application of XML to content development. Traditional writing relies on processes of sequentiality, hierarchy, and the constraints...

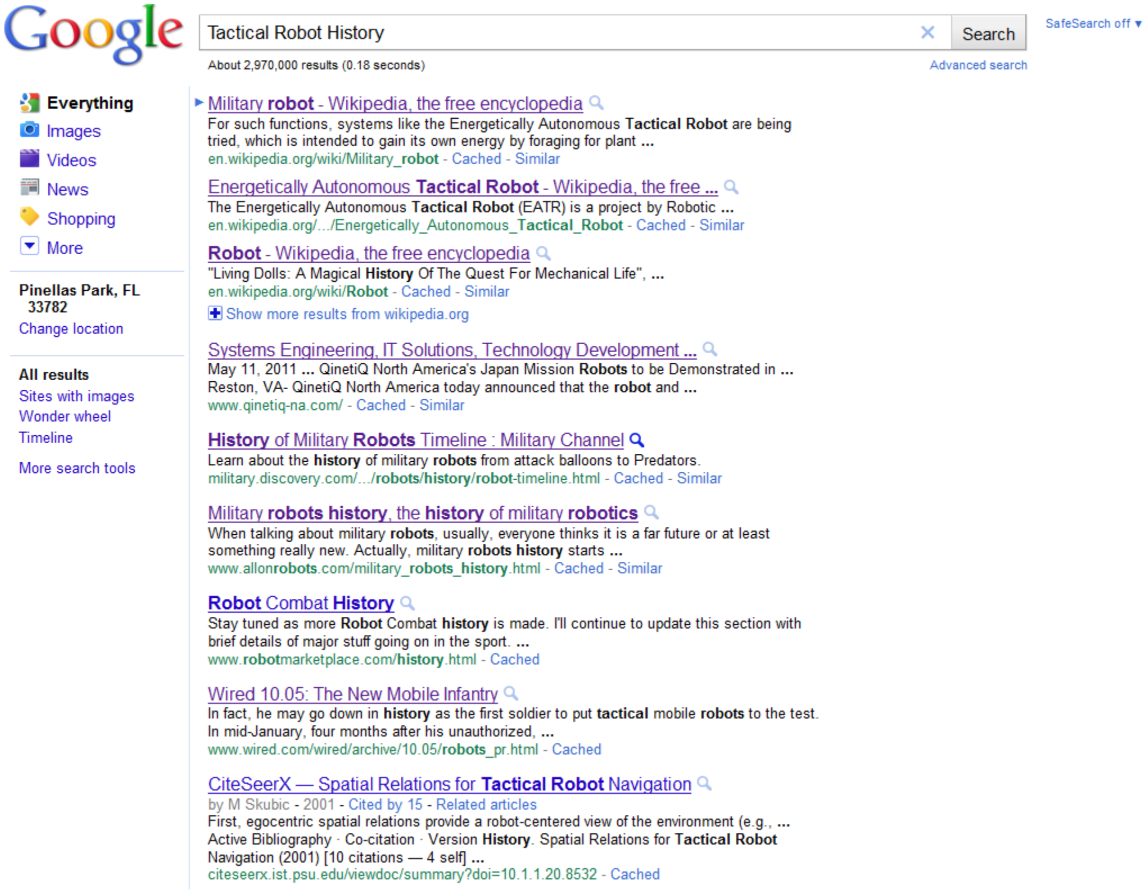
Subjects: DOCUMENTATION; INTERNETWORKING (Telecommunication); INFORMATION science; WRITING; XML (Document markup language)

Database: Communication & Mass Media Complete

Relevancy: ■■■■■■■■■■ | Cited References: (30) | Times Cited in this Database: (6)

Figure 11: Library Query Search Results

If done correctly, the CMS should be able to locate the information the student needs, assuming that information is structured within the CMS in the first place. Because the information is linked with other like modules in the hierarchy, many of the top results should have the exact information needed or the most similar information to what was being requested. If the CMS was created poorly, users would often receive the wrong information or information that is irrelevant. This is similar to some search engines that return results that may have a keyword in the description, but has nothing to do with the information being requested. Figure 12 is an example of a poorly structured CMS. Using a Google® search for Tactical Robot History, Google® displays a number of websites that may have one or all of the search terms entered. In Figure 2, the top result is a website that talks about military robots. The description underneath contains a sample of where one of the terms is located within the site. According to this, the website has tactical robot but nothing about the history of tactical robots. The results below the first also display how the words may be used within the website, but not in the context the person is looking for. Boolean searches can be used to help narrow results, but this often only narrows the amount of websites while still providing inaccurate results.



Tactical Robot History

About 2,970,000 results (0.18 seconds)

Pinellas Park, FL 33782

Military **robot** - Wikipedia, the free encyclopedia
For such functions, systems like the Energetically Autonomous **Tactical Robot** are being tried, which is intended to gain its own energy by foraging for plant **...**
en.wikipedia.org/wiki/Military_**robot** - Cached - Similar

Energetically Autonomous **Tactical Robot** - Wikipedia, the free **...**
The Energetically Autonomous **Tactical Robot** (EATR) is a project by Robotic **...**
en.wikipedia.org/.../Energetically_Autonomous_**Tactical_Robot** - Cached - Similar

Robot - Wikipedia, the free encyclopedia
"Living Dolls: A Magical **History** Of The Quest For Mechanical Life", **...**
en.wikipedia.org/wiki/**Robot** - Cached - Similar

Systems Engineering, IT Solutions, Technology Development **...**
May 11, 2011 **...** QinetiQ North America's Japan Mission **Robots** to be Demonstrated in **...** Reston, VA- QinetiQ North America today announced that the **robot** and **...**
www.qinetiq-na.com/ - Cached - Similar

History of Military **Robots** Timeline : Military Channel
Learn about the **history** of military **robots** from attack balloons to Predators.
military.discovery.com/.../**robots**/**history**/**robot**-timeline.html - Cached - Similar

Military **robots history**, the **history** of military **robotics**
When talking about military **robots**, usually, everyone thinks it is a far future or at least something really new. Actually, military **robots history** starts **...**
www.allon**robots**.com/military_**robots**_**history**.html - Cached - Similar

Robot Combat **History**
Stay tuned as more **Robot** Combat **history** is made. I'll continue to update this section with brief details of major stuff going on in the sport. **...**
www.**robot**marketplace.com/**history**.html - Cached

Wired 10.05: The New Mobile Infantry
In fact, he may go down in **history** as the first soldier to put **tactical** mobile **robots** to the test. In mid-January, four months after his unauthorized, **...**
www.wired.com/wired/archive/10.05/**robots**_pr.html - Cached

CiteSeerX — Spatial Relations for **Tactical Robot** Navigation
by M Skubic - 2001 - Cited by 15 - Related articles
First, egocentric spatial relations provide a robot-centered view of the environment (e.g., **...** Active Bibliography · Co-citation · Version **History**. Spatial Relations for **Tactical Robot** Navigation (2001) [10 citations — 4 self] **...**
citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.8532 - Cached

Figure 12: Google Search Query Results

While a complete breakdown of how to develop a CMS is not the focus of this thesis, I feel it is necessary to mention the "six phases of the development process—analysis, technology assessment, design, development, modification and implementation" (Pennington 65). In the analysis stage, the technical communicator must consider how the information is going to be used, the lifecycle of the document, and if the costs are worth the benefits. While most companies would like to have a CMS due to the long-term savings, the development of a CMS is expensive and may "prove prohibitive to organizations that might nonetheless be able to realize the production efficiencies" (Williams 322). The technology assessment phase is where the tools are determined. The choice of software would be decided based on the company's documentation needs. The designing of the CMS reviews the concerns from the analysis phase, and determines how the CMS will function, how the CMS will be incorporated, and how the metadata will be used within the CMS (Pennington 68). During the development stage, technical communicators create portals like SharePoint to provide training, information, and share documents with others within the company. The modification phase is when the CMS is tested. Technical communicators can find flaws in the system and make changes so the CMS operates smoothly and efficiently. The final step is implementation. This phase is where the system is fully incorporated for use by the company. These six phases provide an overview of what to expect when developing a CMS from scratch, and the development process can take months to develop a fully functioning CMS for single sourcing use.

Single Sourcing and TC

Single sourcing is providing a means to create information once and reuse it for multiple outputs. Although this new way of creating documentation is likely to save the technical communicator time and the department money, not all technical communicators have welcomed single sourcing.

Like issues discussed in Chapter 3 involving the craftsman model, some technical communicators will "initially …perceive structure writing as a restrictive and inflexible way of writing" (Rockley 351). These technical communicators get pride from creating their own documents and cannot imagine how "content somebody else created could possibly meet their needs" (352). Communicators might also feel that their writing is diminished because they are not producing an entire document (352). Another concern is that single sourcing would place "pressure on the seemingly stable constructs of the writer and document" (Carter 318). With the introduction of single sourcing to these communicators, they will be forced outside their comfort zone. The traditional way of writing a full document by themselves will become creating a module that will work for multiple outputs and audiences with a team of technical communicators. While these types of communicators are limited in numbers, they can pose a threat to a single sourcing project through resistance and making changes without informing the rest of the staff.

Fortunately, a majority of the technical communicators are welcoming single sourcing, understanding that the structured writing is a "way to free them from some of the more mundane components of writing and a way to improve the quality of their information" (Rockley 351). With single sourcing and XML, the technical communicator no longer has to focus on presentation. This allows for more focus on the content being tagged and the tagging structure itself. Technical communicators can think of single sourcing as a "reconceptualization of the relationship between audiences, purposes, and documents" and as a tool for practicing "writing and labeling content for reuse" (Eble 345). The scope of a communicator's job will be increased to include multiple media outputs like web, help, and PDF when writing content for a module (Rockley 192). Not only will the type of output be changing, but also the audience will include a variety of users. Technical communicators will need to know that single sourcing "is a more complex way to write and with complexity brings with it a requirement of specialization" (Albers 335).

Team projects will also be affected by single sourcing in the TC field. When teams create their documents, they "must negotiate social tensions and conflict as they work with others to create single source documents" (Breuch 344). Depending on the communicator's experience, each person will have his or her own opinion on how to structure the content and what tags will be used to create that structured content. Lee-Ann Kastman Breuch conducted an experiment with a team of technical communicators being introduced to single sourcing for the first time. Breuch's findings showed that single sourcing "yielded a better quality product" than the previous legacy manuals (350). Her experiment also found that modules are analyzed and critiqued more than in projects with large manuals due to the size difference, which provided a better examination of the writing (350). Many of her participants stated, "the process of writing in this way was stimulating and engaging and allowed them to become more fully immersed in the details of their writing" (353).

Single Sourcing and XML

As mentioned in the earlier sections, XML often plays a large role in successful single sourcing. XML provides a means to make information modular, which allows for efficient extraction of information from the CMS to the document referencing the data. DTDs control the XML tags as well as how the information is acquired from a CMS. The structured writing—the use of a tagging language to "describe the elements of a document based on its content rather than appearance"—makes efficient single sourcing possible (Eble 347).

Since single sourcing relies fully on being able to extract certain information for a purpose, XML is one of the preferred markup languages used to accomplish this goal. This is also why single sourcing influences XML tagging. When a technical communicator creates a module for single sourcing, he or she has to create "content that can be used in different products and label the content accurately so it can be used in various ways" (348). Sapienza reiterates this by stating "XML not only allows developers to produce structured content, it allows them to link content modules to interfaces such that content can drive the navigation that a user experiences" (400). Further, Sapienza adds that "not only does structure refer to the usability and navigation of a single-source document, but the semantic meaning of the text components themselves are derived from a conscious structuring method that parts from traditional approaches of linearity and sequentiality in document design" (85). Instead of just thinking in chapter, section, and paragraph terms, single sourcing requires technical communicators to create content that will work for multiple users as well as an "intelligent tag creation" where the names of the tags "have some identifiable and specific relationship to the content they contain" (98). Tagging the information accurately to form a solid structure is the only way single sourcing can provide the efficiency of write once, use many times and for multiple audiences.

Technical communicators must also have a "shared mental model and vocabulary" for easy extraction from the CMS (Williams 323). Most TC departments will have guidelines for how the tags will be structured, as well as the tagging vocabulary to mark up their technical documents. For companies that do not already have an established custom markup structure, DITA or DocBook can be used to create modular content. DITA and DocBook provide companies with an off-the-shelf set of tags that can be used to markup their documents (discussed in Chapter 2). This “flow diagram that all writers on a team follow” will “ensure quality control and similarities in voice, logic, and tone” (Sapienza 401). If these guidelines are not followed, technical communicators or the audience may have trouble in locating the information being sourced.

Unlike the copy and paste methods of legacy documents from the past, changes to the information do not require the technical communicator to go through all the documents to change the information. If a communicator makes a change in the core information, “the change need only occur one time and then ripple through the document framework” (Sapienza 83). All of the information that is sourced through XML structures will take on the changes that are made to the core XML module since single sourcing is using references to link the information through applications like Xref. For instance, when a module is used throughout multiple documents, the module is tagged with Xref. The Xref tag links the reused module to the core document where the original module was written. When information in the core module is updated, all documents with the Xref tag will update as well since they are all linked to the core module. The CMS will automatically update the information, and if the change does not need to be everywhere the information is referenced, the technical communicator may need to clone the module for another use (Robidoux 127).

Single sourcing is a valuable tool that technical communicators use to help make documentation more streamlined and efficient through the idea of write once, use many times. As discussed, this system relies on communicators embracing the technology, as well as a CMS to transfer the modular information from a database to the document. Markup languages like XML play a large role in the success of single sourcing. The information being used must be tagged correctly, so it can be mined and used in multiple documents. If information is incorrectly tagged or tagged in a manner where nobody can locate the information, single sourcing will not work to its full potential. As the next chapter will show, knowledge management plays a large role in how information is tagged that can be used for single sourcing initiatives.

CHAPTER SIX: KNOWLEDGE MANAGEMENT

What is Knowledge Management?

"Knowledge management" is a term that appears to have a straightforward meaning. The two words give the idea that knowledge management is the managing of knowledge in one format or another; however, unlike what was just stated, there is no easy or single definition that encompasses the entire meaning of the term. The reason is due to the fact knowledge management is "so broad that [it has] truly different meanings for different professionals, different meanings for different organizations" (Wick 1). Depending on the professional and organization, they may focus on the documents, technology, or the knowledge itself. This chapter will first define the types of knowledge, explicit and tacit. Next I will briefly discuss Corey Wick's four perspectives of knowledge management in order to lay a foundation for the possible categories knowledge management can fall into. Section 6.2 and 6.3 will analyze the tools of knowledge management and how knowledge management's use in the TC field. Finally, this chapter will discuss how XML is used in knowledge management and the influences knowledge management has on XML tagging.

In order to understand the different perspectives of knowledge management, knowledge itself has to be defined. The two types of knowledge are tacit and explicit knowledge. Tacit knowledge refers to knowledge that is built through experience and is deep within the human mind. This is knowledge that "cannot be taught to us by a book or trainer" (Applen 303). Quite often tacit knowledge is hard to communicate in a document, but can be communicated through social interactions. An example of tacit knowledge can be found in the way a market research company interviews potential manufacturers for a report for a client. While it may seem easy enough to gain knowledge about a potential manufacturer by asking the point of contact (POC) questions while visiting a laboratory, the POC will typically answer only what the question asks. If a researcher asks for revenues, the POC will strictly give revenue numbers while leaving out any lawsuits, recalls, or cancelled contracts (not publically available) that may affect the revenue. The researcher uses the explicit knowledge (discussed next) gained from sources to manipulate the POC into giving more information. The researcher will ask a question, and by the tone or actions during a response, can learn if the POC is leaving something out or how much information the POC is willing to share. In a sense the researcher "feels out" the POC to determine if more information can be obtained. This trait is something that can only be taught to other researchers through social interactions with the interviewer or witnessing the interview itself.

Explicit knowledge is knowledge "that is already known and written down or recorded in some other way" (Applen and McDaniel 20). This type of knowledge can be seen everywhere. One can find it in the form of a warning label on a pack of cigarettes, a primer explaining how to create DTDs and XML tagging, or on intranet—private computer network used to transfer information within a corporation—finance training classes for new employees that show them how to fill out a timesheet for payroll. In all these cases, the information is printed out or stored where someone can access it. While this type of knowledge seems to be the most prevalent, tacit knowledge is still the most common way of transferring knowledge. Just think of how many times you have asked someone their opinion about a topic or how to accomplish a task compared to you taking the time to locate a tagged source that could provide the same knowledge.

With knowledge categorized into two types, the next step is to determine how each type can be placed into one of Wick's four perspective of knowledge management. Table 3 provides an overview of Wick's four perspectives, their meanings, and the format for communicating the knowledge. As Table 3 shows, the four types of perspectives are document-centered, technological, socio-organizational, and knowledge organization. The document-centered perspective, most commonly used by technical communicators, gathers information to be transferred to an audience that needs or wants the knowledge. The transfer of knowledge can be through email, procedural manuals, websites, and any other medium that has the capabilities to transfer knowledge. Section 6.2 will look at the mediums that technical communicators use to transfer knowledge.

The technological perspective has the same goal as the document-centered approach, but focuses more on the technology to transfer the information. While "both approaches employ technological means," the technological approach "employs a far broader repertoire of technology" like artificial intelligence (AI) or Microsoft's Communicator (Wick 3). An example of the technological perspective can be seen during meetings that are being held through Office Communicator. All the participants are exchanging information/knowledge through the technology. Office Communicator, or the more modern Adobe Connect, allows for meeting rooms to be set up much in the same way professional reserve conference rooms for face to face meetings. The technology let users from different locations exchange knowledge as well as any files during the meeting without having to be physically there.

Table 3 Four Perspective of Knowledge Management

Perspective	**Definition**	**Format of Knowledge**
Document-Centered	"Place primary emphasis on extracting knowledge from individuals, analyzing it, synthesizing it, and developing it into documents that make it easier for others to understand and apply" (Wick 1).	Documents, Codified, Emails, Web
Technological	Information Technology (IT) takes the lead while documents provide a supporting role.	Intranet, Portals, Data Mining, Conferencing Software, AI
Socio-Organizational	"Emphasizes the social nature by which knowledge is shared" (Wick 3).	Human Interaction (documents and technology supports the human interaction)
Knowledge Organization	"An entity that realizes the importance of its knowledge, internal and external to the organization, and applies techniques to maximize the use of this knowledge to its employees, shareholders, and customers" (Wick 5).	Experienced Professionals, Social Interaction

The socio-organizational perspective uses human/professional interaction for the transfer of knowledge. For instance, receiving client feedback on a technical document during a meeting with the client is an example of socio-organization perspective. The technical communicator will discuss edits with the client, which will result in a better understanding of what was missing. While writing comments in the margin may have accomplished this as well, meeting face to face with the client gives a different perspective to what the client wants through gestures and tone. This approach is how tacit knowledge is transferred between employees and is a popular perspective for companies in the financial industry.

The final perspective is knowledge organization. This perspective understands how knowledge will affect the company as a whole, as well as its professionals, clients, and anyone else involved with the company. This perspective is much like the socio-organizational perspective in the sense that a professional creates knowledge that will affect all stakeholders in a company and transfers this knowledge through human interactions. A Chief Executive Officer (CEO) uses this approach to convince the board members, staff, and clients that his or her decisions are the best for the company and everyone involved. CEOs will meet with directors who share the information with lower management, and so on. Once the knowledge has been spread around, everyone involved can understand what is happening or what is going to happen.

KM Tools

For the purpose of this paper, I will only outline the tools used by technical communicators for knowledge management. As mentioned earlier, technical communicators use the document-centered perspective with regards to knowledge management. "Knowledge is codified using a 'people-to-documents' approach," which takes knowledge from the person and transfers it to another medium for use (Hansen et al. 108). Often communicators will take tacit knowledge and turn it into explicit knowledge through the use of multiple formats. These formats are the tools of a knowledge management system.

The most common knowledge management tool in the TC field is the document. These documents can include memos, help manuals, task oriented guides, or any number of other documents that a technical communicator creates to spread knowledge within an organization. The documents are created to provide other professionals with information they need to accomplish a task. For instance, a technical communicator that wants to know what the ethical procedures are for documentation within a company could refer to the code of ethics document. This document would provide knowledge that was previously unknown to the technical communicator and allow him or her to create documents that adhere to the organization's ethical principles. Not only would the code of ethics document spread explicit knowledge, but the technical communicator would also build tacit knowledge through the experience of creating many documents and learning which documents follow the ethical codes within the company. This experience could then be documented to provide information to other communicators in the company learning how to create documents that meet a company's ethical guidelines.

While documents are the main tool for knowledge management in the TC field, technology is also used. This can include the use of emails, Portable Document Formats (PDFs), PowerPoint slides, websites, and some of the technologies mentioned in this paper like data mining and single sourcing. The advantage to using these tools is the accessibility of the knowledge to the professional. A technical communicator can use single sourcing to link the knowledge to a document for multiple outputs. Another professional trying to obtain that knowledge can then mine that document for his or her own use. The technical communicator's use of documents and technology allow for a "natural" knowledge manager; in other words, the skills technical communicators use every day provides them with the ability to be effective knowledge managers if the company takes on a knowledge management system.

XML is also becoming an important tool for technical communicators and knowledge management. XML provides the means for professionals to locate knowledge and provide access to knowledge for anyone in the company. As mentioned earlier in the thesis, XML breaks information into modules to be accessed for multiple purposes. This information can be in the form of documents, emails, notes, and anything else that can be marked up and stored. Documents that are created can be tagged and stored in databases to be recalled by employees that need the knowledge. Through mining and single sourcing, these documents can be reused again and again. XLM tagging also provides a means for specific search queries that result in more accurate results when compared to web searches or traditional searches within folder hierarchies. An important factor in knowledge management is being able to access the knowledge. Quality and accurate tagging helps in searches to locate the information needed by the user. Specifics on how XML is used with knowledge management and how knowledge management influences XML will be discussed in Section 6.4.

KM Use in TC

In the TC field, knowledge management is the transfer of knowledge, both tacit and explicit, in a format which others who need the knowledge can access and use the knowledge. Technical communicators "place emphasis on extracting knowledge from individuals, analyzing it, synthesizing it, and developing it into documents that make it easier for others to understand

and apply" (Wick 2). This can be in the format of an email that instructs a new employee on what courses he or she should take on the intranet, a textbook that informs a student of the periodic table, or even a flyer that provides the code of ethics and ethical decisions within a corporation. The technical communicator must not only know the technology side of knowledge management, but also its social constructs. Sheng Wang and Raymond Noe point out that team members that feel left out from the main group, due to gender, race, or education, are less likely to share their knowledge with other team members (119). Team members who refuse to share knowledge could devalue a knowledge management initiative, preventing improvement within a team. The technical and social skills that a technical communicators uses to create documents, online help, and websites allows for an easier transition into the knowledge manager role and a greater understanding of how to process and acquire knowledge for use.

As a technical communicator, the main goal is to provide an audience with the knowledge they will desire or need to do their job better. The professional decides what information is pertinent and decides how the items should be ordered to allow the smoothest transfer of knowledge. He or she will discover the "importance of community, shared values and beliefs, language, and dialog" when breaking down information for knowledge acquisition (Wick 11). A communicator can design a document that is deemed effective; however, instructors have found that not all learners transferred the skills to the job (Hughes 368). A "performance-centered design (PCD)" is used to develop an interface that helps build knowledge through wizards, coaches, etc. (368). Developing a PCD provides a knowledge exchange interface—employee to employee and employees "searching knowledge from others"—that will allow employees to locate knowledge they need to accomplish their tasks or build skill sets (Wang and Noe 117).

An example of this can be found in a help system. The user has an issue and may not know exactly what to call the issue he or she is having. The first thing the user will do is access the database with a Boolean search. Depending on how the knowledge base is set up, the user may find exactly what he or she needs, or the user may have to go through the hierarchies in the index. A knowledge management system with knowledge properly tagged should provide the needed results through the Boolean search. If not, the index will need to be set up where the user can locate the knowledge based on what he or she believes the issue is associated with. If the technical communicator used broad terminology or lacks an understanding of the knowledge, the knowledge may never be located by the user. A communicator will have to develop the tagged structure of the knowledge that a majority of the team members can agree upon.

Studies show that an effective knowledge management system can reduce production costs; improves team projects, completion times, and new development; and increase revenue and sales for a company (Wang and Noe 115). Professionals use knowledge management to create documents so they can be used effectively and efficiently, or they can decide to use the personalization strategy; this is where the person who developed the knowledge shares it person-to-person (Hansen et al. 107). In most cases, one strategy is chosen and the other is used in a supporting role (109). In either case, the professional will need to analyze which would be better for the company and team. If a technical communicator chooses to use technology for the knowledge exchange, she must know that a person's history, context, and relationship with the technology will affect the knowledge exchange (Applen 301). Knowledge management evolves the technical communicator from the traditional position of breaking down information to a symbolic-analyst who makes meaning out of information and knowledge and understands the system and how it will affect the users (302). The ability to understand the social, professional, and historical context of the knowledge and the users seeking the knowledge, creates a more efficient tagging structure and query results.

Incorporation of XML in Knowledge Management

As briefly described in Sections 6.2 and 6.3, XML provides a medium through which knowledge can be exchanged to team members who need the knowledge. Tagged structures can allow for efficient searches of knowledge, provide modular data that can be reused in multiple documents in multiple outputs and can even provide an understanding of the tagged knowledge through the tags themselves. The key is that the technical communicator tagging the knowledge should have a full understanding of the contextual, social, and historical properties of the knowledge that he or she wants to preserve for later use. Knowledge management greatly influences the tag choice for marking up the knowledge since the technical communicator's final goal is to have a database where any team member can locate the knowledge they need.

For instance, a technical communicator will index the knowledge that he or she gains from the other team members. Should the communicator tag every specific sentence or just broad categories? Further, the professional must also designate the names of the tags so that team members can access the knowledge efficiently. While asking the team members what should be tagged and how may solve some confusion issues in that department, "different organizations (and branches within organizations) have their own take on what the information in their databases means" (Applen and McDaniel 125). Depending on the context, one organization can label knowledge one way and another organization might group that knowledge into another classification with different tagging. If these databases were merged together, it would be very difficult for all users of the database to locate the information they need. Alma Beatriz Rivera-Aguilera states that "the developing of structural content vocabularies must be a cooperative task among different institutions and disciplines in order to validate and give richness" to knowledge management initiatives (342).

Indexing the knowledge in the database makes it easier for team members to locate the information. In the case of merging companies and different classification systems, the technical communicator will have to look at tagging in a symbol-analytic way. The tags should encompass the meaning of the information with accepted terms, yet not go too far by tagging every piece of information or incorrectly tagging information to where the user will not get the knowledge they need. How knowledge is tagged should be apparent to users and not "lead prospective [team members] to believe that there are sources in existence that really do not amount to much" (Applen and McDaniel 123). This is seen when locating information through a search engine on the internet. The tagged information is in excess, providing sources that may have that element but not in the right context.

Because of contextual issues, technical communicators should also consider the "semantic elements that name objects so as to better enable [team members] to find relevant information" (Applen and McDaniel 124). Instead of using `<flu>` as a tag, which can provide thousands of documents that might not be relevant to the information the team member needs, provide the tag `<bird_flu>` or `<swine_flu_N1H1>`. These semantic tags provide the user with more information on the content, as well as better search results through Boolean cues. In addition to accurate tagging, the knowledge manager must know the audience and tag the content appropriately. While some scientists would prefer a tag of `<H1N1>`, research has shown that the "use of complex technical language that is not geared for general" employees can cause major complaints (Turns et al. 52).

Applen and McDaniel provide the following list of questions when deciding how to tag and structure the knowledge:

1. How many elements should be extracted from various bodies of information?
2. Which elements should be extracted?
3. Should the elements be extracted in their natural form or translated?
4. Should elements be in their natural order or constructed order?
5. Should generalization of individual concepts take place?
6. What are the rules that guide extraction (124)?

These six questions provide a good start to creating a tagging structure that will provide better query results for team members looking for information. Along with the tagging structure, quality is also an important feature for tagging in a knowledge management system. "Quality standards can apply to…indexing that permit others to access materials within the knowledge base," which coincides with a proper tagging structure (Turns et al. 54). The process of creating a quality tagging structure influences how the tagging is done due to the technical communicator constantly changing, editing, creating, and analyzing meaning to make the information accessible to the users (Applen 301). This can be seen as well in the dilemma above with deciding how to tag the flu in a database. The accuracy of the tag will allow everyone to gain the information needed about the flu or the poor quality can lead to misguided information or not locating any information in the first place. Technical communicators will take more time in creating the tags to predict how people will interpret the information being marked up, creating a knowledge management system that provides efficient results for its users.

Knowledge management is a way of collecting and making knowledge available for people looking for the information. A successful knowledge management system relies on the capabilities and technology used by the technical communicator. He must understand the context and social constructs behind the knowledge, as well as use technology including DBMSs and XML to store the knowledge for future uses. Once the knowledge is properly modularized, the information must be tagged appropriately. Quality tagging will allow users to find the information they need, when they need it. This means that the tagging structure is influenced by knowledge management, that is, the need to store and make information accessible influences the communicator to create a tagging structure that makes sense and provides the most accurate results possible. Chapter 7 will analyze how the same type of factors that affect knowledge management, political and cultural, also affect the tagging structure of that information.

CHAPTER SEVEN: POLITICAL AND CULTURAL INFLUENCES ON XML

This thesis has mainly discussed the rhetorical influences on XML, particularly how different technologies and the field of TC can influence XML tagging. As discussed earlier, we found that single sourcing, knowledge management, and other technologies that use XML greatly influence the tagging structure because these technologies rely on retrieving information accurately and efficiently. XML's capabilities as a markup language, which separates content from appearance, allows for modular information to be transmitted to multiple formats and uses and rhetorically controlled by the technical communicator through tagging. When creating documents, the technical communicator "produce[s] device-independent content: that is, identical content that can be shaped to the particular constraints of many different interfaces" (Sapienza 84). In order to accomplish this, the technical communicator must design a quality tagging structure that "anticipate[s] the different possible contexts of its reception" (93). Further, the "tag names themselves have some identifiable and specific relationship to the content they contain (98). Depending on the need and/or use of the document, the tagging structure may be different from technical communicator-to-technical communicator or from company-to-company. Technical communicators "often make different decisions about the use of headings, paragraph breaks, lists, tables, and the overall hierarchy" (Robidoux 112). These structural differences between authors transfer over to the semantic tags for content markup

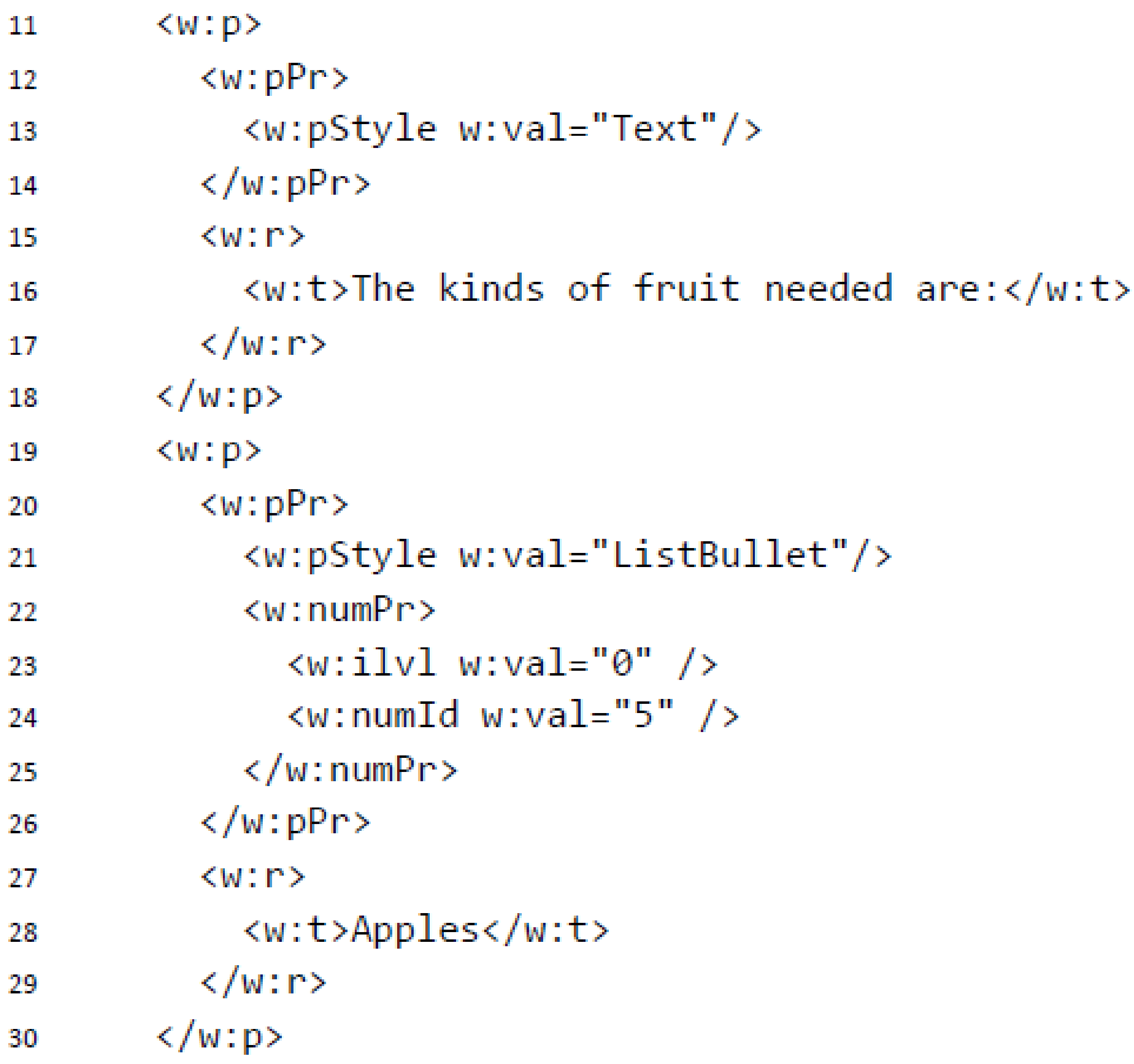
While the TC field, its professionals, and the technology those professionals use influence XML, so do political and cultural situations. Political situations can include product incorporation, the decision of tagging based on a company's needs, or other professional reasons that motivate a decision to implement XML and choose the proper tagging structure. The institutional culture help influence standards between different institutions in different industries, creating one standard that everyone can agree upon and work with.

Political Influences

Politics can be defined as a single authority or group of people that make decisions. XML tagging is influenced by people in society and the idea of a person creating tags for their own use or for a company database provides the creator with authority over the tagging structure. In most cases there will be a lead writer or group of writers that create the tagging and has final say on what tags should be used. From an industrial perspective (e.g., auto, banking, and TC), they are often deciding which XML structures work best for them and should they use off the shelf tags or create their own. Both predetermined tags and customs ones have their benefits, but the end decision will be politically motivated. Ultimately, the company states what tagging structures to use and who will be the lead authority on those structures. Their political reasons for choosing one XML standard over another can be for profit, updating old standards, or necessity to meet current technologies. This section will analyze each of these factors and explain how the politics determines the structure to be used and the need for a structure in the first place.

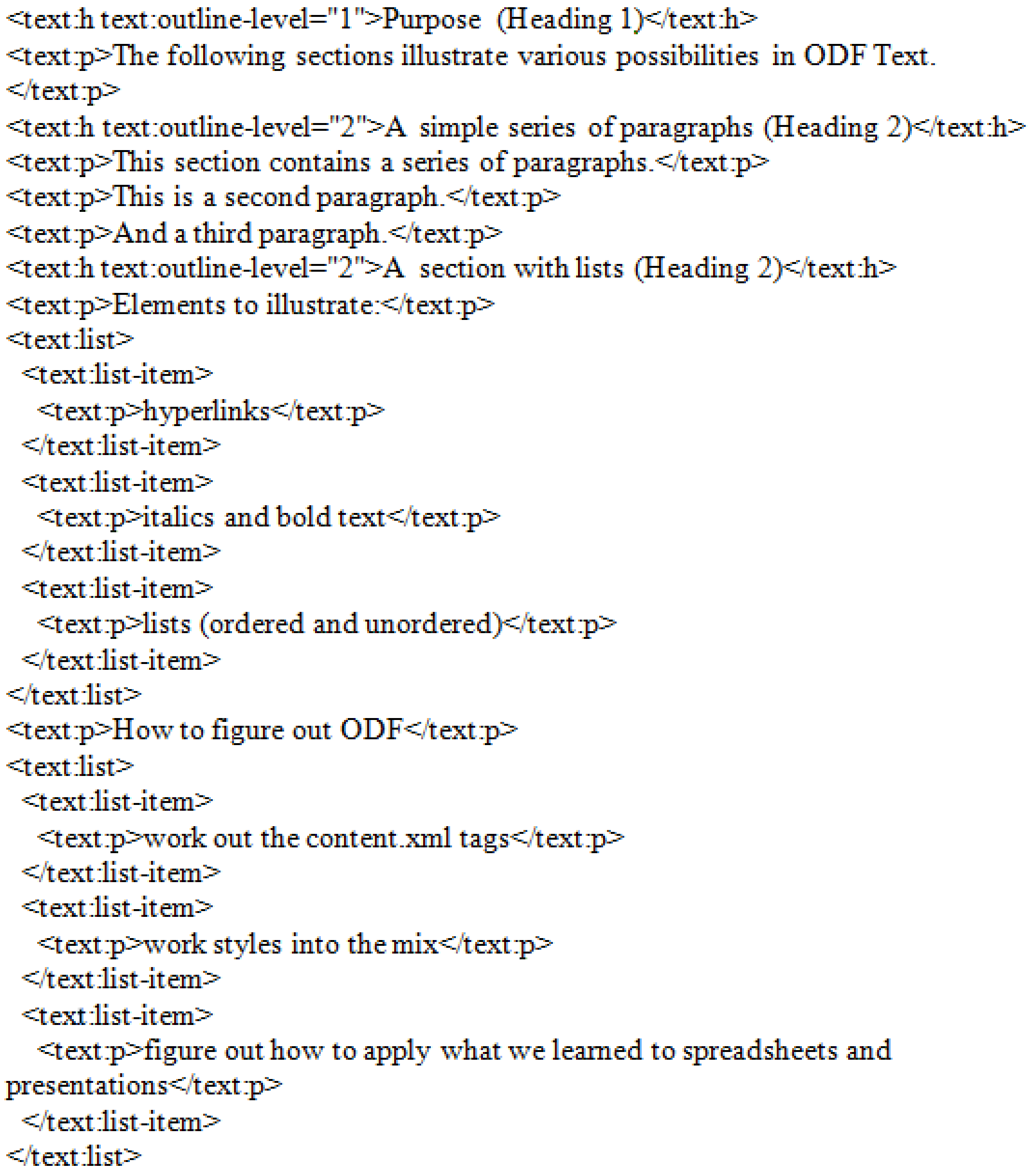
For profit is a large reason to implement XML into the documentation process for any company. As discussed earlier, XML can increase productivity which translates to more money for the company. XML allows the company "to store content in a media-neutral format and apply business rules to that content" (Berger 1). While some companies may be too small for an XML initiative, a majority of companies can find some savings in the long run by switching over to an XML format. The companies wanting to save time and money may just choose an off-the-shelf product with predetermined tags like DITA or DocBook instead of taking the time to create their own tagging structure. Creating a tagging structure from scratch is a larger investment than investing in a preassembled XML vocabulary, hence less profit. The profit influence provides a tagging structure that may work for the company's needs instead of a fully customized tagging structure that is perfect for the company's documents.

The profit aspect can also come from the developer's side, creating a proprietary XML set of tags or structure for companies to use. An example of this is Microsoft's Office Open XML (OOXML). Figure 13 shows a set of tags used to markup a paragraph and a bulleted list (Emca TC45 56). OOXML's tag structure is predetermined by Microsoft and is cryptic to someone seeing it for the first time. Additionally, OOXML's structure leads to interoperability tag issues with other systems without a patch designed by Microsoft. Unlike OpenDocument Format (ODF), the tags are not designed for humans to easily read and understand the structure. ODF's structure for a paragraph and bullets looks like Figure 14 (mashupguide.com):



```
    <w:p>
      <w:pPr>
        <w:pStyle w:val="Text"/>
      </w:pPr>
      <w:r>
        <w:t>The kinds of fruit needed are:</w:t>
      </w:r>
    </w:p>
    <w:p>
      <w:pPr>
        <w:pStyle w:val="ListBullet"/>
        <w:numPr>
          <w:ilvl w:val="0" />
          <w:numId w:val="5" />
        </w:numPr>
      </w:pPr>
      <w:r>
        <w:t>Apples</w:t>
      </w:r>
    </w:p>
```

Figure 13: OOXML Tag Set Example



```
<text:h text:outline-level="1">Purpose  (Heading 1)</text:h>
<text:p>The following sections illustrate various possibilities in ODF Text.
</text:p>
<text:h text:outline-level="2">A  simple series of paragraphs (Heading 2)</text:h>
<text:p>This section contains a series of paragraphs.</text:p>
<text:p>This is a second paragraph.</text:p>
<text:p>And a third paragraph.</text:p>
<text:h text:outline-level="2">A  section with lists (Heading 2)</text:h>
<text:p>Elements to illustrate:</text:p>
<text:list>
  <text:list-item>
    <text:p>hyperlinks</text:p>
  </text:list-item>
  <text:list-item>
    <text:p>italics and bold text</text:p>
  </text:list-item>
  <text:list-item>
    <text:p>lists (ordered and unordered)</text:p>
  </text:list-item>
</text:list>
<text:p>How to figure out ODF</text:p>
<text:list>
  <text:list-item>
    <text:p>work out the content.xml tags</text:p>
  </text:list-item>
  <text:list-item>
    <text:p>work styles into the mix</text:p>
  </text:list-item>
  <text:list-item>
    <text:p>figure out how to apply what we learned to spreadsheets and
presentations</text:p>
  </text:list-item>
</text:list>
```

Figure 14: ODF Tag Set

Although longer than OOXML, the tag sets are understandable and the user can tell when a list tag is starting or when text for a paragraph is being marked up. OOXML was designed by

Microsoft to compete with ODF, which was currently a less complicated, international standard. With so many industries moving to XML, any proprietary format with XML could make a large profit. Since most industries use Microsoft Office Suite to create documentation, OOXML was developed to create and convert Microsoft documents, presentations, and spreadsheets into an XML format that follows Microsoft's tagging structure. Unlike ODF, one needs to own a license to Microsoft's proprietary software in order to use OOXML to its fullest extent. While service packs have mended the gap between ODF structure and OOXML, the bottom line is if you want to use OOXML's tagging structure, you must own a license from Microsoft. Microsoft went to multiple countries to push their new OOXML format; the acceptance of OOXML by many countries has led to OOXML becoming an ISO standard through Emca in 2008 and creating large profit potential for Microsoft. Microsoft knew that many users have Microsoft Office documents saved within their databases, and therefore, would be more willing to convert to OOXML. With so many government and commercial industries turning to OOXML as a standard for tagging, anyone still using another tagging standard, like an open standard created by W3C, may run into interference when transferring information (Fonseca and Scannell 1). In addition to becoming an international standard, Microsoft used its influence to convince the state of Massachusetts to adopt OOXML over ODF for state documents, which will be discussed next.

A second political factor that influences XML tagging has to do with meeting document interoperability standards. In 2007, the state of Massachusetts decided to consider options for converting their government documents into an XML format. Massachusetts wanted a format that allowed for optimal document interoperability between departments and was a recognized international standard. The two formats the state decided on was ODF and OOXML, the latter not being an internationally recognized format at that point (2007). There were supporters for both formats. ODF supporters argued that associations have worked “to make the Open XML formats as useful and interoperable as possible” but “that can’t happen if they’re proprietary” (Aitoro 2). Gail Hodge and Nikkia Anderson further state that “[m]ost experts agree that the best format for preservation is that which is the least proprietary while conveying significant aspects of the original.” Because Microsoft is proprietary, Microsoft is “committed to its own XML” tags in OOXML which may not work with ODF tags (Hayes 40). The context for a tag in OOXML may carry a different context in ODF, leading to issues with transferring information between databases or government agencies. While some plug-ins have been developed to help transfer basic documents between the two XML formats, PowerPoint and Excel still remain a problem (Maleshefski 44). OOXML “defeats the purpose of Massachusetts’ open-standards policy, which has to support public access to documents and encourage choice” (Aitoro 2). Unless everyone has a license with OOXML, access would be limited to the public.

Supporters of OOXML believe that the supporters of ODF are just trying to keep Microsoft from entering the market. One supporter states “[i]f [the Office Open XML] format meets the open-source licensing standards and will do the job, I think they have complied with the state’s requirements” (1). The OOXML supporters believe because of the size of Microsoft’s customer base that more people would use OOXML since many residents and government agencies already own Microsoft products (owners of Microsoft would just download the OOXML add-on). OOXML also offers a predetermined vocabulary to tag the information created with the Microsoft Suite, limiting the customizability of XML tagging and structure arrangement. Because OOXML provides a predetermined structure—somewhat similar to DITA—the government feels OOXML will be easier to implement and provide a savings when compared to creating a fully customized tagging structure that would allow more flexibility when marking up documents for public access. As mentioned earlier, a customized structure (more expensive) would allow for tags to be created for the document while a predetermined tagging structure forces a document to fit into the available tags.

The final political drive behind XML tagging to be analyzed is the necessity of an industry. With XML becoming a standard to transferring and storing information, companies want to keep pace with the technology. When Adobe came out with FrameMaker 7, they included an XML capability. Since the program was designed to do "multichannel publishing," it only seemed appropriate to include XML capabilities (Berger 1). With a popular publishing program introducing XML to its users, the users have the opportunity to experiment with XML and create a tagging structure that works for them. The rules that the users apply to the XML tagging allows them to "determine how and where content can be used and distributed" (1). The fact that the tags "will be shaped by human choices…implies that [they] will be political" (Clark 3). No matter how much the technical communicator tries to be unbiased with the tagging structure, there will always be some influence on the decision to create a specific tag for a specific section of a document.

Around the same time that FrameMaker was including XML in its programming, the federal government was looking for a XML solution. Their current tagging structure was problematic because the government's "XML data structures and vocabularies" were divergent "among various agencies" (Thibodeau 16). The government needed a tagging structure that conformed with the private sector's structure, which the government interacted with continuously (16). This means that a standard would need to be agreed upon so that the private sector and the government could share information with little issue. The tagging structure would need to encompass both information that the government stores as well as the information the private sector has in its databases. For any documents that are preserved in the government database, the tagging structure would need to meet the "seven factors which the [National Digital Information Infrastructure and Preservation Program] NDIPP program uses to evaluate the sustainability of any given format" (Hodge and Anderson 46). The seven factors are the following:

- **Disclosure**- The degree to which complete specifications and tools for validating technical integrity exist and are accessible to those creating and sustaining digital content.
- **Adoption**- The degree to which the format is already used by the primary creators, disseminators, or users of information resources.
- **Transparency**- The degree to which the digital representation is open to direct analysis with basic tools, such as human readability such as a text-only editor.
- **Self-documentation**- Self-documenting digital objects contain basic descriptive, technical, and other administrative metadata.

- **External Dependencies**- The degree to which a particular format depends on a particular hardware, operating system, or software for rendering or use and the predicted complexity of dealing with those dependencies in future technical environments.
- **Impact of patents**- The degree to which the ability of archival institutions to sustain content in a format will be inhibited by patents.
- **Technical protection mechanisms**- Implementation of mechanisms such as encryption that prevent the preservation of content by a trusted repository (46-7).

The seven factors will determine whether or not a XML tagging structure can be used. If the structure meets all of the factors, then the government can use it. "Through tagging and style sheets, the content can be rendered in its native form" to be used by different government agencies or sent to private industries for use (59).

The final example of necessity is from the advertising industry. AdsML is an XML based markup language made for advertisement agencies. With all the physical documentation and projects filed away, there became a need to digitize the information for use throughout the industry. AdsML was the result of this need. At first, agencies discovered that "off-the-shelf use of XML" did not "satisfactorily address the need to have 'clean' solutions to identified problems, such as 'certain types of interaction between application system functionality and a generic exchange process'" (Christopher 5). While the organizations discussed earlier chose a predetermined set of tags due to cost and profit, the advertising agencies preferred a tagging structure that met their exact needs. The tagging structure would require that trading partners—advertising firms that exchange information—"agree on the types of information and the standards and formats each will use" (6). This makes sure that the tagging structure of one firm can be read by the software of another firm, affecting the tags used to markup the documents. The standardization of the tagging structure also allows for a universal schema to be used with all the agencies. Upgrades would include new tagging structures by firms using the AdsML. The need that drove the development of AdsML provided the advertising industry with a standardized tagging vocabulary and schema which "allows backward compatibility and can describe the structure and meaning of data content, as well as validate the document's conformance to ensure that the right data gets to the right place at the right time" (8).

Cultural Influences

Cultural influences can fall into a large category. The term cultural refers to the behaviors of individuals based on their environment. For instance, a person may use the term "pop" to describe a soda while another may use the term "coke." Within technical documents, how sentences are structured and word usage may reflect the cultural traits of the writer. The person's environment is not just limited to the location they live or grew up; it can include the field the professional is working in as well. Tagging in XML often reflects the cultural influences of the technical communicator and determines how and what tags are used to structure documents. Examples of this can be found in organizations like libraries and museums, where each has their own tagging structure designated by the terminology used in the corporate culture.

The Encoded Archival Description (EAD) was developed for archival aids to help users find the information they are looking for within library archives, as well as documents within the archives. With the large amount of documents in a library's database, searching can become tedious if the users are unfamiliar with the search engine. The EAD creates "indexable and searchable fields within a text of a document" through "[r]egularly structured and defined tags" (McCroy and Russell 99). This means that the tags used provide a way to search within documents without the need for Boolean cues, which helps new users locate the information they are looking for. Essentially, the success of EAD is based on accurate tagging structure set up by technical communicators or catalogers. The tags that are chosen for the structures are decided by "collaborating across departments, and to some extent, across cultures" (99). Depending on a department, professionals may define documents differently. McCroy and Russell note that professionals with different cultural backgrounds "has provided the opportunity to set standards, bridge differences in descriptive schemes, and build a base from which it is possible to work toward increasingly sophisticated delivery of information resources" (105). A tagging structure that is built on multiple cultures can find areas where descriptive elements are different, preventing issues later down the line when creating document structures and trying to share these documents between databases. This is important, especially when an archive is made up of many documents, images, and any other information that is electronically stored.

Another example where different business cultures affect tagging is with the Cataloguing Cultural Objects (CCO) initiative. The CCO is an attempt at creating an XML language that could encompass not only items within libraries, but also the items that may be found in museums. Both libraries and museums have their own markup language established; however, neither standard has the capability to be shared across databases. The issue arises from "decisions that catalogers make when describing cultural works are framed by the cataloger's perception of how a work of art is defined" (Coburn et al. 17). The cataloger's perception is based on the business culture's vocabulary used to describe items and may not "pass muster with an art historian" at a museum (25). One cataloger may tag a painting as <post-impressionism> while the art historian feels the painting should be tagged as <impressionism>. This will cause a problem when a union database—central database where libraries and museums keep their structured information—is created between the two institutions since one set of tags may overwrite the other, affecting the user's ability to locate the painting in the database. In order to solve the differences in cultures, catalogers should "tap the expertise of art historians, conservators, dealers, or collectors" to find the best descriptions of the items, leading to the best choice of tagging when structuring information for the union database (25).

The CCO had to also overcome the cultures of an institution as a whole. While the cataloger's culture may determine the tags for documents and information that are chosen, the tags themselves have to be approved by the institution. In the corporate world, "metadata comes from …the institution that owns the corresponding objects or items, and is therefore accurate and authoritative" (Baca 70). In other words, usually the institution that owns the item has the final say on what the item's metadata will contain and how that data will be tagged. This authoritative power allows institutions or catalogers to create a tagging structure that determines what is meaningful or useful to its users through search results in queries (74). With the adoption of CCO, libraries and museums would create a standard with "detailed guidelines on how to describe these unique works, and related images" while being "used as a compliment to the cataloging schemes already entrenched within [the] respective community practice" (Coburn and Paul 76). Adopting CCO creates a universal set of tags, which the institutions would agree on, while allowing each institution to not lose its cultural perspective on how the item's information should be tagged. The institutions will still be able "describe works of art in ways that are necessary and meaningful for their own work" and still provide "accurate sets of results when searching a collection" (78). Even though different institutions and the professional within the institutions have diverse cultural views on works, the CCO initiative creates a standard set of tags that encompasses the institutions' cultural views in addition to providing a better search experiences for users within the union database.

Do Political and Cultural Influences Hinder or Encourage XML Development?

When looking at political and cultural factors that affect XML tagging, one has to ask if these factors are good for XML. For both influences, the case can be made that they both hinder and encourage the development of XML tagging. In this section I will look back at the examples used in 7.1 and 7.2 to show how these influences hindered and encouraged XML tagging development.

In the case of profit driven influences (Microsoft's OOXML), this political factor greatly hinders the development of XML tagging. In the example earlier, Microsoft was pushing its way into the standards community to make their new format (OOXML) into a standard used by government agencies in Massachusetts. While Microsoft claimed the standard could be integrated into other documents tagged with ODF, using OOXML would require the user—or in this case the government agencies—to own a licensed version of Microsoft Office. Having OOXML as a standard used by the agencies would allow Microsoft to issue multiple licenses and build a profit. OOXML has its own set of tags (Figure 13) that are used in conjunction with Microsoft documents, and unlike ODF, is not open source to build and improve the tagging structures available. Microsoft does provide updates to the structures available; however, these updates are done by developers that work for Microsoft. Having a tagging structure that cannot be modified to fit the documentation being used takes XML a step back. The point of using an XML language is to structure a document so that it can be transmitted and accessed as easily as possible by users, which can only be done if the tags are built around the document. OOXML requires the technical communicators to structure a document in a way that Microsoft says it should, not necessarily in a way best for the user.

The other two political factors discussed, necessity and document interoperability, encourage XML tagging development. With regards to necessity, I analyzed how FrameMaker and the federal government needed to incorporate an XML tagging structure into their documents. In both cases, the industries realized the potential of XML tagging structures and developed tags that could be customized for their specific documents. FrameMaker allows the technical communicator to create tags for the document being used. This allows freedom of choice with tagging in addition to adding to the collections of tag sets available in the open source communities. For the federal government, all the agencies had different tag sets for their documents. Even though creating a single standard for all agencies sounds like limiting the tagging possibilities, the new standard will create more tags to be used on all government documents that were not previously available. With all agencies agreeing on a standard, anytime a new tag is needed that they see fit will be added to the set of tags to be used. Necessity and document interoperability takes a lot of work and planning to create the new tags but results in adding to the already vast vocabulary of tagging available to technical communicators.

For cultural influences, EAD and CCO both hinder and encourage XML tagging development. EAD and CCO created new tagging structures that can be used between institutions using a union database or for institutions just sharing information between each other. “New tags, or elements, can be structured as they are needed, and many communities” share these structures to allow users from different institutions have access to the information they need (Yeates 75). This creates new tags that were not previously available to communicators structuring documents, and therefore, encourages the development of XML tagging. EAD and CCO also allow developers to update and create better structures that are tailored to the end-user’s needs (87). If the developer finds that some information is not easily accessible to users of certain cultures, he or she can structure the information better to solve this problem, creating new tags in the process.

While EAD and CCO provide a means to further develop XML tagging structures, they also hinder it. CCO in particular, uses a standard set of only 23 tags, which “created a list that is extremely short [and]…diluted, to meet the pragmatic needs of those who are not” professional developers (Coburn et al. 24). Having such a short list of tags to structure such a broad range of document types and cultures creates the issue of fitting the document to a tag instead of creating tags around a document. Another issue arises with EAD and CCO is that most of the developers are not experts in tagging structures. Often, the expertise of scholars creating the tagging structure is assumed to have the abilities to provide high quality structuring, when in fact, some scholars are not aware that standards exist for that document or database system (22). Incorrect tagging will result in the document being rejected by the parser, or if the document does get placed in the database, not being pulled into a results list through Boolean searches. This goes against the purpose of creating documents in XML. However, further development of these standards may result in more tagging structures or limitations on who can develop the structures, which will help improve the quality of XML tagging structures.

How Political and Cultural Influences Affect Tagging in the TC Field

Earlier sections in this chapter provided examples of how politics and culture can have an effect on the tagging structures used to mark up documents for databases. The TC field also finds political and cultural influences affecting the tagging structures being created.

From personal experience at Lockheed Martin, XML tagging structures are currently being used to create documents from scratch. The technical communicator does not have any input to the tags being used, but instead follows a manual of the approved tags set by the government and military. This manual is over 450 pages long and attempts to have a tag for every possibility that can be found in the documentation. The tags themselves show the political and cultural influences enforced on them. The tags are abbreviated—much like how the military continuously uses acronyms in their documents and projects—or are short, one word descriptions like <email>. Many of the tags also have military or government terminology as the tags so that the technical communicators can associate the proper tags with the proper documents. The idea for this massive tagging structure is to have the capabilities to have one standard that fits all circumstances. However, even with over 450 pages of tags, there comes times when there is not a proper tag for a specific product. In this case, the technical communicator is told to find the tag that closely relates to the situation or the head of the department will discuss the issue with someone higher up. Any changes that occur in the tagging structure have to be approved by the military or government. This means that these agencies have the absolute authority over what and how the tags are used.

For companies that do not have an established XML tagging structure, it is an opportunity to create one for easier access to information. Companies will make sure the tagging structures are easy to understand since, more than likely, the staff will not yet be familiar with tagging documents. The structure will also be built to fit the documents being currently used. The culture of the company will determine if the tags will have multiple word tags like <government_standard> or single words like <govstandard>. The company will also have to consider the culture of its end-users and if the end-users will be able to locate the information the company tagged. Depending on how much they work with military or government agencies, the tagging structure may have to be approved by those agencies first. Politics will have a great influence on whether or not a standard will meet the agency's expectation or if the company's standard will be added to the standard already set by the government.

Political and cultural influences greatly affect how a tagging structure is used and developed. Political influences can come from government agencies, industries, or even the technical communicators themselves. We found that cultural influences can include more than just where a person comes from, instead, it can include the culture of a workplace and/or industry the tagging structure is being created for. Both political and cultural influences not only affect the creation of tags but also, in certain cases, can hinder or advance XML development. Chapter 8 will look closer at the future development of XML tags and associated technologies, as well as future political and cultural influences on XML tagging.

CHAPTER EIGHT: FUTURE FOR XML AND ASSOCIATED TECHNOLOGIES

Ever since XML became a standard in 1998, professionals have been manipulating the markup language for their own uses. From the rapid growth of XML customization, a number of DTDs/schemas were developed and more efficient use of data mining, knowledge management, and single sourcing were created. Further, no matter how simple or complex an XML based language is, we found that politics can still play a factor in deciding the outcome of XML development.

This chapter will look at the future of XML technologies, political factors, and XML itself. First I will make predictions on how standardization and RNG will play into the future of schemas and DTDs. Next the chapter will analyze how future data mining algorithms will be able to pull more accurate results with broad search keywords in formats that were previously not available to algrithms (video, audio, etc.). The following section (8.3) will be an example of a future knowledge management system utilizing XML, RDF, and ontologies. Section 8.4 discusses the future political and cultural influences on XML tagging while 8.5 provides future predictions of XML that include mobile messaging, Semantic Web, standardization, and scientific publishing. The chapter closes with the conclusion to the thesis, summarizing topics analyzed and any final thoughts.

Currently, there are not many sources that look into the future of the topics discussed; therefore, many of the ideas discussed in this chapter are purely speculative based on information found in this thesis.

Schema and DTD Predictions

With the first XML languages, DTDs were commonly used to set the rules for the markup. Through development of XML languages and the need for a more flexible and customized set of rules, XSD was developed. XSD, as discussed in Chapter 2, was developed by using XML. This allowed for a more customized set of rules that could provide a more powerful and expressive method of creating elements and attributes (Harold and Means 278). However, XSD is much more complex than DTDs, which means the user will need to have a solid grasp of XML in order to use the XSD effectively. Another issue arises when different businesses want to merge their information. Unless the business or industry uses a single schema, the information in the databases will not merge smoothly. We found in Chapter 7 that vocabulary is crucial in merging databases. Different cultures and different workplace politics utilize different tags sets for their information. This means that when the databases merge, some tags may overwrite other tags which keep users from finding the information they need.

The first prediction I would like to make is that DTDs will eventually be replaced by schemas. A lot of companies have devoted much time and resources into their DTDs, and therefore, want to hold onto them. However, there will be a point in the development of XML where DTDs will not have the customization power that schemas can offer. Most companies will make the transition in order to keep up with the development of XML. With regards to schemas, there will still be a lot of options available to the technical communicator tagging a company's information. Because schemas are customizable through the use of the XML language, unlike DTDs, there are thousands of possibilities.

The availability of schemas brings me to my next prediction, standardization. Eventually the number of available schemas will become a hindrance. Much like the websites on the Internet, the abundance of schemas will create issues when trying to merge the information they encode into one database. With companies like Google trying to mark up documents for any Internet user to search, standardization will be needed to make sure no information is overlooked during mining. Standardization could be accomplished much like a dictionary. This source could track all created elements and attributes while merging like tags to eliminate redundancy. In order for this to be successful, a standardization group would need to be created to specifically handle the enormous amount of tags. This source would be similar to DocBook or DITA where a user could search for tags that he or she needs for the document. If none are available, new tags could be created and added to the source. Due to the amount of tags in the source, there could be a search feature like what is found when searching databases.

Since creating a complete source for tags would be a daunting task and take years to accomplish with the number of tags being created every day, I predict the use of RNG will become more abundant. RNG is a compromise between DTDs and schemas; in other words, XML is used to create RNG like schemas but has the simplicity of DTDs. The reason that RNG can have both these traits is because RNG lacks many of the features that an XML schema has

(Vlist 4). RNG focuses on document validation, and because it is created with XML, still has the power to express almost any XML vocabulary. Further, a technical communicator can convert RNG into other schema languages. The fact that RNG is easier to write, generate, as well as easier for applications to use are good reasons RNG will become more common in the future.

As my predictions show, XML will still play a large part in creating rules to govern XML languages. Using XML to create XML tags is the logical way to create future schemas that allow for more flexibility and conversions. Like XML used today, the tagging structure will still play a key part in structuring information for other technologies to locate the information; therefore, even in less complex schemas like RNG, the technical communicator will need to consider how the tags are going to structure the information.

Data Mining Predictions

As discussed in Chapter 4, we found that data mining is used to search through tagged information to locate what the user is looking for or discover trends in what a user searches. Unfortunately, especially when using Internet search engines, the results are often not what the user is seeking. This is due to the amount of unstructured content on the web that is more readable by humans than machines (Stumme et al. 124). The issue is the structures. There are numerous structures that websites are designed with, which limit the web mining results. Some are tagged with only HTML, XHTML, or any other types of languages available for structuring information on the web. Sure et al. state that “finding and maintaining information is a tricky problem in weakly structured representation media” (2). As discussed before, the web mining algorithms are usually custom made to look for specific features when searching through databases or websites. The more the web mining program does, the more labor intensive and expensive it becomes. With the development of the Semantic Web, data mining will see some changes in the future.

The Semantic Web is an idea from Tim Berners-Lee which envisions the use of more machine-processable information (Stumme et al. 125). The Semantic Web is “a highly connected network of data that could be easily accessed and understood by any desktop or handheld machine” (Feigenbaun 90). This will allow for search engines to provide more accurate results because the data will point the machine to the correct location for the information being searched. The information will need to be coded with other languages, but the XML tagging structure will play a key role in managing the content for retrieval. Since data mining is used for locating information, a Semantic Web would be beneficial to this technology as well. Assuming the Semantic Web becomes an additional layer for the current World Wide Web (WWW) we are familiar with today, data mining would adjust accordingly.

The first change would be the type of information data mining algorithms will locate. Currently, this technology is used to locate textual information, some visual information, and find trends in how users look for items or relate purchases for websites. The future data mining could also locate information in the form of video and voice. Granted, the Semantic Web would improve on data mining capabilities with regards to textual information, trends, and images, but the big changes would be to locate specific information in sound and video. If a user wants to locate the lyrics to a song, he or she could mine the data to not only find textual information on the lyrics, but also the specific music video or the sound file for the lyrics. This can become useful when a user can only remember a couple words of a line, using common phrases, which would normally return a mass of useless results. The success of this mining would rely on the data having efficient tagging and having a tagging structure that can account for voice and video.

In the medical field, doctors' offices could also utilize future data mining algorithms. When a doctor is researching the symptoms of a patient, he or she could mine the information through a database and not only pull up articles on the ailments, but also pictures, sounds (if applicable), and video of what a person with this sickness looks and sounds like. The additional information mining algorithms could pull through a semantic database and/or web could provide quicker and more accurate diagnosis for patients in need. The benefits of the additional information could help many other fields, including TC, by providing more accurate and informative query results.

A second change the future holds for web mining is the possibility to return results for a broad search topic based on authoritative information (Stumme et al. 129). Stumme et al. mention that Google uses a similar technique where the top website is usually the website that has the most links to it from other websites. However, this can still return results that have nothing to do with what someone is looking for. Basing the result on authoritative information would provide better results in addition to listing the best source first. Since this would be very complicated to do, an algorithm could not do this alone. A person would need to be available to judge the context and accuracy of the information, tagging and placing it into a system where information can be ranked.

An example of this system could be seen in a large company with tens of thousands of documents in their database. For modern mining, a technical communicator would put in the search requirements and the mining system would return results; however, there may be hundreds or more documents with those search terms, requiring the communicator to sift through many documents. With an authoritative system in place, the communicators can rank the authority of a document by version number, resulting in the document that is most current. The tagging structure could inform the mining technology that this document is the most current version. Although there would still be many document possibilities depending on the search parameters, legacy documents would not interfere with the results.

Knowledge Management Predictions

In Chapter 6 we found that XML helps with organizing knowledge in a database. Knowledge managers take information, tag it, and store it in a repository for others to locate and use. The success of a knowledge management system depends in large part on the competency of the technical communicators structuring the information to tag it in an efficient and quality manner. If information is tagged appropriately, the user will locate the knowledge, and if not, the user will not be able to find what he or she needs. The future for knowledge management systems appear to be more complex. Since there are over one billion documents on the Web, the traditional ways of creating knowledge management systems will not work (Sure et al. 1). For this section I use Sure et al. research to show what the future holds for knowledge management systems. I agree their system would make knowledge management more efficient, forcing the necessity of proper tagging by technical communicators in order to function correctly.

As mentioned earlier, to counter the number of documents, whether the billions on the web or thousands in a company database, a knowledge management system will need to have different technological components beyond XML, DBMS, and a query system. There will still need to be the technical communicator to construct the hierarchies and tagging, providing meaning through ontologies. The purpose of ontologies is to "interweave human understanding of symbols with their machine processability" (2). This means the context of a word can be intertwined with the tagging structure, allowing for applications to understand the knowledge the same way humans do. An example of this can be seen in a company that keeps track of lessons learned from prior projects. These lessons are tagged and placed in a DBMS for review later. If a previous client makes a comment that he or she liked a part of a document, this gives little information on the degree the client liked the document. With ontologies, the degree can be associated with the statement through symbols that both the person searching for the information and the machine processing it could understand.

The next step would be information retrieval. Sure et al. propose a system similar to RDF*ferret*. This query system can be set up in several ways depending on the knowledge manager's needs. RDF*ferret* can provide results like a typical search engine (i.e., pull up results with and/or for the search terms) or pull up results associated with different classes. These classes can be determined by the user entering the keywords. The query system will then search through the tagged information, and through the use of ontology, will find results that have the same context as the user is searching. The contextual meanings in ontologies are set up in search fields by class, which is usually a drop down box the user can select from. For instance, the user searches for the term *document*. Document is a very broad term that would result in thousands of results in a large company's DBMS. The professional does not have time to waste skimming through all the documents. By selecting a class, for example project type, the documents would be narrowed down. This could be further narrowed by including classes based on client, timeline, or any other context associated specifically with the stored documents. RDF*ferret* is able to do this because the query obtains the content descriptors (words and phrases) from a text analysis and processes literal values that are related by property (4). Having ontologies interwoven into the tagging structure allows RDF*ferret* to pull what the user needs.

While having a query system that could know the context of what a user is looking for would save time and money for a company by providing only relevant results, a technology that already knows what each user will look for will make the process even more efficient. A user that runs into a problem would be able to search the ontology for the information they need at any moment: "ontology helps new users to navigate and act as a store for key learning and best practices accumulated through experience" (5). The user can modify the ontology to better suit his or her needs for the problems that may arise, making the ontology more precise in the information it can provide. Further, the user can make a custom ontology available to other users that are having the same issues. For new employees creating a marketing document for the first time will run into formatting and content questions. They would need to search the database for tagged information to help them solve the issue. If this information is assembled into an ontology, the professionals could just search through this information, allowing them to find what they need quicker.

Once information is located by a user, the information needs to be presented. We learned earlier in this thesis that XML is a structuring language. In order to present the information tagged with XML, we must use another language. For future knowledge management systems, Sure et al. recommend a technology like Spectacle. Spectacle does not just present information found in ontologies, but presents the information in a manner suitable for the user performing the search (5). This technology is also used to "disclose both the content of databases, document repositories and other enterprize information sources, as well as the semantics of that information" (5-6). Because Spectacle itself is a presentation platform customized for the user, the information Spectacle displays can be seen in a web style format (such as items in a list of results with links and pictures) or in a graphical format (can show relationships between results or allow analysis of the results) (5).

While this system seems like the best alternative for future knowledge management systems, it would require a lot of work to fully incorporate it into daily use. One issue would be that ontologies would need to be continuously updated. Meanings change between people and as time moves on, new terminology and context will develop within the professional community. However, the first step to making this system common use is the ability to structure the information in a way that it can be discovered. Technical communicators tagging the information/knowledge need to have a solid grasp on efficient tagging structures as well as how potential ontologies will work with the tagging structures. The future knowledge management system will rely heavily on the ability of the communicator for the success of the system as a whole.

Future Political Issues

Chapter 7 illustrated how politics play a central role in influencing the tagging structure of documents. Politics can include company policies, personal opinions within a department, or even the governance of a country. Further, politics can come from a major company forcing their own tagging structure on a group that uses the company's other products or a widely accepted not for profit organization pushing open source tagging structures. When it comes to the future of politics on XML tagging, I do not see much changing.

The reason I say this is because there will always be government standards for military documents, companies with their own tagging structures, for profit companies wanting to make a dollar off the mass use of XML, and so on. These types of political influences will not go away any time soon. For instance, whenever the federal government standardizes a structure for a specific section of the government (i.e., US Air Force (USAF), Internal Revenue Service (IRS), etc.), there are no questions asked. Companies working for the government will follow this structure for their documents or they will no longer be a part of the documentation process for the government. If technical communicators at Lockheed Martin decided to no longer follow the 400+ page XML tagging guide for their deliverables, the government would not accept their documents, causing them to lose out on future contracts.

Another example can be with industry structures. In Chapter 7 we discussed how the libraries and museums developed a tagging structure that allows both industries to share information in their databases. If one of the institutions decides to do their own XML vocabulary, they would not be able to share information with other museums and libraries using the standardized structure. This would cause many problems and extra work for the technical communicators retagging the information, as well as the users no longer being able to locate information they once could. The need for standardization and communal changes within industries are required to keep the information available to users. Standardization politics will always be around to influence the tagging within Industry.

One prediction I do make for politics is the use of more standardization. I mentioned in 8.1 that I believe a universal set of tags may one day be created. This could solve the issues currently plaguing structured information and transferring this information between different databases. Although it would require a lot of money, time, and expertise in XML tagging structures, it would be worth the effort to have all information available to users with the appropriate credentials (e.g., SECRET information could only be accessed by people with SECRET or higher clearance and approval from the source). Having a universal dictionary-like source for all tags could eliminate the needs for thousands of different XML languages that cause information to be lost during transfers or sharing DBMSs. The source could be searched like a database, and with the use of ontologies, the user can select the context of their document and industry. Current tags would be compared to other tags used for the same purpose, leaving a technical communicator with help from an expert in the associated field to decide which tag would be the most appropriate, efficient, and have the best quality for users looking for the tag. Tags with similar meanings that may overlap each other when searched will no longer be an issue. While admittedly this task would be enormous in size and perhaps impossible to accomplish at this stage of XML development, I think one day there will be a source for tags much like dictionaries for languages that are not limited by field (DocBook) or by the tagging categories (DITA).

Future for XML

The future of XML is almost limitless with the evolution of the language for multiple applications. Although XML will always be for structuring information, many developers are using XML with technologies not previously thought of, like cell phones. There are also further developments with scientific publishing and the long desired Semantic Web. These are just a few of areas of interest being researched for future applications of XML structures. These developments influence the tagging structure due to the needs of the technical communicator using XML for the specified purpose. That purpose determines how the tags will be used and how to create them.

Cell phones are everywhere, and like other technologies, they are capable of sending and receiving documents and/or text messages. People typically relate XML to computer based documents that can be looked at by smartphones, with the belief that XML structures are only on the DBMS side of the viewing. This belief is not correct, and for the last couple years, XML has been used to work with documents and text messages on phones. Cell phones use HTTP, Simple

Object Access Protocol (SOAP), and XML to create and transmit documents and messages. Unfortunately there is no standardized tagging structure for cell phones.

The roadblock for standardization is the verbosity of XML. The verbosity of XML "takes up quite a bit of the scarce available bandwidth" (Kangasharju et al. 29). Even if compression software is used, this will take a toll on the cell phone's battery due to the extra processes to decompress the information. In order to solve these issues, a binary XML (ASCII converted into 1s and 0s) will be used as the standard. Binary XML tokenizes attributes and elements to prevent the need for "string reading and construction" (30). This saves time and battery power during processing of messages. Processing is typically left to Document Object Model (DOM) and Sample API for XML (SAX); however, these both have flaws in their abilities. DOM automatically transforms the programming language into an XML structure, which may not provide the best layout for the structured information. SAX does not have standardization so developers will need to pick an implementation of SAX and stick with that. Although currently under development, StAX will be used to parse XML information. StAX takes pieces of XML one at a time from the parser to save on processing time. The combination of StAX and the tokenization of XML tagging will help reduce processing time for XML tagged information, as well as save on bandwidth and battery power. Further, tokenizing tags will lead to a standardized structure that all programmers can follow.

Another area for the future of XML languages is the scientific publishing arena. According to Reaner et al., scientists try to avoid reading an entire article and while total read time for articles has only risen slightly, "the number of journal article read per year has gone up"

(829). This means that the articles need to be stored in not just a structured format in XML, but also include other languages (RDF) and ontologies. With documents stored in XML, it will still require a specialized Boolean search to pull what exactly a user is requesting. A broad search within an XML DBMS will pull a number of articles, though fewer results than a search engine, which the users will need to read through to see if the articles are relevant. XML works with structure, not semantics; therefore, XML is not "suitable for defining logical relationships among terms" (830). Combining XML, RDF, and ontologies will help in getting the results to the right people with as little reading as necessary. Future XML structures will have terminological annotations mapped into the XML tagging, connecting "names and phrases in narrative text with appropriate standard terminology" (832). This is the same setup as what was discussed with ontologies and knowledge management in section 8.3, and appears that this will be the future, or possible standard, for XML tagging in the future.

The final future XML application will be in the Semantic Web. This idea has been mentioned several times throughout the thesis. The Semantic Web uses "software agents roaming from page to page can readily carry out sophisticated tasks for users" based on keyword entries (Lee et al. 1). The Semantic Web utilizes XML by allowing multiple platforms to share information and determine if the information found in a website is what the user is requesting. This sounds familiar since XML is known to be interoperable between different platforms; however, the previous paragraph pointed out that XML is only for structures. In order for XML to be truly semantic, XML will need other languages like RDF and ontologies. I reiterate these two technologies often in this chapter, but they are the future of XML structure as well as the

Semantic Web. The Semantic Web allows for users to find the exact information they need based on certain requirements. The Semantic Web would eliminate the days of 100,000 possible results, and instead, only provide the results the user needs. This is exactly what was discussed for the future of scientific publishing. For instance, *Harper's Magazine* utilizes a semantic setup for their articles. The website presents "annotated timelines of current events that are automatically linked to articles about concepts related to those events" (Feigenbaum et al. 91). This is helpful to Harper's readers because it provides a link to information that the readers may not know. This would provide context for the information and help the readers gain a better understanding of the current events.

Technical communicators can apply the XML tagging structure with RDF and an ontology for use within their databases. Users could search the database for parts of a document to reuse, assuming there is no single sourcing system in place, and have the search results only pull that section from all documents that currently use it. Another use can be for training and troubleshooting material stored in a database. When a user needs help with a task or trained on a certain program, the user can search the database and have only the necessary information show up. Instead of using only Boolean cues to narrow the search, a communicator could search the phrase "How to use lasso crop an image with transparent background" and the results would pull up instructions for that action in Paint. Clients could also use the DBMS to locate reports that are similar to what they are contracting the company for. This would allow clients to see the quality of work done for specifically those types of documents. For security concerns, the information can be encrypted and database searching would have to follow the security rules for the information within the database, preventing the stealing/sharing of personal or sensitive material (97).

This chapter has discussed what the future may hold for XML and the technologies that interweave XML's tagging structure within them. We found that the biggest change will come in the form of how XML documents are tagged. Future tagging structures will include ontologies and RDF technology to apply semantics to the structured information, resulting in more accurate results for diversified users. Another prediction was more standardization of tags on a large scale as well as for cell phone documents. Due to the steady evolution and development within XML tagging, it is hard to say what exactly the future holds for XML. The goal of this chapter was to provide some possibilities based on the information within this thesis and the limited scholarship done on the future of XML and the technologies that rely on XML for efficiency.

Conclusion

The purpose of this thesis is to analyze XML tagging and how the tags themselves are influenced when created. Many technical communicators take for granted that the tags are available to them and they only have to insert data into the tag set. Unfortunately, this is a close minded approach to tagging and limits the full potential of XML languages. If the professional wants to use XML to its fullest capabilities, he or she must understand what goes into developing a tagging structure.

In this thesis we found that there are many contributors to developing XML tagging. First, this thesis looked at XML itself, breaking down what XML is and the DTDs/schemas that set the rules for what the tagging structure can do. Understanding the basic of XML structuring allows for a better understanding of how the development of tags can be influenced. Then I assessed the many technologies (data mining, knowledge management, etc.) that take advantage of XML's capabilities, allowing for a more efficient system of locating structured information. I found that these technologies also play a part in determining how tags are created depending on the needs of the technology (i.e., if a knowledge management system is created to help new employees, the tagging structure should reflect terminology used with the associated task for easier query searches).

Further, political and cultural influences were looked at with several examples illustrating how they influence the development of XML tagging. The thesis discovered that not only can political, that is, government, influence tagging standards and structures, but that political influence from large corporations and department politics play a role. Often, it is more than just one political influence that plays a role in tag development. From personal experience at Lockheed Martin, I found that the tagging structure is influenced by in-house communicators, software packages, and government standards. All of these provide some political influence that determines what tags are to be used when structuring information.

Finally, I considered the future and made predictions of XML and the technologies associated with XML. The main theme for the future of XML tagging and the associated technologies is RDF and ontologies. These will provide tagged structures with the capabilities to not only provide relevant information more efficiently, but do so in the correct context, file type (.doc, PDF, mp3, jpeg, etc.), and in a more efficient manner. I also suggested that a standardized tagging source may be developed so that all users could access tags without having structural issues when merging or sharing databases between different users.

While this thesis is far from an in-depth analysis of every influence on XML tagging or the technologies that use XML, I hope this thesis will provide some direction into more scholarship on the influences of tag development. Current research on tagging influences is surprising limited in the field of TC, which is unacceptable considering the rapid adoption and use of XML within this field. If companies want to use XML tagging to its fullest extent, more research needs to be done in tagging development research. The more scholarship done on this topic will lead to more developments in the future of XML tagging.

REFERENCES

"Custom data mining algorithms." *Microsoft Technet*. Google Images. 2010. Web. 8 Oct. 2010.

Aitoro, Jill R. "Open-Source Battle; Massachusetts now hopes to support Microsoft's Office Open XML format, in addition to the OpenDocument Format. Leaders present their views as the discussion heats up." *Lexis Nexis*. VARBUSINESS, 6 Aug. 2007. Web. 14 Feb. 2011.

Albers, Michael J. "Single Sourcing and the Technical Communication Career Path." *Technical Communication* 50 (2003): 335-43. Web. 14 Sept. 2010.

Ament, Kurt. Single Sourcing: Building Modular Documentation. Norwich: Andrew, 2003. Print.

Applen, J.D. "Technical Communication, Knowledge Management, and XML." *Technical Communication* 49 (2002): 301-13. Web. 14 Sept. 2010.

Applen, J.D. and Rudy McDaniel. The Rhetorical Nature of XML. New York: Taylor & Francis, 2009. Print.

Baca, Murtha. "CCO and CDWA Lite: Complementary Data Content and Data Format Standards for Art and Material Culture Information." *VRA Bulletin* 34 (2007): 69-75. Web. 14 Feb. 2011.

Battalio, John T. "Extensible Markup Language: How Might it Alter the Software Documentation Process and the Role of the Technical Communicator?" *Technical Writing and Communication* 32 (2002): 209-44. Web. 3 Apr. 2010.

Berger, Matt. "XML presses the publishing business." *CNN.com*. CNN, 9 Apr. 2002. Web. 3 Apr. 2010.

Berners-Lee, Tim. "The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities." *Scientific American* (2001): 1-7. Web. 5 Oct. 2011.

Breuch, Lee-Ann Kastman. "A Work in Process: A Study of Single-Source Documentation and Document Review Processes of Cardiac Devices." *Technical Communication* 55 (2008): 343-56. Web. 14 Sept. 2010.

Broberg, Mats. "A Successful Documentation Management System Using XML." *Technical Communication* 51 (2004): 537-46. Web. 20 May 2010.

Carter, Locke. "The Implications of Single Sourcing for Writers and Writing." *Technical Communication* 50 (2003): 317-20. Web. 14 Sept. 2010.

Chang, Che-Wei, et al. "Mining the Text Information to Optimizing the Customer Relationship Management." *Expert Systems with Applications* 36 (2009): 1433-43. Web. 14 May 2010.

Chen, Ling, et al. "Fracture Mining: Mining Frequently and Concurrently Mutating Structures form Historical XML Documents." *Data & Knowledge Engineering* 59 (2006): 320-47. Web. 14 May 2010.

Christopher, L. Carol. "The Long Road from Concept to Implementation." *Analyzing Publishing Technologies* 5 (2005): 5-11. Web. 14 Feb. 2011.

Clark, Kendall Grant. "The Politics of Schemas: Part 1." *XML.com*. XML, 31 Jan. 2001. Web. 3 April 2010.

Coburn, Erin et al. "The Cataloging Cultural Objects experience: Codifying practice for the cultural heritage community." *International Federation of Library Associations and Institutions* 36 (2010): 16-29. Web. 14 Feb. 2011.

Coburn, Erin. "Beyond Registration: Understanding What Cataloging Means to the Museum Community." *VRA Bulletin* 34 (2007): 76-80. Web. 14 Feb. 2011.

Day, Don et al. "Introduction to Darwin Information Typing Architecture." *developerWorks*. IBM, 28 Sept. 2005. Web. 18 May 2010.

Dick, Rodney F. "Does Interface Matter? A Study of Web Authoring and Editing by Inexperienced Web Writers." *Business Communication Quarterly* 69 (2006): 205-15. Web. 3 Apr. 2010.

Eble, Michelle F. "Content vs. Product: The Effects of Single Sourcing on the Teaching of Technical Communication." *Technical Communication* 50 (2003): 344-54. Web. 14 Sept. 2010.

Emca TC45. *Office Open XML*. Emca. Web. 24 May 2011. < http://www.ecma-international.org/news/TC45_current_work/Office%20Open%20XML%20Part%201%20-%20Fundamentals.pdf>.

Erinjeri, Joseph P. "Development of Google-Based Search Engine for Data Mining Radiology Reports." *Journal of Digital Imaging* 22 (2009): 348-56. Web. 15 July 2011.

Evfimievski, Alexandre, et al. "Privacy Preserving Mining of Association Rules." *Information Systems* 29 (2004): 343-64. Web. 14 May 2010.

Feigenbaum, Lee et al. "The Semantic Web in Action." 297 (2007): 90-7. Web. 7 July 2011.

Fonseca, Brian and Ed Scannell. "Microsoft plays XML politics." *CNN.com*. CNN, 16 Apr. 2002. Web. 3 Apr. 2010.

Frand, Jsaon. "Data Mining: What is Data Mining?" UCLA.edu. Web. 2 Aug. 2010 <http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm>.

Giudici, Paolo. "Bayesian data mining, with application to benchmarking and credit scoring." *Applied Scholastic Models in Business and Industry* 17 (2001): 69-81. Web. 15 July 2011.

Gu, Yueguo. "From Real-Life Situated Discourse to Video-Stream Data-Mining." *International Journal of Corpus Linguistics* 14 (2009): 433-66. Web. 14 May 2010.

Hackos, JoAnn T. Content Management for Dynamic Web Delivery. New York: Wiley, 2002. Print.

Hansen, Morten T., et al. "What's Your Strategy for Managing Knowledge?" *Harvard Business Review* (1999): 106-16. Web. 14 Sept. 2010.

Harold, Elliotte Rusty and W. Scott Means. XML in a Nutshell: A Desktop Quick Reference. Sebastopol: O'Reilly Media, Inc., 2004. Print.

Hayes, Frank. "'Office' Politics." *Frankly Speaking.* Computerworld, 26 May 2008. Web. 14 Feb. 2011.

Hodge, Gail and Nikkia Anderson. "Formats for digital preservation: A review of alternative and issues." *Information Services & Use* 27 (2007): 45-63. Web. 14 Feb. 2011.

Hughes, Michael. "Mapping Technical Communication to a Human Performance Technology Framework." *Technical Communication* 51 (2004): 367-75. Web. 14 Sept. 2010.

Johnson-Eilola, Johndan. "Relocating the Value of Work: Technical Communication in a Post-Industrial Age." *Technical Communication Quarterly* 5 (1996): 245-70. Web. 3 Apr. 2010.

Jones, Scott L. "From Writers to Information Coordinators: Technology and the Changing Face of Collaboration." *Journal of Business and Technical Communication* 19 (2005): 449-66. Web. 3 Apr. 2010.

Kangasharju, Jaakko, Tancred Lindholm, N. Sasu Tarkoma In Anerousis, G. Kormentzas, eds. *Requirements and design for XML messaging in the mobile environment*. Second International Workshop on Next Generation Networking Middleware, 2005. 29-36. Print.

Le Vie, Donald S. Jr. "Data Mining, eCommerce, Fraud Detection, and Technical Communication." PowerPoint Presentation. Orlando, FL. 21-24 May 2000.

Magkos, Emmanouil, et al. "Accurat and Large-Scale Privacy-Preserving Data Mining Using the Election Paradigm." *Data & Knowledge Engineering* 68 (2009): 1224-36. Web. 14 Sept. 2010.

Maleshefski, Tiffany. "Three roads to an OpenDocument-friendly Office." *INSIGHT*. 23 Jul. 2007. Web. 14 Feb. 2011.

McCroy, Amy and Beth M. Russell. "Crosswalking EAD: Collaboration in Archival Description." *Information Technology and Libraries* (2005): 99-106. Web. 14 Feb. 2011.

Morrison, Michael. Sams Teach Yourself XML in 24 Hours. Indianapolis: Sams, 2005. Print.

Nayak, Richi and Wina Iryadi. "XML Schema Clustering with Semantic and Hierarchical Similarity Measures." *Knowledge-Based Systems* 20 (2007): 336-49. Web. 14 Sept. 2010.

Pennington, Lori L. "Approaches/Practices: Surviving the Design and Implementation of a Content-Management System: Do the Benefits Offset the Challenges?" *Journal of Business and Technical Communication* 21 (2007): 62-73. Web. 14 Sept. 2010.

Pluempitiwiriyawej, Charnyote and Joachim Hammer. "Element Matching Across Data-Oriented XML Sources Using a Multi-Strategy Clustering Model." *Data & Knowledge Engineering* 48 (2004): 297-333. Web. 14 Sept. 2010.

Renear, Allen H. et al. "Strategic Reading, Ontologies, and the Future of Scientific Publishing." *Science* 325 (2009): 828-32. Web. 7 July 2011.

Rivera-Aguilera, Alma Beatriz. "XML Markup and Information Retrieval in Magazine Articles: Exploratory results and Implementation Issues." *The International Information & Library Review* 37 (2005): 337-43. Web. 14 Sept. 2010.

Robidoux, Charlotte. "Rhetorically Structured Content: Developing a Collaborative Single-Sourcing Curriculum." *Technical Communication Quarterly* 17 (2008): 110-35. Web. 14 Feb. 2011.

Rockley, Ann. "Single Sourcing: It's About People, Not Just Technology." *Technical Communication* 50 (2003): 350-4. Web. 14 Sept. 2010.

---. "The Impact of Single Sourcing and Technology." *Technical Communication* 48 (2001): 189. Web. 14 Sept. 2010.

Sapienza, Filipp. "A Rhetorical Approach to Single-Sourcing Via Intertextuality." *Technical Communication Quarterly* 16 (2007): 83-101. Web. 14 Feb. 2011.

---. "Usability, Structured Content, and Single Sourcing with XML." *Technical Communication* 51 (2004): 399-408. Web. 14 Sept. 2010.

Shah, Divyesh and Sheng Zhong. "Two Methods for Privacy Preserving Data Mining with Malicious Participants." *Information Sciences* 177 (2007): 5468-83. Web. 14 Sept. 2010.

Stolley, Karl. "Using Microformats: Gateway to the Semantic Web Tutorial." *IEEE Transactions on Professional Communication* 52 (2009): 291-302. Web. 10 Apr. 2010.

Stumme, Gerd et al. "Semantic Web Mining State of the Art and Future Directions." *Journal of Web Semantics*. 4 (2006): 124-43. Web. 13 June 2011.

Sure, York et al. "On-To-Knowledge: Semantic Web Enabled Knowledge Management." *Web Intelligence*. Ed. Ning Zhong, Jiming Liu, and Yiyu Yao. New York: Springer-Verlag Berlin Heidelberg, 2003. Print.

TEI By Example. *Module o: Introduction*. tbe.kantle.be. 16 Oct. 2011. < http://tbe.kantl.be/TBE/modules/TBED00v00.htm>.

Text Encoding Initiative. *TEI: Text Encoding Initiative*. tei-c.org. 16 Oct. 2011. < http://www.tei-c.org/index.xml>.

The OpenDocument Format: Chapter 17. Mashing up Desktop and Web-Based Office Suites. mashupguide.com. 24 May 2011. < http://mashupguide.net/1.0/html/ch17s03.xhtml>.

Thibodeau, P. "Feds See Value in XML but Face Deployment Problems." Computerworld, 2002. Web. 14 Feb. 2011.

Trotman, Mike. "XML data model vs. relational model." *Going Native: Making the Case for XML Databases*. XML.com, 7 Apr. 2005. Web. 29 October 2011.

Tseng, Frank S.C. and Wen-Jong Hwung. "An Automatic Load/Extract Scheme for XML Documents Through Object-Relational Repositories." *The Journal of Systems and Software* 64 (2002): 207-18. Web. 14 Sept. 2010.

Turns, Jennifer, et al. "Moving Toward Knowledge-Building Communities in Informational Web Site Design." *Technical Communication* 52 (2005): 52-63. Web. 14 Sept. 2010.

Vlist, Eric van der. *Relax NG*. Sebastopol: O'Reilly & Associates, Inc., 2004. Print.

Wang, Sheng and Raymond A. Noe. "Knowledge Sharing: A Review and Directions for Future Research." *Human Resources Management Review* 20 (2010): 115-31. Web. 14 Sept. 2010.

Wick, Corey. "Knowledge Management and Leadership Opportunities for Technical Communicators." *Technical Communication* 47 (2000): 515. Web. 14 Sept. 2010.

Wheeles, Dana. "Digital Forum: Redesigning NINES." *Journal of Victorian Culture* 15 (2010): 145-9. Web. 15 July 2011.

Williams, Joe. "The Implications of Single Sourcing for Technical Communicators." *Technical Communication* 50 (2003): 321-7. Web. 14 Sept. 2010.

Yang, Liang Huai, et al. "Efficient Mining of Frequent XML Query Patterns with Repeating-Siblings." *Information and Software Technology* 50 (2008): 357-89. Web. 14 Sept. 2010.

Yeates, Robin. “An XML infrastructure for archives, libraries and museums: resource discovery in the COVAX project.” *Program* 36 (2002): 72-88. Web. 14 Feb. 2011.

Zhao, Qiankun, et al. “XML Structural Delta Mining: Issues and Challenges.” *Data & Knowledge Engineering* 59 (2006): 627-51. Web. 14 Sept. 2010.

Zhu, Dan, et al. “Identity Disclosure Protection: A Data Reconstruction Approach for Privacy-Preserving Data Mining.” *Decision Support Systems* 48 (2008): 133-40. Web. 14 Sept. 2010.