Digital Asset Management Using A Native XML Database Implementation

Shalaka Natu
Purdue University
401 N. Grant St.
West Lafayette, IN 47907
765-494-4545
snatu@tech.purdue.edu

John Mendonca
Purdue University
401 N. Grant St.
West Lafayette, IN 47907
765-496-6015
jamendonca@tech.purdue.edu

ABSTRACT
Digital Asset Management (DAM), the management of digital content so that it can be cataloged, searched and re-purposed, is extremely challenging for organizations that rely on image handling and expect to gain business value from these assets. Metadata plays a crucial role in their management, and XML, with its inherent support for structural representation, is an ideal technology for this. This paper analyzes the capabilities of a native XML database solution via the development of a "proof of concept" and describes implementation requirements, strategy, and advantages and disadvantages of this solution.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database Applications – *image databases;* and D.3.2 [Programming Languages]: Language Classifications – *extensible languages, XML*.

General Terms
Languages, Performance

Keywords
Digital Asset Management, DAM, digital images, XML database

1. INTRODUCTION
Digital asset creation and management evolved in the late 1990s. Companies have created massive digital assets in the form of images, video and audio files, streaming media, Power Point templates, web pages, and PDF files containing engineering specs, legal documents, internal memos and more. The World Wide Web has drastically increased the need for digital information and its exchange. Ille [8] identifies a digital asset as a strategic asset like any other form of working capital, and states that its efficient management is being increasingly recognized as a competitive lever for companies all over the world.

Development of the model for storing any form of digital object into a structured format requires a deft combination of asset analysis, strategic thinking, business planning and appropriate technology. Companies can achieve early strategic advantage by implementing management systems for digital assets that can be repurposed or customized, providing process efficiencies in collaborative work. Digital Asset Management (DAM) can deliver competitive advantage for advertising agencies, technical or engineering documentation departments, designers, producers and others by reducing time spent locating creative assets.

Enterprises often require reusing or sharing their digital assets. It is indispensable to store content in an organized way to reuse or process it for future needs. Global enterprises are facing the daunting challenge of figuring out how best to address the growing complexity of creating digital assets and managing the flow of assets through a single infrastructure [11]. Exacerbating the challenge is the fact that companies are creating a massive volume of digital assets but are rarely examining their organized storage and retrieval methods.

2. SIGNIFICANCE OF THE PROBLEM
DAM systems are still relatively new, but organizations have started realizing the importance and need for digital asset management. The Gartner Group affirms that only a limited number of technically advanced commercial content providers use DAM systems today to digitally construct, store and distribute rich media content in single medium forms [7]. The systems also have limited corporate use in advertising firms and marketing departments. Gartner predicts that by 2005 more than 25% of all the enterprises with commercial internet operations will use DAM systems. By 2010, more than 45% of all the enterprises with commercial internet operations will use DAM systems.

Recently reported cases provide evidence that companies have started investing in technology for DAM. For example, the Coca Cola company has bought technology from IBM for its digital advertisement archives, which contain 9,000 graphical images, 7,000 scanned documents and more than 25,000 corporate videos and television advertisements [13]. The technology includes search tools for retrieving, updating, managing and disseminating historical records online, including the company's famous marketing and advertising icons.

Another case is that of the Shoah Foundation. Steven Spielberg established the Shoah Foundation with the mission to videotape and preserve the testimonies of the Holocaust survivors and witnesses. The foundation has collected more than 50,000 eyewitness testimonies in 57 countries and 32 languages [12]. The challenge now is to manage, catalog and disseminate the video and digital collection of testimonies of those survivors. These digital assets of the Shoah Foundation are cataloged with lists of keywords, text summaries describing the survivors, related documentaries focusing on topics such as the ghettos or labor camps they lived in, and past and present photos of them and their families.

The following summarizes the major challenges that are faced during any wide adoption of DAM.

- Storage: One of the fundamental problems is physical deterioration of storage devices that store digital information. Magnetic media are particularly subject to data loss and errors. There is also the important question of hardware and software obsolescence and the future availability of today's technologies.
- Procedural Issues: Technical problems are further complicated when resources to be preserved need to be transformed into digital form. Digitization of paper analogs for access and preservation is a time consuming and labor intensive task. Beyond this technical problem, there is a host of financial, legal, and policy problems.
- Security: Securing digital assets against misuse, theft, or damage is an ongoing concern of organizations.
- Copyright: One of the major legal issues associated with electronic assets is the actual scanning or digitizing of materials. Copyright holders have the exclusive right to reproduce, distribute, and display their work. Ferullo [3] points out that digitizing and posting infringes on these exclusive rights.
- Distribution: Digital assets will be utilized to the fullest only when they can be distributed via proper communication channels to the intended users.
- Infrastructure: DAM requires a robust IT infrastructure to support creation and distribution.
- Human Factors: While the purpose of DAM is to provide greater efficiency, getting users to adapt to the new environment can be a challenge. This is an important issue because many DAM solutions require a change in work processes before the users see any benefits [6].

The requirement to manage and deliver digital assets is becoming critical to almost all content-related applications, due to the evolution of the Internet and growth of digital assets. Enterprises need plans to benefit from the new world of rich, valuable digital assets that will affect everything from their internal processes and customer relations to their web site and telecommunications infrastructures.

3. AN XML SOLUTION

3.1 Components of DAM

Widespread use of rich media has spurred the growth of the DAM market. Frost and Sullivan, a market analysis firm, claims the average user in a media enterprise looks for a media file 83 times a week and fails to find it 35% of the time [5]. Canto Software research predicts that DAM solutions will drop that figure to 5% [9]. With the growth of internet publication, document metadata is increasingly important because it provides contextual information about digital assets necessary for customization of the content. In response, Adobe Systems plans to unveil new metadata technology designed to ease the process of applying content to multiple types of media. The XMP (Extensible Metadata Platform) provides a framework for standardizing the creation and processing of document metadata across publishing workflows, according to Adobe officials [10].

A review by Doering [2] provides the insight into methods for digital asset management. According to Doering, a DAM solution must include the following critical features:

- Indexing: As content is generated and stored, it is indexed according to various possible criteria. Metadata does not merely describe by whom, when, and how a piece of content was created; it can also be used to interpret and catalog the content itself.
- Rights Management: Includes handling rights to the content or restricting the use of the content by the purchaser/end-user. This might occur, for example, with corporate information or licensed images from a third party incorporated into the content.
- Reuse: With a viable DAM system in place, the internal content developer can research and select appropriate content for reuse in new content. This represents a significant savings potential for the companies.
- Review: A final benefit of an online catalog with a DAM system is the ability to review older content more easily.

3.2 The XML/Metadata Approach

By incorporating a DAM system, a company gains both the savings from reusing content as well as revenue from continued sales of the same elements. According to Fontana [4], XML databases serve in a complementary role to traditional databases especially as XML becomes prevalent. Nearly 85% of large corporations are expected to move all their web-based content to XML over the next three years.

Fundamentally, two high-level approaches may be adopted for implementing XML databases.

1) **XML-enabled database**: In an XML-enabled database, the documents are stored in constituent fragments. Here the XML data is stored in object relational form and one can use an XML SQL Utility (XSU) or SQL functions and packages to generate the whole document from its object relational instances.

2) **Native XML database**: In a native XML database approach, XML is not fragmented but rather is stored as a whole in the native XML database. This means that documents are stored, indexed, and retrieved in their original format, with all their content, tags, attributes, entity references, and ordering preserved. In this technique, the XML Database is designed to handle XML in its native format, thereby speeding access to data by eliminating the need to transform it into the rows and columns of a standard relational database.

Biggs [1] suggests there are three principal reasons to implement a native XML database: 1) enterprises today use a mix of data, such as the data housed in object-oriented databases and unstructured data that needs to be exchanged with partners (native XML databases can leverage all of these disparate sources); 2) XML databases can boost processing performance due to XML-based transactions; and 3) digital assets described in the XML format can be used by other applications.

3.3 A Technique Using Native XML

When developing a native XML solution, certain steps should be followed:

1) Identify the need for DAM

2) Know your assets: Identify various assets, define and understand their use in the organization

3) Define search needs and key attributes of your assets

4) Capture the objects (digitized) and data about the objects (metadata)

5) Process: Generate and store XML files associated with each object

The basis of this technique lies in creation and usage of semi-structured metadata stored in an XML format to manage digital assets efficiently. When the structure of the data is irregular and implicitly given by the document itself, not by a global scheme, they are often referred to as semi-structured data. The ability of an XML to handle dynamic data provides leeway for the applications to use semi-structured data to describe their digital assets.

XML is becoming the de facto data exchange format for the Web. The XML enclosing metadata can be either stored in the database for future retrieval or be easily transferred over the Internet to any other systems.

The Oracle 9i DBMS (used for the Proof of Concept) provides an "XMLType" data type that can be used to store and query XML data in the database. The XMLType has member functions one can use to access, extract, and query the XML data using XPath expressions, a standard developed by the World Wide Web Consortium to traverse XML documents.

3.4 Proof of Concept

This study analyzed existing approaches for non-text based digital asset management and implemented a DAM solution by applying a native XML Database technique. Meta-data of digital assets is often semi-structured and the digital files are of varied types. XML databases are most appropriate for managing diverse types of semi-structured digital assets.

For this project, the Proof of Concept (POC) was developed using facilities and resources of the University's Digital Enterprise Center (DEC) at the School of Technology. The Product Lifecycle Management (PLM) lab at DEC creates, simulates and shares digital information related to a company's products, processes and resources. These digital assets encompass graphical images, presentations, and video/audio clips of manufacturing processes representing manufacturing models. A digital asset produced in the PLM lab is the intellectual property of the DEC and needs to be managed for future use or research. Though the demonstration focused on the manufacturing process models in the PLM lab, the XML Database technique can be applied to any form of digital asset.

Since the potential file population is very vast and is of varied types, the POC restricted the sampling to the following categories of digital assets: Audio files, Video files, Images, and Text based files (presentation slides, Word documents, Acrobat files). The POC confined the sample to a limited number of files describing assembly line parts available from the PLM lab.

Metadata was stored for each of the digital objects of the assembly line. The global parameters and sub-parameters used to describe the digital files included the following: File Name, Type (Image, Audio, Video, Word etc), Author, Creation Date, General Description, Keywords, and a Comment. A keyword search capability was added for searching within these and other parameters.

The Proof of Concept application was developed to provide a storage, search and retrieval facility to manage the digital assets of the PLM lab using the School of Technology's software and hardware resources. The application provides a web-based interface for convenient management of the digital models and has an "n-tiered" architecture. The backend database is Oracle 9i with native XML support. The web interface was developed using JSP and the middle tier was incorporated using Java Servlets. Analysis was conducted with the following steps:

- Various types of data (heterogeneous data) were collected for demonstration purposes.
- The validity of metadata was checked before entering it into the system.
- Upon validation, data was entered in the system (Oracle 9i database) through the web-interface.
- With the appropriate search parameters, the data entered could be searched for and retrieved. Depending on the requirement, this retrieved data could be viewed, reused or repurposed.
- Providing different search criteria tested consistency and reliability in the retrieval of the asset.

The following technologies were used to develop the POC: Apache Webserver 1.3; Apache JServ 1.1; JDK 1.1; Oracle 9i (9.0.0.2); XML; Java, JSP, Servlets. Tools used for development purposes included: ERwin database design software; Borland Jbuilder; Microsoft Front Page; and Oracle Enterprise Manager.

The POC was developed with a three-tier architecture as shown in Figure 1. The first tier of the architecture presents an interface to the user to facilitate access to digital files. The interface is web-enabled and was developed using Java Server Pages. This tier provides a user-friendly navigation through the site for managing digital files, including screens for inserting, deleting and searching on file data.

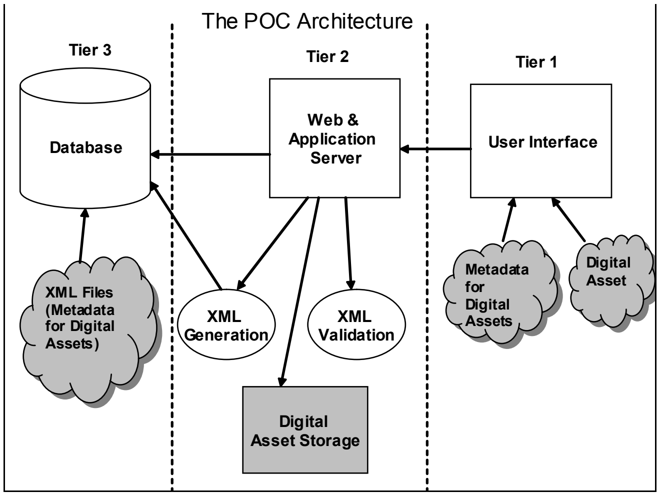

Figure 1: 3-Tier Architecture for POC

The middle tier of the architecture consists of Servlets and Java helper classes. Servlets direct the flow of the control and provide database connectivity for the application. In addition, the business logic is implemented in Servlets. Helper classes are used by Servlets to create and manage user-defined Java objects. Servlets and JSPs are deployed on an Apache Web server and Apache Jserv.

The third tier of the architecture is the database layer itself. As noted previously, the POC uses the XML capabilities of the Oracle 9i database management system.

4. OBSERVATIONS AND EVALUATION

4.1 The XML Database

The XML database technique stores attributes of digital assets in the database in the form of an XML file. While the XML resides in the database system, digital assets might be files in a file system or a BLOB in a relational database. This demonstration of the technique stores digital assets in a file system.

XML databases provide some features that are analogous to conventional text databases. Common to all native XML databases are indexes, which allow the query engine to jump to a particular point in an XML document. This gives such databases a tremendous speed advantage when retrieving entire documents or document fragments. This is because the database can perform a single index lookup and retrieve the entire document or fragment in a single read.

Normalizing data for an XML database is largely the same as normalizing it for a relational database. The programmer needs to design the documents so that no data is repeated. One difference between native XML databases and relational databases is that XML supports multi-valued properties while (most) relational databases do not. This makes it possible to "normalize" data in a native XML database in a way that is not possible in a relational database.

4.2 Native XML vs. XML-Enabled

A native XML database defines a (logical) model for an XML document -- as opposed to the data in that document -- and stores and retrieves documents according to that model. A native XML database has an XML document as its fundamental unit of (logical) storage, just as a relational database has a row in a table as its fundamental unit of (logical) storage. It is not required to have any particular underlying physical storage model. For example, it can be built on a relational, hierarchical, or object-oriented database, or use a proprietary storage format such as indexed, compressed files.

An XML-enabled database has an added XML mapping layer provided either by the database vendor or a third party. This mapping layer manages the storage and retrieval of XML data. Data that is mapped into the database is mapped into application specific formats and the original XML meta-data and structure may be lost. Data retrieved as XML is NOT guaranteed to have originated in XML form. Data manipulation may occur via either XML specific technologies (e.g. XPath) or other database technologies (e.g. SQL). The fundamental unit of storage in an XML-enabled database is implementation dependent. The XML solutions from Oracle and Microsoft, as well as many third party tools, fall into this category.

4.2.1 Advantages of Native XML

Native XML databases have several advantages over relational databases. Since native XML databases do not have database schemas, one can store similar documents with more than one schema in the database at the same time. While one may still need to redesign queries and convert the existing documents -- a non-trivial process -- this may ease the transition process.

XML databases make search within the XML documents very efficient. If the data is parsed into Document Object Model (DOM) format, XPATH can be used. XML database solutions usually add a text indexing system so that query performance is improved.

4.2.2 Disadvantages of Native XML

Currently, only a few native XML databases enforce referential integrity. The reason for this is that most native XML databases do not currently support linking mechanisms, so there are no references for integrity checking. Therefore, applications that rely on referential integrity mechanisms of databases must enforce these constraints themselves for XML databases. In the future, many native XML databases will probably support linking mechanisms and referential integrity.

Another disadvantage of XML databases is that while query performance is better, update performance suffers. The reason is that the index entries for a document must be deleted and new entries created whenever a document is inserted or updated.

In general, an XML Database approach is better because it supports the full power of XML. However, a major drawback is performance degradation, as data must be constantly reparsed into a DOM tree, wasting cycles and memory. Additionally, update capabilities are weak, and finally, automated enforcement of integrity constraints needlessly places unreasonable burden upon application programmers, increasing risks and costs.

5. CONCLUSION

Digital Asset Management (DAM) is an evolving field with a great potential. With the evolution of computers and the Internet, companies have been creating an enormous volume of digital content in various formats. Managing this content, so that it can be cataloged, searched and re-purposed, is extremely challenging for organizations.

XML is a commonly used standard in internet applications. Hence, representing the metadata of a digital content in an XML format is, on the surface, a good design decision. XML, by its very nature, provides for a complex, well-defined and yet extensible structural representation of the metadata and interoperability between various applications dealing with digital assets. A major advantage of having XML natively in the database is that one can perform the relational manipulation operations such as insert, update, and delete on the whole or partial XML and can also perform XML specific operations like XPATH search and node modification using the power of SQL. This, and other advantages give native XML databases an edge over systems that don't use XML or that manage XML externally.

6. REFERENCES

[1] Biggs, M. (2001, December 3). Two ways to bring XML to your databases. InfoWorld, Framingham, 23 (49), 20.

[2] Doering, D. (2001, August). Defining the DAM thing: how digital asset management works. Emedia Magazine, Wilton, 14(8), 28-32. Retrieved February 27, 2002, from http://proquest.umi.com/pqdweb?Did=000000077222944&Fmt=4&Deli=1&Mtd=1&Idx=5&Sid=12&RQT=309

[3] Ferullo, D. L. (2002). The challenge of e-reserves. Net Connect, 48 (8), 33-35.

[4] Fontana, J. (2001, November 5). Upgraded database makes the most of XML. Network World, Framingham, 18 (45), 29. Retrieved February 27, 2002, from http://proquest.umi.com/pqdweb?Did=000000088274369&Fmt=4&Deli=1&Mtd=1&Idx=7&Sid=3&RQT=309

[5] Frost & Sullivan.s (n.d.). U.S. Digital Asset Management Markets. Retrieved March 23, 2002, from http://www.frost.com/prod/servlet/fcom?ActionName=DisplayReport&id=A192-01-00-00-00&ed=1&fcmseq=1043213084248

[6] Garcia, K. (2001, November). Broadcasters starting to ease into digital workflow. TVB Europe, 10(11), 22-24.

[7] Gilbert, M., Landers, G., Latham, L. & Lundy, J. (2001, November 26). What's cool, what's hot: content technology hype cycle. Gartner Advisory. Retrieved February 9, 2002, from http://gartner.adpc.purdue.edu/rasdquest/RESEARCH/RAS/102700/102760/102760.html

[8] Ille, C. (2001, February 26). Market definitions: digital document imaging. Gartner Group. Retrieved February 27, 2002, from http://proquest.umi.com/pqdweb?Did=000000093522513&Fmt=3&Deli=1&Mtd=1&Idx=4&Sid=3&RQT=309

[9] Martin, N. (2001, September). DAM right! Artesia technologies focuses on digital asset management. EContent, Wilton, 24(7), 60-62.

[10] Moore, C. (2001, September 24). Adobe touts images. InfoWorld, Framingham, 23(39), 36. Retrieved February 27, 2002, from http://proquest.umi.com/pqdweb?Did=000000081956628&Fmt=3&Deli=1&Mtd=1&Idx=2&Sid=12&RQT=309

[11] Moore, C. (2001, September 24). Content management plays role in cross-media publishing. Computer World. Retrieved February 9, 2002, from http://www.computerworld.com/storyba/0,4125,NAV47_STO64339,00.html

[12] Solomon, M. (2002, January 14). Managing the memories. Computer World. Retrieved April 16, 2002, from http://www.computerworld.com/storyba/0,4125,NAV47_STO67304,00.html

[13] Weiss, T. (2001, December 10). Coca-Cola ad legacy gets help from IBM. Computer World. Retrieved February 9, 2002, from http://www.computerworld.com/storyba/0,4125,NAV47_STO66495,00.html