LEGAL XML

Unofficial Note

XML and Electronic Filing Issues for Courts

Document Number
UN_10008_2000_01_04
Current Version
1.0
(http://www.legalxml.org/DocumentRepository/UnofficialNote/Clear/UN_10008_2000_01_04.htm)
Previous Version(s)
None.
Workgroup Information
Workgroup Name: None
Workgroup Chair(s): None
Workgroup Mailing List: None
Workgroup Mailing List Archive: None
Workgroup Website: None

Document Author(s)
Roger Winters, Department of Judicial Administration, King County, Washington (Roger.Winters@metrokc.gov)
Document Editor(s)
None (None)

Short Statement of Status
Active

Abstract

This Unofficial Note overviews electronic court filing issues and various document format solutions. It discusses images, word processing documents and XML documents. It concludes there is a need for standards and provides a list of recommendations.

Status of Document

This is a Legal XML Work Group Unofficial Note for review and discussion by interested Members.

Table of Contents

9. Notes

Introduction

XML, the e**X**tensible **M**arkup **L**anguage, figures importantly in many plans for the *electronic court file* of the future and, ultimately, for all electronic legal documents.

This paper discusses how courts have come to work toward electronic documents and case files. It explains what XML is and why it has captured the attention of so many court managers, administrators, judges, clerks, academics, developers, and technicians from Singapore to Seattle, Georgia to New Mexico, Maryland to Manhattan, Utah to Virginia, and beyond. The paper concludes with comments on work toward a standard for XML and the electronic court file.

Court Filings: What's the Problem?

It has been estimated that up to *half* the cost of operating a court may be attributed to the costs of manipulating paper!

There are many courts: federal appellate, federal, state appellate, state, local, administrative, and more. Court business is based on cases. Cases have identifying numbers, titles, named parties, and a record of actions taken. The court file is the official case record, usually found in one or more file folders, sometimes supplemented with information in a computer system. There are many case types: criminal, divorce, traffic, bankruptcy, domestic violence, probate, juvenile, child custody, civil lawsuits, appeals, and more.

Court records are usually kept for long periods, often "permanently." The officer that keeps the record is usually called "Clerk." Often the Clerk is independent of or organized separately from the Court.

Courts live on documents. Most court files are composed of *paper* documents and file folders. Data(1) is presented as words on paper. It is a familiar, friendly medium. We are used to handling paper. We like the look and feel of it. We like the physical cues that help us find our place in a file or document. We like the portability of paper.

The papers that "document," or prove, that information was filed, action taken, rulings made, data recorded, *etc*., are usually signed in pen-and-ink, sometimes further authenticated by notarizing. Documents that are "records" are "officially filed" to become standing proof of what transpired regarding a case. All courts have *de facto* "document management systems" (however well or ill developed) for *paper* documents. That is, papers are sent from place to place according to rules, procedures, and traditions.

In King County Superior Court, we have around one-half million paper file folders on shelves at four different sites. The case file has all the original-signature documents formally "filed" with and accepted by the Clerk. We receive over 7,000 documents on an average workday. Every single document gets read by a deputized clerk whose job is to re-key information from it into a mainframe computer system. Many filed documents require other clerk's actions before they can be placed in their respective file folders. Stacks of processed documents have to be put in order before documents can be distributed to their respective folders. Getting documents to their file folders takes time (our rule: "from filing to file = five Court days").

Only one person at a time can use the official paper case file. Misfiled documents are perceived to be "lost." File folders re-shelved to the wrong location are in effect "lost" until re-discovered. Some people try, and a few succeed, to steal, deface, or tamper with documents or files. Some file systems are efficient; others are outdated and service is slow. No one likes to wait for needed information. The Internet has taught us to expect information to be there NOW (if not sooner). Nevertheless, a file may not be available even when a decision is urgently needed. Sometimes file rooms burn, are inundated by floods, or crumble in earthquakes. Disaster recovery plans for records are often inadequate or non-existent.

What are Electronic Filings?

Electronic filings are documents created in or converted into electronic form before, during, or after filing. Electronic filing systems replace the paper "file room" with a computer based system for electronic storage and retrieval.

With electronic files, more than one person can see a document at a given time. The "original" or "official copy" of the document is never touched because it is kept in the electronic "file room" in a storage device (usually in a "jukebox" — a huge CD player filled with disks or platters — or an array of magnetic hard disks). An electronic document has to be indexed, usually twice: 1) to track the document to its location in the electronic file system and 2) to identify the document with its case number or other basic identifying information. Electronic documents are copied to us over our connections with the computer file room systems (*e.g.*, the Internet, an office-wide computer network)[2]. The costly steps of paper processing go away: case files don't have to be retrieved by a file clerk and they never have to get re-shelved after use. They rarely if ever get lost.

The possibilities of electronic filing have inspired compelling visions for many court administrators, judges, clerks, attorneys, and litigants. Some courts and related organizations have gone far in planning, funding, and constructing systems or pieces of systems. Others don't have (may never have) the time to do it themselves; they need help and practical applications they can use.

Images

Imaging is a well developed way to incorporate hard copy *source documents* into an electronic filing system. Imaging is a lot like faxing. When a document is passed through an electronic *scanner*, each page is converted to computer code that represents the patterns of light and dark that make up the page (the result is called a "bit map"). The resulting computer code is compressed and stored electronically (in a fax, the code is transmitted across telephone lines to the recipient's fax machine).

An imaged or faxed electronic document acts more like paper than like a word-processed electronic document.[3] When viewed by humans, each shows the same *words on paper* the writer produced.[4] An image viewer un-compresses and interprets its code to reconstruct a viewable picture of what was scanned, displaying it on a computer screen or printing it on paper.[5] Even though it can be shown on a computer screen, an image cannot be spell-checked, text searched, or edited like a word-processed document, because it is merely a picture of the *shapes* which made up the *source document* from which it was made. An image has to be "read" by a human to find and use the data within it.[6]

Imaging is already in use by many courts. It is a well-established technique with a mature service industry ready to provide quality systems and support. For an existing document management system, imaging speeds up many processes and saves time eaten up by paper manipulation, physical storage, and unavailability/loss of documents. Often, imaging is purchased with a "workflow" product that allows redirection and redesign of how information moves and how work is sequenced. Imaging will not in itself *fix* a bad workflow or document management system, but it provides a wonderful opportunity to do so.

management system, but it provides a wonderful opportunity to do so.

Paper will almost certainly continue to make up some part of what is presented for inclusion in the court file. Document imaging needs to be a part of a full-service electronic court filing system.

What is Document Mark-Up?

"Mark-up" means to use special coding within a document to specify any of three characteristics for the document: 1) **formatting,** how the document is meant to be displayed, *e.g.*, how "special effects" (*i.e.*, underscoring, bold-face, italics, or a combination) are to be used, 2) a document's **logical structure,** how the document is organized (*e.g.*, that it has a title, chapters, sub-headings, an outline structure, tables, illustrations, footnotes, appendices), and 3) **data**, pointed out within the document by using conventional "tags" to name and locate it. Different types of documents have different combinations of these characteristics.

What has excited courts and clerks is the potential that data mark-up ("tagging") has for eliminating repetitive data entry and laborious paper handling in favor of automated, more efficient systems. The format and structure characteristics are also important in legal documents. Since the mid-1990s, many court leaders and developers have hoped "mark-up language" would be a way courts and litigants could point to the data in a court filing so it can be located and processed automatically from within a word-processed electronic document.(7)

Mark-up does not provide formatting, recognize structure, or interpret data in and of itself. Even in the World Wide Web, where pages are built with HyperText Markup Language (HTML), which has specific tags for formatting, it is the **browser** software (*e.g.*, Microsoft's Internet Explorer, Netscape's Navigator) that interprets the tags and applies the formatting. Sometimes the formatting can be shown quite differently by different browsers. Once tagged, marked-up data can be grabbed or "parsed" by a "parser" software application. Then, the extracted data is ready to be used elsewhere.

Where did mark-up come from? In the 1980s, work began to find a standard way of describing documents in terms of their structure, appearance, content, and use. Documents (and the data within) needed to be analyzed and set up so they could be viewed in different systems and so their data could be variously manipulated. To accomplish this, the "Standard Generalized Markup Language, SGML," was developed; it was adopted as a standard by the International Standards Organization (ISO) in 1986. Complex and difficult for a lay person, SGML's coding and structure provide a rational and consistent method to mark up documents.

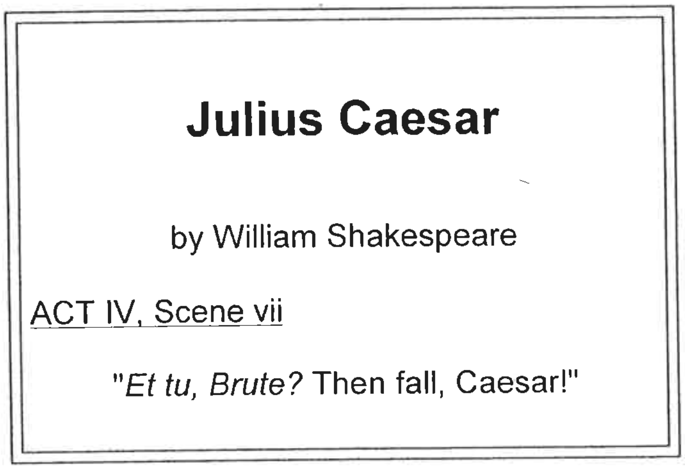
An electronic document's characteristics are described using mark up techniques specified in a "Document Type Definition (DTD)" which has been built following SGML procedures and syntax (*i.e.*, rules of usage). Most mark up languages use angle brackets, slashes, and other conventional symbols consistent with the SGML standard.(8) Mark-up codes are often paired, with a starting point indicated by a Name (of a datum, format instruction, or structural element) placed within two angle brackets as shown here: <Name> and a paired closing code also showing the Name within angle brackets, this time with a slash, as shown here: </Name>. This is a "toggle," a switch that turns a given Name "ON" then "OFF." (WordPerfect users who have ever used the "Reveal Codes" feature have seen such paired toggles, *e.g.*, "<B>This text is boldfaced.</B> This is not.")

An illustration of mark-up may help.(9) Consider the following hypothetical code:

```
<?sgml version=1.0?>
<?DTD=Shakespeare, version=1.0?>

<Title>Julius Caesar</Title>
```

```
<AuthorLine>by

  <Role=Author>William Shakespeare</Role>

</AuthorLine>

<Heading, Level 2>Act IV, Scene vii</Heading, Level 2>

<Paragraph>

  <Italics>"Et tu, Brute?</Italics> Then fall,
<Role=Emperor>Caesar</Role>!"

</Paragraph>
```

Here's what the "marked-up" document should look like when displayed:(10)



Julius Caesar

by William Shakespeare

ACT IV, Scene vii

"*Et tu, Brute?* Then fall, Caesar!"

Any SGML-compatible computer application could interpret (using the above codes as they would be found defined in a [hypothetical] "Shakespeare, ver. 1.0" Document Type Definition) how the document is meant to be organized **(logical structure)**, how some text is displayed differently **(format)**, and what items **(data)** are named for possible parsing. SGML provides standard ways to define DTDs that can be used for a vast number of types of documents. A compatible parser or browser application will handle the above in the expected, "standard" way (*i.e.*, just like a similar application would, following the codes and rules of the "Shakespeare DTD").

For this document, the version of the SGML standard is cited as "1.0," found in a standard location on the World Wide Web. The first line also references a hypothetical DTD for Shakespearean plays. The DTD is also where one would look to find the data elements being tagged. For example, consider the data element called "Role," marked by <Role>[Role Attribute]>Character's Name</Role>.(11) For the data element "Role" there could be many different values that would be *attributes* of that element. The attribute that is associated with a data item tagged with a Name would be placed to the right of the > sign after <Role>. This would facilitate, for example, displaying a List of Roles found in this particular play.

The DTD (along with a "Stylesheet" for page formatting) would stipulate special rules for handling this type of document, such as:

- *A paragraph may use italics for emphasis within a paragraph of normal text text (12-point type).*
- *Titles are centered in 18 point type*

- *Titles are centered, in 16 point type.*

- *Author Attribution lines are centered below the Title, in 12 point type.*

- *Level 2 Headings are underscored and formatted to the left, in 12-point type.*

- *Role attributes may be: "Emperor," "Centurion," "Wife," "Senator," "Slave,' "General," "Dagger Sharpener,"*

To review, SGML is a standard that tells how to define DTDs. DTDs define, among other things, "elements" of a document. "Tags" are only one part of an "element." DTDs define other things, like entities (*e.g.*, Roles) and their attributes (*e.g.*, Author, Emperor). Any program that displays the document should refer to and follow the rules of the DTD. The DTD was built by the rules of SGML. The document is built to conform to the DTD; the document does *not* follow the rules of SGML.

Developers and computer programmers use mark-up languages to ensure their software works properly when used to 1) mark-up documents, 2) display marked-up documents, and/or 3) extract data elements and attributes from marked-up documents for use elsewhere. In actuality, SGML-derived DTDs and their documents look and are much more technical, rich, and complex.

HTML

SGML is used to create DTDs. The most famous SGML DTD is HTML, "HyperText Markup Language," almost universally used to construct Web Pages. HTML's rules and features are supposed to be centrally controlled. Every extension of its codes and features is supposed to be submitted, reviewed, and approved by an international committee before it can be added to the HTML Standard. However, anyone can extend HTML in "unauthorized" ways. Indeed, Netscape and Microsoft have both done that. This is why documents are not presented the same way in the two browsers — this practice, in turn, causes huge headaches for programmers. HTML mark-up is used to manage the display and behavior of text, tables, lists, banners, graphics, and other features used in Web Pages. It is *not* used to name or manipulate data elements within the Web pages. It shows how to display titles, subheadings, footnotes, *etc.*, but it can't be used to tell you that "Julius Caesar was an Emperor" or that "William Shakespeare was an Author."

XML

In the State of Washington, Superior Court Clerks read documents filed with them in order to identify data, including *at least* the document's case number, serial number in the case, and title. Clerks re-type this information into the State's Office of the Administrator for the Courts' mainframe system, SCOMIS (Superior Court Online Management Information System). If data elements were tagged, software could copy data from a document and put the data into corresponding data fields in SCOMIS. No one would have to re-key that information.

The "eXtensible Markup Language, XML,"(12) seems to be "all the rage" today. It is young, it has excited many, and it has received lots of press in technical journals and on the World Wide Web. XML, like SGML, is a document architecture used to create DTDs. HTML is the most famous DTD created with SGML. XML is a subset of SGML; it is less complex and less featured than SGML. It has perhaps 80% of SGML's power but it is easier to use than SGML.

A single XML document can contain its own data and logical structure definitions, if they are declared within the document as it is set up. XML is much more useful when a large number of data names and tags are part of a shared, standard convention, at least among people and companies in the same field. Formatting in XML is also done with stylesheets. Today, there are two standards for stylesheets: CSS (Cascading StyleSheets) and XSL (eXtensible Stylesheet Language). CSS is a Microsoft product and XSL has been submitted for consideration as a W3C recommendation.

There are two aspects of XML it is helpful to distinguish: 1) XML can be the basis of a mark-up language to enable programs to move data and information from documents, enabling automated capture and processing. 2) XML is emerging as a new tool for designing Web pages.(13) XML on the Web is an area of hot competition for companies seeking a market advantage in web design and e-commerce.

XML DTDs as mark-up languages are *not software!* They are not something to be bought and sold. XML is a standardized set of tools that can be used by anyone. It has a level of technicality which most of us will never understand nor need to master. It also has a business level requiring business leaders to direct technicians on how to apply XML. Business needs determine what software has to be developed in order to serve those business ends.

There can be XML for any type of business or industry. There is XML for Chemistry, XML for Pharmaceuticals, XML for Web Page Building, and there will eventually be XML for Legal Documents and/or XML for Court Filings. Whether there should be a single standard for XML for Legal Documents and/or Court Filings is a question now before us all, as of this writing.(14)

There will be hundreds or thousands of data elements which various courts and clerks will want to be able to collect from documents through XML tagging to avoid repetitive keystroke labor, speed processing, and reduce data entry errors and costs. An XML markup language for Court Filings will specify how to structure documents and point to data in them. Software applications developed for each court will provide the tools and techniques for extracting data from marked-up documents and copying it over to whatever systems or uses a court or clerk has for the data. These applications are yet to be developed.

Formatting instructions, such as margins, footnotes, or where the Court Name, Case Caption, Case Number, *etc.*, are to go will be specified within an XSL (eXtensible Stylesheet Language), which tells the application displaying Legal XML documents how their elements are to be organized and shown. A Court system or individual court might have its own suggested or required stylesheets, but none could create universal, standard stylesheets since so many variations in format and style exist across all the courts.

A Legal XML mark-up language (DTD) will specify the data elements to be tagged (pointed to) within documents, so the data can be located by a parser and copied for use in a table, another document, another computer system, *etc.* Examples of such diverse data are: Case Number, amount of Court Costs, months of incarceration, names of participants, Courtroom Number, plaintiff's attorney's Bar ID Number, date/time of hearing.

A Legal XML DTD will show how data elements are related to each other. For example, under an element called <PersonIdentity> there could be *nested* sub-elements such as <FirstName>, <MiddleName>, <LastName>, <DateOfBirth>, <IdentificationNumber>, <Tattoos>, <StreetAddress>, *etc.* Documents called "Namespaces" will provide rules and guidance for collecting information on the various roles a person may have in relation to a case, *e.g.*, Judge, Prosecutor, Defendant, Clerk, Juror, Reporter, Witness.

The task before those who want to use XML will be to identify and assign clear definitions to terms and relationships, so they can be used consistently in the electronic documents filed in a court. The tasks for technical experts in XML will be to help the courts develop standard DTDs and build other tools they will need to set up documents and forms. Another technical task is to build the software to capture data from filed documents and carry the copied data to mainframe, case management, or other systems, where data is now transferred by hand or perhaps not at all. These tools are still to be developed.

The Search for a Standard

Technically, nothing prevents every court from creating its unique XML DTD for court filing. It will be sad if we move in that direction. Though there must always be room for local variations in procedures and customs, all actors in the justice system, from judicial officers and clerks to litigants, will benefit most from an XML that is (technically) *standardized.*

There are multiple levels of abstraction between the tools/applications that will be used by judges, lawyers, clerks, and litigants and the technical standards for Legal XML. Most of the technical infrastructure of XML will be invisible to users. The tools used to create legal documents will remain relatively familiar. Using standard Legal XML DTDs, especially where similar laws and legal institutions operate, will mitigate the need to learn completely different rules or to purchase different software in different courts or counties. XML provides a golden opportunity to make legal practice in more than one court jurisdiction much less confusing and costly for attorneys and litigants.

There are some business reasons not to standardize, such as when a software company wants its products to communicate only with its other products. Even so, such a company has to have a standard it uses internally. If there is to be efficiency, there must be a standard. The more who use a standard, the more value it has.

Technical standards allow many important things to happen:

1. Computer systems are more efficient in communicating with each other.

2. Computer applications can be copied and re-used.

3. Application developers can build on existing standards to create new and better applications.

4. Users can focus their time and investment on learning one system, not many different systems.

This is why several initiatives are under way today to develop a Standard Legal XML. The same vision of automation of data capture, transfer, and entry across systems inspires the people behind all the initiatives. All are inspired by the enormous savings in time, labor, and cost that can be achieved. Every XML visionary is excited by the prospect of significantly improving access to justice and supporting judicial decision making through electronic legal documents and this powerful technology.(15)

Those responsible for developing and operating document management systems in federal, state, and local courts are eager to learn how to use these tools to address their day-to-day problems. It is not clear today how a national standard for Legal Court Filing XML will emerge, whether it will be a product of coordinated effort among those who have started their own initiatives, or whether there will be a variety of approaches from which to choose.

The State of Washington has a committee composed of judges, administrators, clerks, bar members, and other officials concerned with court filings and efficient integration of legal information systems. The committee is considering recommendations on what such a Standard should provide. The recommendations are:

1. A coordinated, cooperatively developed national standard is desirable. It should incorporate the best work of all present XML initiatives.

2. If work on legal XML standards proceeds in parallel, each effort needs to contribute to one national standard. When courts become ready to implement XML-based filings, they should not have to choose among competing, incompatible approaches.

3. The standard needs to be open and non-proprietary, so anyone can without cost make use of it.

4. Those who develop technical standards must recognize the need for educational and instructional materials showing how the standard can be implemented for practical use.

5. The standard should clearly structure identities, roles, and relationships of those involved in a legal case.

6. Early development efforts should be kept as simple as possible. Early work should be focused on data in court filings and other legal documents.

7. The standard needs to support applications, stylesheets, and other work products that provide for local variation in terminology, procedures, institutions, and culture.

8. The standard needs to be ready soon. Some courts, hard-pressed to realize the efficiencies of electronic document management, will not be able to wait for years if complexity, politics, or other reasons delay development of the standard.

9. The standard must recognize that there will likely always be some hard copy submitted for filing in court files. Support is needed for both word-processed electronic documents and electronically viewable formats, such as images or non-tagged documents.

Many are nearly ready to implement this exciting technology to improve courts and better serve the public. They want to help ensure that a useful XML Standard is developed. The time and talent that has been devoted to this work needs to be the foundation of a strong movement to apply the promise of XML-related technology to improve the judicial process.

The author welcomes questions and comments. Please direct them to:

Roger Winters
Electronic Court Records (ECR) Manager
King County Department of Judicial Administration
Mail Stop KCC-JA-609
516 Third Avenue, E-609
Seattle, Washington 98104
Roger.Winters@metrokc.gov

Notes

1. Data is not *information*. Data is something that is recorded as knowledge. In itself, data has no value. *Information* is data that is indexed, sorted, stored, and retrievable. Information is valuable. It is most valuable when it is accurate and

when the process of moving information (into and out of storage) is quick and reliable. Thanks to Winchel "Todd" Vincent, III, Esq., of the Georgia State University (GSU) Electronic Court Filing Project for providing the author with this distinction and for providing technical comments throughout this paper.

2. When we think we are "downloading" a document to our desktop, we are actually receiving an exact copy of the stored "original," which remains untouched at its stored location.

3. Electronic documents that are created in a computer with word-processing use bits and bytes (standard codes) to represent words, numbers, lists, tables, and format instructions. Images are only electronically *viewable*.

4. Affordable grayscale scanners do not pick up many colors; indeed certain colors, like blues and reds, tend to disappear during scanning. Color imaging is available, but it is presently too costly and slow for large systems.

5. Please notice that paper is, in this sense, very much like a "screen" on which the electronic information is displayed.

6. There are tools that attempt to "convert" the shapes of words and figures on an image into their respective computer codes that would have been created if the document were word-processed. Such products as Optical Character Recognition (OCR) and Intelligent Character Recognition (ICR) have their uses. They cannot *perfectly* replicate all the information. To check every single page processed this way to ensure exact matching is much too labor-intensive and cost-prohibitive. The mission of Clerks is usually to preserve the records exactly as filed.

7. Images are but one type of "electronic document." Besides images, electronic documents include three other categories: 1) **page description languages** (Adobe's Portable Document Format or PDF, WordPerfect's Envoy), 2) **compound document languages** (Rich Text Format or RTF, Word, WordPerfect), and 3) **mark-up based languages** (HTML, Legal XML, Financial XML, *etc.*). Of these, RTF, Word, WordPerfect, mark-up based, and those PDF documents that are not originally images would count as "word-processed" electronic documents. All electronic document types, including images, are human readable when displayed to look like "words on paper." Documents which cannot be text-searched, *etc.*, are sometimes called "dumb documents," since they usually must be processed or used just like paper documents. Thanks to Winchel "Todd" Vincent, III, Esq., for clarifying these categories.

8. A DTD *could* use other symbols for the same purposes.

9. In this example, we take liberties with some technicalities to keep the information comprehensible for the non-technical reader. A different good example that illustrates mark-up concepts can be found in an article at: http://www.oasis-open.org/html/goldfarb.htm.

10. Display is done with an appropriate and particular "stylesheet;" using a different stylesheet, the document could look completely different. To control appearance, one needs to require a particular stylesheet be used to display it.

11. This is not the only way to structure tags for data elements. There are different styles. Some would use "nested tags" such as <AuthorLine><Role><Author> William Shakespeare</Author><Role><AuthorLine>). The DTD must have a set of consistent rules, procedures, definitions, and symbols in order to be useful.

12. The World Wide Web Consortium (W3C) is an industry consortium which has officially recognized XML as a "Recommended Standard." (The XML 1.0 specification is located at: http://www.w3.org/TR/1998/REC-xml-19980210.html.)

XML is not an ISO (International Standards Organization) standard, though SGML is. ISO is an international organization that sets forth the basic rules and procedures to construct SGML DTDs. W3C owns the copyright on XML, which it has licensed XML to the world, royalty-free, perpetually. Only W3C can change XML. However, if one changes XML or fails to perpetuate the copyright and its accompanying "copyleft" license, then that person infringes the W3C's intellectual property rights and the W3C can sue.

13. For a brief discussion of this aspect of XML, see Charles F. Goldfarb, "XML in an Instant: A Non-geeky Introduction," at http://www.oasis-open.org/html/goldfarb.htm.

14. October, 1999. The way XML works, because of another W3C Recommendation called "Namespaces," there cannot be a single exclusive (holy) standard. There can be several DTDs that work in harmony. Together, they can form a larger set of working standards. The larger set of standards can be national or even international with proper coordination and rational development.

15. A useful brief "history" of much of the Legal XML work has been written by Winchel "Todd" Vincent III, who leads the Legal XML initiative at the Georgia State University (GSU) Electronic Court Filing Project. See http://www.legalxml.org/Information/LegalXMLOverview.asp.

Sponsored by FindLaw and Georgia State University Electronic Court Filing Project