SciaNews

Data Exchange in the Pharmaceutical Industry

Data exchange, whether in the form of documents, databases, or other media, has become an essential part of daily business. Additionally, there is a growing need for companies to maintain data in a useable form for an extended period of time, particularly in the pharmaceutical industry.

A new drug application (NDA) to a regulatory agency requires considerable preparation of reports and supporting documentation. This process can result in substantial paper-output, making data and document management an issue for the sponsor and the reviewing regulatory agency.

There is a growing trend for electronic submissions. Such submissions are often referred to as "computer assisted new drug applications", or CANDA. CANDAs address the issues of paper-output and storage, and also provide a means to facilitate the review process by allowing information contained in reports, appendices and databases to be fully accessible by the regulatory agency. Electronic submissions have appeal since the entire submission can be provided on, for example CD-ROM, and the information can be easily archived.

In an effort to standardize new drug submissions, the MERS (multi-agency electronic regulatory submissions) project was started in January 1994. The MERS project is a mutual collaboration between regulatory agencies in North America, Europe and the Pacific Rim. The objective of the group is to develop standards for submission, review, and management of electronic regulatory information. The benefits of an international standard for new drug submissions include :

- shorter review times (time to first response)
- better submissions shorter approval times (time to market)
- consistency of reviews (good review practices)
- enhanced industry capacity to respond (time to market reduced)
- improved industry submissions (time to filing)
- facilitation of joint reviews
- facilitation of review sharing
- enhanced ability to accept foreign regulatory decisions

The purpose of this article is to look at the document management process and to consider one approach to maintaining the longevity of information.

The Problem

The critical question is "How to maintain information in a form that will endure as technology changes?". In order to grasp the problem, consider document exchange at its simplest level.

Suppose we create a document in Microsoft Word. Can someone using WordPerfect access this file? In most cases a Word document cannot be read by WordPerfect unless a "filter" exists which converts the document format from Word to WordPerfect. Such filters do exist but are limited for a number of reasons :

- File conversion between applications leads to loss of information as filters attempt to convert the structure of the document to one which is recognizable by the application.

- Software applications are continually enhanced. As new versions become available, it is not possible for an older version of the application to access documents created by newer versions. For example, a document created in Word 6.0 cannot be opened in Word 2.0. Conversion "up" is possible but conversion "down" is not.

- System configurations can lead to font substitutions which alter the appearance of the document.

We see that at a simple level, document exchange raises compatibility problems. Consider this problem on a larger scale. How confident can we be, given that technology is constantly evolving, that a decade from now, our information will still be accessible?

Document Formats & Viewers

The problem of information exchange across computer systems and applications has been addressed in several ways :

- Document/Application Standards
- Document Viewers

One approach to resolving this issue has been the development of document/application standards. RTF (rich text format) and PDF (portable document file) are two types of document formats. OLE (object linking and embedding) and OpenDoc are two types of application standards. Both attempt to allow users to exchange information on different platforms or systems and applications. Document formats such as RTF and PDF primarily describe text appearance and are proprietary. Application standards allow information exchange between applications and are also proprietary.

Most software companies provide "document viewers" to view certain document formats, even if the user does not have the application in which the document was created. However, the viewer allows the user to "view" and "print" the document only - i.e. no editing is possible, and these viewers remain system dependent.

We need a data format that is independent of :

- system and platform - files are useable across systems
- version - files can be updated or reviewed without loss of information
- application - files can be exported to any application format
- vendor - non-proprietary format

The ASCII format fulfills these requirements but is severely limited by not allowing for formatting the visual appearance of the information. An alternative solution, with perhaps the most potential, is SGML.

What is SGML?

SGML is an acronym for *"standard generalized mark-up language"*. It is an international standard (ISO 8879) for electronic document exchange. A more widely used derivative of this language is HTML ("*hypertext mark-up language"*), the de facto standard for document publishing on the world wide web (WWW). As an international standard, SGML is

- non-proprietary (no-one owns it)
- system independent
- software independent
- internationally supported

Accordingly, SGML appears to satisfy the requirements of open data architecture. It must be emphasized that SGML is a language for describing the *structure*, not the *appearance* of text. It is a language based upon syntax and rules for identifying and enforcing structure.

SGML - Structure, Content and Format

To gain an understanding of what SGML can do, again consider a simple document. A document can be divided into three segments- structure, content and format (style). How does SGML work with each segment?

Structure

The structure of a document is its organization or arrangement. In SGML, the idea is to break down the structure of the document into identifiable components or elements with relationships. For example, where a document consists of chapters, the "chapters" themselves can be broken down into the "chapter heading" and the "paragraphs" within the chapter. Each of the elements can then be further broken down into, for example, "paragraph headings" and "sub-paragraphs". So we can think of a document structure as an hierarchical tree consisting of main elements, sub-elements and perhaps even sub-sub-elements. Each element of the document is identified and the relationship between elements can be defined.

The *structure* of the document is defined in the DTD (document type definition), which usually accompanies the SGML document. In this file, the elements, relationships and attributes of the document are defined. The DTD is used as a template for writing the document: it ensures that the document structure follows the organization defined. In this way, the DTD frees the author to focus on content.

Content

The content of a document is the information itself - the text, graphic and even audio or video which is inserted into the document. The contents are surrounded by "tags" identifying the position of the information, relative to the rules defined in the DTD. The contents of the SGML document must conform to the rules outlined in the DTD. For example, suppose the DTD defines an element called "title". Then for a given text string "Clinical Management", the SGML tagging would be :

```
<title> Clinical Management </title>
```

Note the method of marking the beginning and end of this string by the tags "<title>" and "</title>, respectively.

Format

Format refers to the visual representation of text (bold, italic, ariel font, etc). SGML however, is *not* concerned with the visual appearance of the text. Recall, the objective of SGML is to maintain the structure of information. If SGML is concerned only with *structure*, how can we specify a *format* for output? The strength of SGML lies in its flexibility to output data in various mediums (printed or electronic output). Output specifications specify the output formatting of a document. Although SGML is an open standard, output specifications tend to be proprietary. As a result, the various SGML software currently available have different ways for specifying output. Output specifications for each element in the document are usually defined in *style sheets*.

The proprietary nature of output specifications are currently being addressed by two developments for output standards - FOSI (format output specification instance) and DSSSL (document style semantics and specification language). Both attempt to provide output formatting which is vendor-independent.

The Document Type Definition (DTD)

We have discussed how SGML works relative to *structure*, *content* and *format*. In the section on structure, two keywords were introduced - the *DTD* and *elements*. Recall, the DTD is a series of rules which define the relationship between elements. It consists of declarative statements for elements, entities and attributes. Elements have already been discussed briefly - they are the structural components of the document. An *entity* can be a short text string which replaces complex strings, groups of elements, special characters or graphics. For example, symbols such as "" are not available on a standard keyboard. In SGML, we define an entity for this character. *Attributes* modify elements, that is, they define the characteristics of an element.

Table 1 : SGML

Component	Definition
DTD	Document type definition file. A file containing the rules and descriptions which define the structure and content of the document. It is a template for a document marked up with SGML.
Elements	Components or building blocks for information.
Entities	A string of text characters, symbols, text file or graphic file which can be referenced as a unit in the document. E.g. the entity for "-" is "&ndash".
Attributes	Characteristics of elements.

SGML and Word Processing Programs

How can we relate SGML to word processing programs? SGML is a language, not a program. In word processing programs, styles can be defined within a template. Although it appears that the *style sheet* is similar to that of the *DTD* in SGML, the DTD is not at all related to style sheets. This is because word processing programs allow users to define styles within a template, thereby implicitly defining structure. Although style sheets in word processing programs define "tags" for the different elements of the document, they do not *enforce* structure. Relating *format* to *structure* is a common misconception. In word processing programs, document structure is the responsibility of the author; SGML requires the author first to define the document structure.

An Example

The relationship between a DTD and an SGML document will be illustrated by an example. Figure 1 graphically represents the structure of an SGML document as defined by a DTD. The structure of the document is defined by the DTD in this figure to have *required* "Study Title" and "Protocol Number" elements, but *optional* "Version Number" and "Revision Date" elements. Now consider the SGML document in Figure 2. Will the DTD accept this document? Yes, because it contains the required elements "Study Title" and "Protocol Number"; the elements "Version Number" and "Revision Date" were defined in the DTD as *optional*. Figure 3 illustrates an SGML document which will be declared invalid by the DTD because, although it contains the optional elements, it does not contain the *required* elements. Thus, we can see how the DTD defines and controls the structure of a document, and the flexibility of this template.

SGML Tools

The tools of an SGML system vary according to function. In general, there are tools for data design and validation, input, output and information management. Some basic tools and their functions are summarized in Table 2.

Table 2 : SGML Tools

Tool	Purpose/Function
Parser	Parsers read, interpret and process an SGML document and DTD. The DTD is checked against the rules of the ISO standard. The SGML document is validated against the DTD.
DTD tools	Create, view and analyze the DTD
Input tools	Editors. These range from the simplest text editor to the complex native editors. The difference lies in the SGML validation. Simple editors do not check SGML validation until the document is exported or converted. Native editors provide immediate feedback while the document is being created. Although any editor can be used to create an SGML document, it is often preferred that editors with capabilities for document validation are used.
Output tools	1. Output specification of formats (style sheets) 2. Viewing of SGML documents (viewers and browsers) 3. Filters, converters - convert to and from SGML
Information management tools	Tools for transformation, search, retrieval and management of SGML data.

In most cases, the tools for DTD, input and output are often available in a single software package from vendors. WordPerfect SGML (Corel Corp.) is an authoring tool with utilities for DTD validation, SGML input and output. SGML Author for Word (Microsoft Corp.) is an add-on to Microsoft Word to allow users to open, create and save files in SGML format. Panorama Pro (Softquad Inc.) is a browser/viewer for SGML documents. Near & Far (Microstar Software Ltd.) allows the user to create, design and validate DTDs.

Summary

The applicability of SGML as a possible solution to maintenance and exchange of information was briefly discussed. To summarize, the advantages of using SGML are :

- separation of format from content
- multiple output formats (e.g. paper output or CD-ROM)
- multiple applications of data (data reuse)
- information to the data for computer applications (e.g. databases, images)
- longevity of data is assured (human readable and machine processable)
- integrity of data is maintained (ISO standard)
- user definable tagging
- electronic exchange of data with other users (portability)

For further information …

The table below lists some useful links on the World Wide Web (WWW) for additional information relating to SGML.

Item Uniform Resource Locator (URL)

MERs Project http://www.pharmasoft.com

SGML Info http://www.sil.org/sgml/smgl.html
http://www.arbortext.com/wp.html

SGML Software Vendors http://www.interleaf.com
http://www.softquad.com/products/
http://www.corel.com
http://www.microsoft.com/msword/productinfo/sgml/
http://www.arbortext.com
http://www.microstar.com