ELECTRONIC PUBLISHING
BUILDING ON SGML

Dr. Bob Jansen
Consulting Director, Technology & Research
Impart Corporation
Brisbane Qld
Email: bob.jansen@cmis.csiro.au

1. INTRODUCTION

Electronic publishing offers many opportunities for broadening the readership of information as well as the increasing the bandwidth between author and reader leading to a more satisfying experience for both: authors can produce information in the most appropriate form whilst readers can experience the performance more widely.

This enriched communication is made possible through the transformation of the material into electronic forms and their availability through electronic communication networks utilising increasingly more powerful desktop computers. All of this hinges on standards for representing the material without constraining its possibilities. Markup languages have been developed to assist in this task.

SGML (Standard Generalised Markup Language) is increasingly being adopted by publishers and libraries as the primary standard for producing electronic documents. However, SGML, albeit a relatively new language, is already criticised by many as too difficult and cumbersome. It is not the panacea to the problems of electronic documents, but is one weapon in our armoury. Other technologies have been developed and in some circumstances, may be more applicable.

This paper will discuss some of these options and provide experience from a research project, EPICentre[1], into the electronic publishing of scholarly journals.

2. DOCUMENT MARKUP

Document markup is the term commonly applied to the process of converting a document into a form suitable for publishing electronically. It is not generally recognised, however, that document markup also converts the document into a form suitable for reading electronically.

[1] EPICentre was a collaborative project involving CSIRO Division of Information Technology, InfoRMIT, CSIRO Publishing and Monash University and was funded, in part, by a grant from the AVCC Electronic Publishing Working Group.

In fact, the history of reading/writing demonstrates quite clearly that these two ends of the communication channel are inseparable. Any change in one echoes to the other. Manguel (1996) (pp178-9) attributes this dependence to the invention of writing itself:

> 'The inventor of the first written tablets may have realised the advantage these pieces of clay had over holding memory in the brain: first, the amount of information storable on tablets was endless - one could go on producing tablets ad infinitum, whilst the brain's remembering capacity is limited; second, tablets did not require the presence of the memory-holder to retrieve information. Suddenly something intangible - a number, an item of news, a thought, an order - could be acquired without the physical presence of the message-giver; magically, it could be imagined, noted and passed on across space and beyond time...With a single act - the incision of a figure on a clay tablet - that first anonymous writer suddenly succeeded in [overcoming the obstacles of geography, the finality of his death, the erosion of oblivion]…
>
> But writing is not the only invention come to life in the instant of that first incision: one other creation took place at the same time. Because the purpose of the act of writing was that the text be rescued - that is to say, read - the incision simultaneously created a reader, a role that came into being before the actual first reader acquired physical presence...The writer was a maker of messages, the creator of signs, but these signs and messages required a magus who would decipher them, recognise their meaning, give them voice. Writing required a reader.
>
> The primordial relationship between writer and reader presents a wonderful paradox: in creating the role of the reader, the writer also decrees the writer's death, since in order for a text to be finished the writer must withdraw, cease to exist. While the writer remains present, the text remains incomplete. Only when the writer relinquishes the text, does the text come into existence. At that point, the existence of the text is a silent existence, silent until the moment in which a reader reads it. Only when the able eye makes contact with the markings on the tablet, does the text come to active life. All writing depends on the generosity of the reader.'

In not recognising this "primordial relationship", it is fair to say that the majority of systems for marking up a document do not address the issue of reading except through the assumption that if the document is displayable on a computer screen, then it must be readable. In addition, the electronic manipulation of a document creates a feedback process to the act of authoring whereby a reader can now alter the document itself and become an author.

3. MARKUP LANGUAGES

Since the development of writing, a complex and rich language has evolved to assist in the transmission of intricate concepts using print. This language encompasses size of pages, types of letters, styles of print, layout of pages, layout of documents, features to assist in locating specific items, methods for recognising ownership of chunks of content, citations to other documents, etc. The language covers issues associated with both authorship and reading. Aspects of this language associated with authoring are usually found in style guides, eg. AGPS (1995), whilst our education systems imparts those aspects related to reading.

In the electronic domain, this rich lode of knowledge breaks down as we increasingly manipulate information using seemingly endless scrolling windows on differently capable computer monitors. In fact, in the electronic domain we are in danger of losing the 'page turning' activity, an action that physically breaks down the stream of information into more easily integratable chunks (see for example, Drucker, 1995). More importantly, this rich and complex knowledge that would have been most valuable for using a document in creative ways fails to be incorporated into the document and thus is now not available at reading time (Maler and El, 1996).

SGML is one mechanism for capturing aspects of this meta information, information about organisational structure and display characteristics. The use of SGML as an inherent component of the authoring process ensures this meta information is available to the reader.

4. DOCUMENT MODELS OR DOCUMENT TYPE DEFINITIONS (DTD'S)

Since the invention of printing, many types and formats of documents have been produced. Until the interposition of computer technology, this was not a problem due to the flexible and powerful processing capabilities of the human brain coupled with the versatile eye.

The new technology, however, is incapable of coping with the immense diversity of, and variation in, documents in their conventional format. This limitation is especially obvious when the solution of a task is dependent on an understanding a document, or corpus, ie. provide a list of all documents without an abstract, or a list of documents authored by Jansen, or journal articles on behavioural science, etc. To cope with these tasks, researchers proposed a scheme of defining the type and structure of documents.

Document models, or document type definitions, were thus developed as a method of limiting the scope of the problem by enabling the author of a document to explicitly define the characteristics of their document, rather than leave this to be determined by the reader. This development enabled authors to either select a suitable type from a list of known types or to build their own specific type definition, as part of the authoring process resulting in the document marked up according to its type definition.

The use of 'standard' type definitions enables the document's content to be communicated to the reader's computer where a copy of the standard type definition is used to present the content to the reader. For a proprietary definition, this must now be transmitted in concert with the document's content. The document type definition, or model, acts analogous to a style guide, albeit a very specific style guide, providing the instructions for rendering the document's content on the user's output device, eg. monitor, printer, etc.

Once a document type definition is known, the content of the document is able to be processed in a limited fashion by automated document processors who parse the type definition to identify relevant document characteristics.

5. DOCUMENT ‘STANDARDS’

In line with the varied types of documents being manipulated, there are a number of competing ‘standards’ for content representation. Each of these addresses some aspects of representation but few if any address the issue of reader interaction outside the existing facility of ‘convert to paper and read as normal’. The number of available standards exceeds the ability to address them all, and in this paper we will focus on those standards commonly applied to electronic publishing.

5.1 PostScript

Probably the most widely used standard related to electronic document systems, PostScript (Adobe, 1985) was developed as a mechanism for printing onto paper using the then recently developed Xerox electrostatic imaging technology (now widely encountered in photocopiers and laser printers). PostScript has the advantage of market penetration in that a postscript device will be found in nearly every office in the world.

PostScript, unlike other markup systems, is focussed at the rendering of a document’s content and has no application to document structure, logics, etc. The development of PostScript by Adobe Systems and its application in laser printers by Apple Computers gave birth to the desk top revolution. PostScript is supported by every non-trivial word processor application and in most cases, postscript support is provided by the operating systems themselves. This makes PostScript a widely used technology and thus attractive for remote printing of electronic information.

5.2 Portable Document Format (PDF)

Developed by Adobe Systems, PDF represents a technology capable of delivering ‘electronic paper’ via a communication network. Built on Adobe’s expertise in PostScript, PDF is capable of being generated from any valid postscript file and thus is widely supported across many computer platforms and software packages.

PDF is a reader-centric standard mimicking paper in an electronic form. Documents are accessible via page numbers, are displayed as ‘pages’ with page characteristics, ie. headers, footers, margins, gutters, etc. Pages are annotatable but only if the reader has purchased the reading tool with annotation functions: the simple reading tool, Adobe Acrobat Reader, does not support annotations.

PDF is being increasingly adopted for electronic documents. The availability of a PDF plug-in for web browsers facilitates the reading of PDF documents using conventional web browser interfaces, thus ensuring readers need only the single software environment provided by their web browser. The US National Archives recently announced it was trialing PDF as the archiving standard thus commencing the move from a paper-based archive to an electronic archive.

From a publishers perspective, PDF is attractive because once converted, the content of the document is unalterable by the reader. In addition, the publisher can value add the content by the provision of hypertext-like navigation structures, but the navigation is constrained to be within the current document. PDF, in mimicking paper, also conforms with existing management processes in that the object being processed, ie a document, remains identifiable as a document: its characteristics are enhanced not changed, and hence it is still shipped, billed, etc. This is an important attribute as our society continues to view and manipulate documents as single objects.

From a reader perspective, PDF is limiting because page sizes are defined when the PDF file is created and thus bear no relationship to the actual size of the reader monitor or other characteristics. Inter- and intra-page navigation is clumsy, analogous to moving a real page beneath, for example, a smaller window. PDF readers support the zooming in and out of documents, but this does not affect font characteristics, and hence zooming out reduces text readability. Zooming in aggravates intra-page navigation. However, the annotation capability is very good in that annotations are separate from the document's content and 'float' above the page. Annotations can be colour coded, akin to using coloured Post-it[™] pads.

If the requirements is for unalterable 'paper', then PDF is an excellent choice recognising its limitations.

5.3 Standardised General Markup Language (SGML, ISO 8879)

SGML is a language for recording and storing document information. The information is written as a set of rules that a group of related documents should follow. The process of determining these rules is a modelling process and culminates in the development of an SGML Document Type Definition (DTD). The DTD can be used to guide the authoring process ensuring that the document's content does not contradict the blueprint. The DTD can be used to maximise the fidelity of the rendering process so that the anticipated structural visualisation and document style is what is rendered on the reader's output device.

In SGML, a document is defined as a collection of information that is processed as a unit. This broad definition enables SGML to address both a newspaper and an individual article in a newspaper as a document. SGML can model whatever level or size of document is appropriate for a particular purpose. A conventional document is made by its medium and hence two different renderings of the same content would constitute two documents. SGML, however, treats the content and its markup as the document as distinct from its rendering, the presentation instance. Thus an SGML document may be considered as the source of potentially unlimited number of presentation instances.

[™] "Post-it" is a trademark of 3M

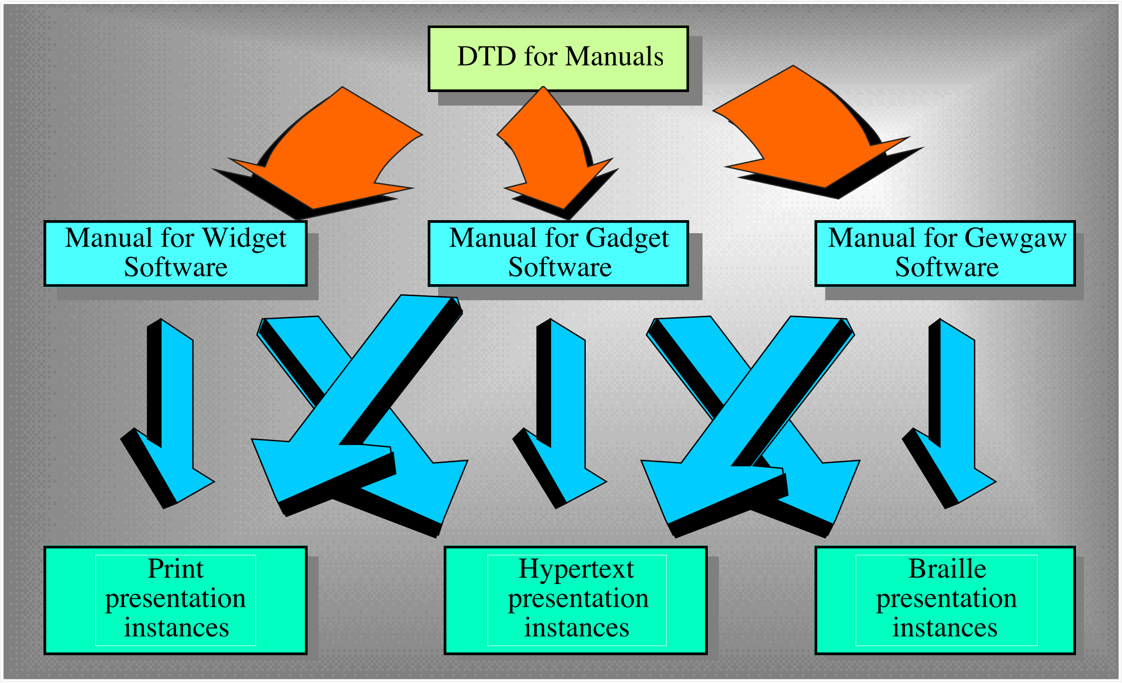
DTD for Manuals
Manual for Widget Software
Manual for Gadget Software
Manual for Gewgaw Software
Print presentation instances
Hypertext presentation instances
Braille presentation instances

Figure 1. SGML Document and Presentation Instances (adapted from Maler and El, 1996)

SGML markup has several advantages, namely:

- It is declarative which facilitates multi-purposing of each document and its interchange with others who might use the document in different ways;
- It is generic across systems and is an accepted international standard which maximises a document's survival across hardware and software changes; and
- It is contextual which facilitates the validation of document structure and ensures adherence to this structure by authors.

5.3.1 ISO 12083 (AAP, Standard for Electronic Manuscript Preparation and Markup, Electronic Markup Series, Association of American Publishers, Washington, D.C., August 1987)

The ISO 12083 standard defines a particular DTD that has been developed by AAP in the USA and made more generic through a number of implementations. This 'generic' DTD can be applied to any number of electronic publications, but experience has dictated that its application will necessitate customisation to the particular publication at hand (see below). The DTD, however, serves as an eminent starting point for the development of a proprietary DTD.

The DTD is capable of supporting various document types, including books and journal articles. Experience has indicated, however, that realistic use of the DTD is unlikely without some customisation.

5.4 Office Document Architecture (ODA, ISO 8613, Information Processing Text and Office Systems Office Document And Interchange Format)

As described in Ressler (1997), ODA is an electronic document standard not limited to the structure of a document but addressing also how a document should look. Unlike SGML, ODA is an entire framework for representing both the structure and the visual presentation of the various components that comprise the document.

ODA documents have a logical structure and a layout structure. The logical structure of a document is very similar to SGML document structure as defined through a DTD. The layout structure, describes the way document components, eg. page sets, pages, frames, and blocks, should be rendered on the output device. A layout directive maps between the logical and layout structures. The content itself forms a sort of interface between the logical and layout trees of an ODA document.

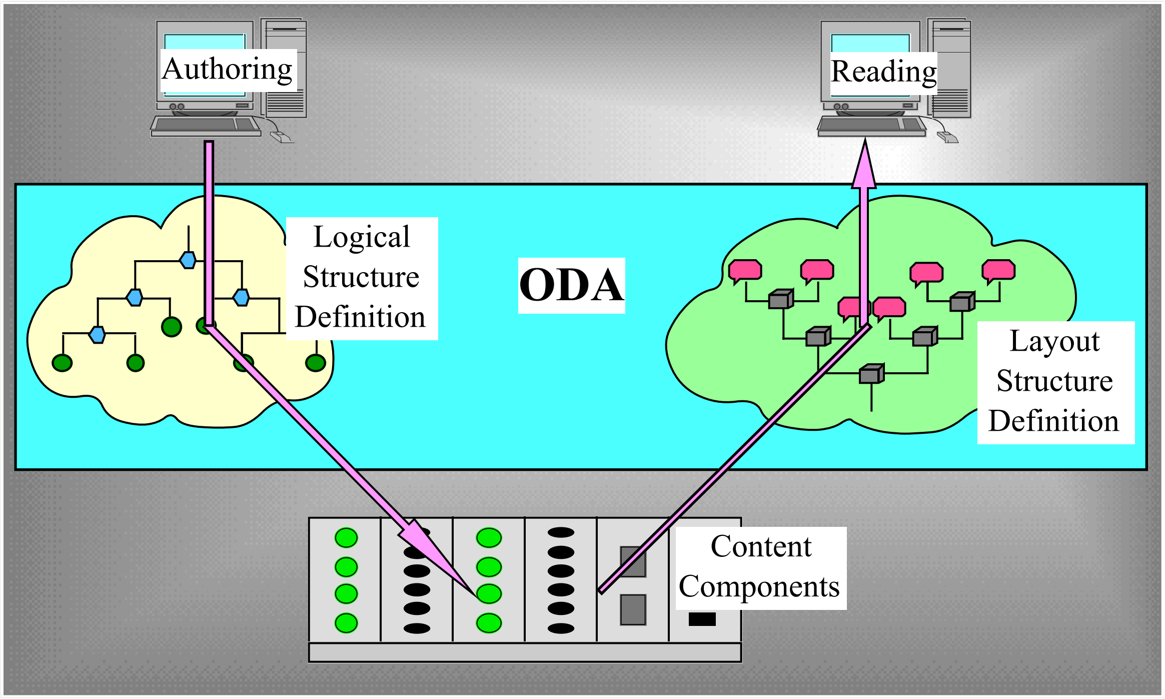

Figure 2. ODA Architecture

It is important to clearly understand that ODA is an interchange standard. If one system’s idea of an object to be rendered is not the same as another's, interchange will suffer. ODA attacks this difficult problem by standardising the representation for the component parts.

ODA and SGML take fundamentally different approaches to electronic document standardisation. SGML is a robust extensible language that focuses on document structure and does not address document layout. ODA provides an architectural framework that addresses both document structure and layout.

Todate, support for ODA is lacking, except for certain areas in Europe. Tools and technologies are difficult to obtain and even then, ability to process ODA formatted documents is scarce. The ODA architecture, however, is elegant and comprehensive. Certain ODA advantages are now being implemented using the DSSSL and XML activities (see below).

5.5 HyperText Markup Language (HTML)

Hypertext Markup Language is a language for programming the behaviour of a client browser for rendering document content. HTML is added to a document to specify style information, eg. bold, italic, underline, font sizes, etc, as well as tabular information. Headings are identified but their rendering is totally dependent on the settings of the client browser. The client browser receives a stream of HTML marked up information which it interprets, mediated by its user preferences, leading to the rendering of the content on the client terminal. HTML additionally, enables the identification of *anchors* and *links*, structures capable of representing non-linear navigational pathways through an information space. The wide usage of HTML's simple anchor and link facility is a major driver behind the explosion in the World Wide Web.

Although initially an uncomplicated markup, HTML soon proved too simplistic for real world applications and thus the language is undergoing major reforms. One early reform was the description of the HTML language using SGML and thus HTML can now be considered a SGML DTD. See also the section on XML as a major development that may subsume HTML altogether.

Like SGML, HTML provides little support to the reader of a document. The browser in use on the client's workstation interpret tags and formats content according to simple mapping rules. This client side behaviour can even change the author's intention unless the author is aware of the behaviour of browsers to specific HTML commands and the reader has not overridden default mapping rules. However, the simplicity and ease of use has ensured that HTML is the premier electronic publishing language today supported by a myriad of tools and technologies.

5.6 Document Style Semantics and Specification Language (DSSSL, ISO 10179, ftp://ftp.ornl.gov/pub/sgml/WG8/DSSSL/)

The Document Style Semantics and Specification Language is a language for specifying document transformation and formatting in a platform independent and vendor independent manner. In particular, it can be used to specify the presentation of documents marked up according to SGML.

DSSSL consists of two main components: a transformation language and a style language. The transformation language is used to specify structural transformations on SGML source. For example, a telephone directory structured as a series of entries ordered by last name could, by applying a transformation specification, be rendered as a series of entries sorted by first name instead or to specify the merging of two or more documents, the generation of indexes and tables of contents, etc. The style language provides mechanisms for specifying style information to the rendered content using style sheets, etc.

In adding this capability to SGML, there appear to be little differences, at a conceptual level, between SGML/DSSSL and ODA technology, in that both now begin to cater for the rendering of the content on the reader's workstation. The SGML/DSSSL technology seems more powerful in being able to re-format content so as to provide services based on the content.

5.7 HyTime, ISO/IEC 10744, Hypermedia/Time-based Document Structuring Language

HyTime provides a worldwide standard technical framework for integrated open hypermedia. In terms of syntax, HyTime is an application of SGML. In terms of functionality, however, HyTime extends the power of SGML through addressing, validation, inheritance from multiple parents, linking, scheduling and re-use.

With HyTime, everything becomes addressable in any convenient terms at any granularity. Attribute values and data content can be checked for conformance to language models, references can be constrained to refer to particular kinds of things. HyTime greatly enhances the object-orientedness of SGML. Elements can inherit semantic and syntactic features not only from the governing DTD, but also from any number of other parent DTD's enabling the re-use of software supporting semantic processing. HyTime facilitates the re-use of document components in other documents without copying, the scheduling of the rendering of components in time and/or space and the association of access policies.

5.8 The Extensible Markup Language (XML, http://www.w3.org/XML/)

The Extensible Markup Language (XML) is being developed under the auspices of the World Wide Web Consortium (W3C) as a means for delivering structured documents over the Web. It is a scaled-down version of SGML that is a lot more powerful than HTML but much easier than SGML thus bringing the key benefits of generic SGML to the Web in a manner that is easy to implement and understand while remaining fully compliant with the ISO standard.

As in the case of HTML, the implementation of XML on the Web will require attention not just to structure and content, but also to the standardisation of linking and display functions. A key design feature of XML is its clear separation of syntax from other processing behaviours, the explicit standardisation of the most important of those behaviours (linking and stylesheets) is a necessary part of the XML activity in order to ensure the vendor- and platform-neutral interoperation of XML documents.

The XML development strategy will deliver through three phases: a specification of the XML syntax suitable for Internet applications: a specification of standard hypertext mechanisms for XML applications based on HyTime (ISO/IEC 10744) and the Guidelines of the Text Encoding Initiative; and the specification of a standard stylesheet language for XML publishing applications based on DSSSL.

5.9 Handheld Device Markup Language (HDML, http://www.w3.org/pub/WWW/TR/NOTE-Submission-HDML-spec.html)

The Handheld Device Markup Language (HDML) is a simple language used to create hypertext-like content for small display, hand held devices whose characteristics are not supported by the HTML language. In particular the navigation and display models inherent to HTML collapse when applied to a typical handheld device. HDML, by combining the use of standard web protocols with an alternate but complementary markup language, allows hand held devices to function as full-fledged web clients.

While there are many types, styles, and classes of handheld devices, HDML is useful for a significantly larger class of constrained by: small display size; limited input capabilities; limited bandwidth; and limited resources such as memory, processing power, permanent storage, battery life, etc.

In specifically addressing these hardware/network characteristics, HDML is one of the few markup languages that considers the end user and their interaction with the down loaded information, ie. the reading process. It should be noted, however, that HDML is currently being discussed within the World Wide Web Consortium and not a recognised or adopted standard.

6. EPICENTRE AND LESSONS FOR LIBRARIES

The aims of the EPICentre project (http://www.informit.com.au/epicentre/) were to investigate the processes of electronic scholarly journal publication. It targeted two scholarly journals: PSYCHE (http://psyche.cs.monash.edu.au), an existing electronic journal, and the Australian Journal of Chemistry (AJC, http://www.publish.csiro.au /journals/ajc/index.html), an existing paper-based journal. The activities of the project were confined to converting both journals into suitable form for web serving, a survey of users of both journals to determine their comfortability with an electronic form and an investigation into the transformation issues and DTD design.

6.1 End user requirements vs publisher’s intent

Early on in the project, it became obvious that any activity in defining models/DTD's must take into account end-user requirements as well as publisher's intent. Without an understanding of these two constraints, any DTD or document model is unrealistic and will necessarily be incomplete: serving one or the other stakeholder but not both. The publisher and end-user requirements provide a context within which a document may be marked-up with a mark-up supporting those requirements. For example, suppose a reader required word-level searching of the document. Unless the mark-up identified each word, such searching would be impossible. Similarly, suppose the publishers wanted to explicitly disambiguate commentary from the core text: without a suitable mark-up scheme this would be impossible.

Thus the early work on DTD's was involved with the preparation of the questionnaire for dissemination in both AJC and PSYCHE. In addition, an informal poll was accomplished in an effort to determine the different types of facilities desired by different types of readers.

The formal poll provided little new information: the respondents, all but one of whom subscribed to the PSYCHE electronic journal, were all comfortable with electronic journals and had the appropriate equipment to interact with electronic information spaces.

The informal poll was more interesting and was conducted as a series of interviews with some CSIRO colleagues, all of whom were electronic-information workers. Abroad cross section of staff was chosen, from computer literate to novice, electronic publishing literate to novice, to get as broad a response as possible. Although informal, and obviously not statistically significant (sample size of 12), interesting views were obtained.

The main conclusion reached through this informal poll was that most information workers look for the commonly-accepted facilities offered by electronic documents, eg. free-text searching, sharing of publications, simple annotation, etc: all facilities offered by the notion of ‘electronic paper’ value added through its embodiment in the World-Wide Web. Few stretched the boundaries of what is known into what is possible. The information professionals amongst the group looked at electronic cataloguing and indexing, but then mainly from the perspective of electronically-aided cataloguing and electronically-aided indexing. There were no requests for active documents, derived documents, or linked documents.

The results of the poll indicate either an overall lack of awareness of the capabilities of the electronic document or a misunderstanding of the questions. The latter is unlikely, in that the sessions were of some length and there were plenty of opportunities and cues to elicit those ‘advanced’ requirements.

Why then, a lack of overall awareness of the possibilities?

The WWW is widespread and usage, amongst information workers, is high. The capabilities offered, however, progress little beyond `electronic paper' coupled with hyperlinks. The recent emergence of Shockwave, Java and Virtual Reality Modeling Language (VRML) provide, at least technically, the capability to animate a WWW page. Active pages have been available for some time through the Forms facility supported by the HTML protocol: animation however represents a further step by broadening the bandwidth of communication from author to reader. The reverse communication, ie. from reader to author, is not necessarily broadened, depending on the facilities provided by the applets associated with each animation.

The issue being identified here, and in the above discussion, is that the current situation compares to a technology-push and not a market-pull. Although this is customary with the introduction of new technologies, there are signs that the end-user community is beginning to demand more consideration and that the technology providers are becoming more aware of their technology’s limitations. For example, animation for animation’s sake is not accepted anymore: many web sites have had to be drastically re-designed away from high quality image formats to a greater dependency on textual content due, in part, to unacceptable down-load times but more importantly to a negative response to interactivity not adding to the information exchange.

6.2 Application of ISO12083

EPICentre trialed the application of the DTD detailed at ISO standard 12083. The DTD was not really suitable to application for the Australian Journal of Chemistry as there was little direct support for equations and foreign character sets, chemistry requiring many Greek symbols, superscript and subscript characters, etc. In addition, the DTD enforced a rigid structure that did not mirror the case with the Australian Journal of Chemistry. The AJC editing style was loose, with in some publications, two identically titled sections appearing in reverse, or a section not appearing at all.

To successfully apply ISO12083 would have necessitated the editing of the DTD to loosen some of its restrictions, to expand its capabilities in terms of allowable character sets section sequencing, etc, or the tightening of AJC editorial guidelines. Although the latter seems the most favourable, this might have alienated authors and readers, as well as changed the look and feel of the journal - both situations to be planned with great care.

6.3 Conversion into electronic forms

A large effort in the EPICentre project involved the conversion of the journal content into appropriate electronic formats. The PSYCHE journal was converted easily as the data was already served as discrete ASCII files and thus the process consisted of tagging the existing electronic formats using the HTML language.

The AJC journal, however, existed only on paper and thus the content had to be transformed and converted. Unfortunately, and probably realistically, the journal contents could not be found in electronic form except for a set of discs submitted by the authors - unedited text with diagrams held as separate proprietary-format files. The process of merging the diagrams with the text proved extremely time consuming, but the subsequent tagging was a greater problem. The AJC was converted into a three formats to compare the effort involved: PDF, HTML and SGML.

The conversion into PDF was simple consisting of 'printing' the document using the appropriate PDF driver to take the intermediate postscript file and converting to PDF form. Once converted, the document was immediately available for delivery.

The HTML form was derived through the application of the Adobe Pagemill HTML editor which had just been released. This editor made the task repetitive but simple, but showed the lack of consistent editorial policy had expensive effects on this process. Each document had to be done individually and continual referral was made to previous documents to apply a consistent style given the lack of style sheets.

The conversion into SGML proved completely unmanageable due to the lack of good tool support. As many AJC articles contain multiple large tables, and each cell of each row of each table had to be manually tagged, this was a very time consuming and expensive process. The tools used indicated they were designed for the authoring process not the conversion process. Investigations were made into suitable batch conversion tools, eg. Omnimark, but the lack of consistent editorial policies and style sheets meant such batch conversion would be extremely inefficient and unproductive over hand conversion[2]. The SGML editors trialed proved unsuited to the task and the whole experience flagged serious concerns with the conversion of legacy data.

7. CONCLUSIONS

This paper has described some major activities in supporting the electronic publishing of documents and some concerns regarding the support for reading of these documents in an electronic form.

Reading can not be separated from authoring: both must be considered in any project delivering electronic information. Each is related to the other and should be seen as roles that a person can play rather than jobs people hold. In reality, we all annotate documents we read, we form our own views of their content and remember those views using our own tables of content and indexes. It is these processes that really adds value to one of our documents.

The hyper linking of two or more documents, if continued in the uncontrolled manner in evidence today, will ensure that eventually every document will be linked to every other

[2] SCANTEXT, a commercial data conversion organisation, endorses the re-keying of data over its conversion for reasons of accuracy and cost.

document thus reducing the value of such linkages to nothing. It must be recognised that links are made between chunks of information for particular reasons: knowing those reasons and knowing what is being linked to form two powerful analytical techniques for deciding whether to follow a link. These two characteristics of links must be made explicit within electronic documents.

Existing technologies are slowly moving to support reading processes by beginning to consider how to present content taking account of the characteristics of the reader's workstation. Markup languages are evolving and new ones are being created. The problem surfacing now, for an author/publisher/library is which markup language to choose. Even the usual adage of 'choose an international standard' is in danger of being inappropriate in that many markup systems are being considered, or have achieved, international standard status.

The conversion of legacy data represents a major challenge to organisations and existing technologies. Many of the available tools are designed to be used by authors: many conversion tools are predicated on strict adherence to style guides which although desirable may be unrealistic. The conversion of paper data through scanning technologies is getting better but still not 100% accurate and completely fails to cater for hand written texts.

Where does that leave libraries and other content holders moving into the electronic age? Essentially, take care, analyse and understand the requirements from the perspective of all stakeholders, choose the right technology for the job, and be prepared to admit an error has been made and start again.

8. ACKNOWLEDGMENTS

The information reported in this paper, especially the EPICentre project results, owes much to the work of others. In particular, we acknowledge the input from the EPICentre team: Paul Reekie (CSIRO Publishing), Andrew Treloar (Deakin University), Don Schauder (Monash University), Patrick Wilken (Editor, PSYCHE), Tom Denison (White Room Electronic Publishing P/L) and Paul Mercieca (InfoRMIT).

9. REFERENCES

Adobe Systems, Inc., *Postscript Language Reference Manual*, Addison Wesley, reading, MA, 1985

AGPS, *Style Manual for Authors, Editors and Printers*, Australian Government Publishing Service, 1995

Drucker J, *The Century of Artists' Books*, Granary Books (pub), 1995

Maler E & El Andaloussi J, *Developing SGML DTD's From Text to Model to Markup*, Prentice Hall PTR, 1996

Manguel A, *A History of Reading*, Harper Collins, 1996

Ressler S, *ODA*, URL=http://www.prenhall.com/electronic_publishing/html/chapter5 /05_5.html, in *The Art of Electronic Publishing The Internet and Beyond* (URL=http://www.prenhall.com/ electronic_publishing/), Prentice Hall, 1997