Electronic Publishing Specialist Group
of the British Computer Society

Paper circulated prior to KIDMM: an EPSG-hosted meeting on shared interest among BCS groups about the management of knowledge, information, data and metadata.

Knowledge, information and data management – and technology

Conrad Taylor

DISCLAIMER

This paper has been written by Conrad Taylor of BCS-EPSG. The views and ideas herein are not necessarily those of EPSG nor of the BCS.

They have been circulated to provide one starting-point for discussions at the KIDMM event on 6 March 2006.

BCS
THE BRITISH COMPUTER SOCIETY

February 2006

http://www.epsg.org.uk/KIDMM/

Knowledge, Information and Data Management – and technology

Conrad Taylor [1]

1 Introduction

The purpose of this paper is to initiate a conversation, primarily between Specialist Groups (SGs) of the British Computer Society. It was created in April 2005, and has been revised in February 2006 in preparation for an EPSG-hosted workshop on 6 March 2006.

Having groups that specialise in different aspects of computer use is a great thing, but also has disadvantages, especially the compartmentalisation of knowledge. As computers become useful to society in more and more ways, new topics are bound to emerge that will impinge on the scope of several BCS Specialist Groups, not falling cleanly within any single SG's domain. When topics are inherently cross-disciplinary, it may make sense to find ways of bringing the combined intellectual power of several BCS Specialist Groups – and other external organisations – to bear on them.

I suggest that the practice of Knowledge and Information Management is one of these large cross-disciplinary topics – together with which I would bracket the tasks of organising the presentation and dissemination of information (i.e. publishing it, in the broadest sense of that word).

In the pages that follow, I explore what this topic encompasses, in relation to the growing use of computers for these tasks. My initial working shorthand term for this was **KIMtec** (Knowledge and Information Management technologies), in which name I set up a small electronic discussion group.

The ideas and opinions expressed in my paper are informed a great deal by discussions within the BCS, especially in the Electronic Publishing Specialist Group of which I am a member. However, the opinions shouldn't be taken as the views of EPSG or any other body. Finally, I must confess to having an 'electronic publishing' slant to the way I view this subject! I look forward to having that counterbalanced in further discussion…

1. This discussion paper is available from the following URL:

http://www.epsg.org.uk/KIDMM/

2 Some definitions

What is information? knowledge? data?

It's hard to have a discussion of knowledge and information and data without *some* definitions, so that's where I shall start – even though my definitions may be contested later in our conversation. (Indeed that discussion may be valuable in its own right.)

The term **information** is quite hard to pin down: it has different meanings in different contexts. In the field of Information Theory[2] as developed by Claude Shannon, for example, 'information' takes on a specific meaning that most people wouldn't naturally think of.

2. 'Information Theory regards information as only those symbols that are uncertain to the receiver.' – part of a definition of information at http://www.lucent.com/minds/infotheory/what1.html

In other discourses, 'information' has become synonymous with 'anything that happens inside a computer' – computing machinery is often referred to nowadays as Information Technology. While this is useful in reflecting the fact that computers now have many uses that are not strictly computational, to my mind it pollutes or dilutes the meaning of 'information', confounding it with data.

For the purposes of this discussion, I use a definition of information neatly expressed on the BBC's GCSE Bitesize website: information is *a collection of words, numbers or pictures which have meaning.* Please note: the *meaningfulness* of information is what I consider essential in this definition.

As for 'knowledge', I shall follow a definition offered by Elizabeth Orna:[3]

> *Knowledge is what we acquire from our interaction with the world; it is the results of experience organised and stored inside each individual's own mind in a way that is unique to each (though there are features common to how we all do this). It comes in two main kinds: knowledge about things, and know-how, and our knowledge is available to us at various levels from 'tacit' – what we know and use without expresssing it in words, to 'explicit' – what we can readily formulate and explain…*
>
> *Knowledge also depends on memory – and memory too comes in two kinds: internal – inside our heads, and external – knowledge transformed into information… and put into external stores like libraries or databases or reference books so that we don't have to try to carry everything we need in our heads.*

As for the relationship between knowledge and information, Orna adds:

> *Information is what human beings transform their knowledge into when they want to communicate it to other people. It is knowledge made visible or audible, in written or printed words, or in speech.*

To which I would add: diagrams, animations and other kinds of information presentation. An author transforms knowledge into information when she writes a text-book; the student transforms information back into knowledge when he reads and comprehends it.

It's easy to see how one can manage information; but what about managing knowledge? One knowledge management company admits:[4]

> *Strictly speaking, knowledge is something in your head, so you can't 'manage' it in any traditional sense. In knowledge-management circles, knowledge may refer to recorded information (perhaps optimized for effective communication) or to internalized information ('know-how' or tacit knowledge) that is very difficult or impossible to record and communicate completely. In other words, the term knowledge is used very loosely.*

However, the adoption of the phrase 'knowledge and information management' is, I accept, an improvement on merely saying 'information management' – if only because it acknowledges that much useful human knowledge is tacit, and therefore not readily accessible to others.

3. This is from *Information Strategy in Practice*, published in 2004 by Gower. I also recommend her *Making Knowledge Visible*, published by Gower in 2005.

Liz Orna also explained her view of the relationship between information and knowledge in a talk given on 1 Nov 2005 to EPSG: see www.epsg.org.uk/meetings/orna2005

4. Knowledge Management Connection: http://www.kmconnection.com/

Of particular interest is *knowledge about information* – where it is, how it is organised, how to structure it, how to find it, how to interpret it. This kind of knowledge is referred to by some as 'Information Literacy,'[5] but I would prefer to avoid the L-word. Meta-knowledge, perhaps?

Data – the stuff computers handle well

We also need to come to an understanding of the relationship between information and *data*. Much information is based on an understanding of data, and computers have long proved their value in processing data.

'Data' comes from the Latin verb *dare* 'to give', and it means 'that which is given' – the given facts from which any analysis or calculation must start. Data are often measurements of the real world. In meteorology, data consists of measurements of temperature, barometric pressure, rainfall, wind speed and distance, and each measurement is correlated with a geographical position. As it happens, almost all meteorological information can be represented numerically, and those numbers relate to the actual state of a dynamic physical system, all of which makes meteorological data eminently 'computable'.

In other applications, data is a lot more arbitrary in nature. A telephone company's records contain fields for names and addresses of customers; these are elicited in the course of business, but aren't 'measurements' as in the meteorological example above (there's no way of measuring someone to discover their name!). Nor is only the textually-represented data that is arbitrary; the phone number itself is usually a chance allocation. Other parts of the data *are* both measured and computable, such as the duration and cost of calls.

For four decades and more, computing machinery has been harnessed to work with this sort of data, managed and processed through 'databases'. A database is essentially a collection of fields, stored in an array or table. This remains essentially true even though databases have been made more complex and capable by enabling look-up relationships between data tables, as in relational databases, and by the ability to store 'BLOBs' (binary large objects), as in photographic library databases and document management systems.

In the 1960s, limitations of memory and processor speed were reasonably matched to the storage and data-processing requirements of payroll systems, customer accounts, order processing and so on, and the utility of computers to society was enabled to expand beyond their mere calculating abilities.

Computers in the library

However, there were always knowledge workers who dealt not only with data, but with something more meaningful: information.

An early encounter between computing and information management was experienced by the librarianship community. Although the information that librarians managed was stored in paper form, they had accumulated decades of experience in indexing collections of books and journals – and assigning

5. I personally find that the way the term 'Information Literacy' is used is imbued with the librarian's bias towards the skills of finding and consuming information. Should not a 'literacy' also encompass the skills of creating and organising information?

See my critique in *New kinds of literacy, and the world of visual information*, a PDF downloadable from http://www.ideography.co.uk/infodesign/eigvil/presentations/literacies.pdf

them to categories, for example the categories of the Dewey Decimal System. This practice had intellectually prepared librarians to harness the power of databases to build library management systems.

3 Computers and information products

1970s: computers become tools for information production

The term **information product** is a useful one of recent coinage. It brackets together in a convenient way all the print-on-paper information products such as books and journals and documents, together with those that are conveyed by electronic media such as email lists, Web sites and intranets. Even radio and television broadcasts count as 'information products' when they are used to convey information to people rather than to entertain.

The term also usefully reminds us that information has to be produced by someone, in a process of authoring, editing, design and production.[6] For centuries, the repository of these skills has been the publishing industry, and the writers who provide input to it.

It was only in the 1970s that a set of evolving characteristics of computers combined to make them useful to authors and publishers of information products. Those characteristics were:

- the eight-bit byte, and standardised text encodings (e.g. ASCII)
- the visual display unit, displaying computer-generated text characters
- keyboard input of text
- convenient read-write magnetic storage capable of holding useful quantities of text, such as the floppy disk
- methods of printing texts, e.g. to a Selectric typewriter, daisy-wheel printer or dot-matrix printer for word-processing; or to a photographic typesetting machine for publishing-quality output.

6. Or is this a moot point?

For example, if a scientific visualisation system converts 20,000 oceanographic measurements into a graphic representation from which a scientific worker can derive knowledge, did the visualisation software produce information?

When the word went electric

In the 1970s, two kinds of publishing activity started to exploit the ability of computers to handle text: **word-processing** and **typesetting**. Initially the two kinds of equipment were quite distinct, and not even inter-operable.

An early precursor of word processing systems was the IBM Magnetic Tape Selectric Typewriter (1964), and a later version which used magnetic cards. However, these lacked an electronic visual display. In 1973, the Vydec Word Processing System made its debut as the first machine with both a VDU and floppy discs for storage of text, followed by products from Wang, such as the Wang 2200 system. These machines were dedicated word-processors: they were not equipped with software to perform any other task.

The first word-processing system to run on a general-purpose PC was Michael Shrayer's *Electric Pencil* (1976), followed in 1979 by *WordStar*. In the early eighties more PC word-processors emerged: Multimate, DisplayWrite, PC-write, Xywrite, etc. In Britain, a seminal product that

encouraged many authors to retire their typewriters was the affordable Amstrad PCW 8256, running CP/M on a 4-MHz Zilog processor, with one floppy disk drive (the later 8512 model had two). LocoScript word-processing software was supplied with the machine.

Typesetting goes computerised

Mechanised typesetting until the 1970s was usually done by casting metal type from molten lead alloy in a casting machine – such as the Monotype caster, driven by punched paper tape that had been produced on a special compositors' keyboarding machine.

In 1944, French inventors René A. Higonnet and Louis Moyroud filed patents for a method of setting type by flashing a strobe light through a spinning disk carrying type characters in photographic negative form, then through focussing lenses onto bromide paper. The first 'Photon-Lumitype' machine was installed in 1954 at the *Patriot Ledger*, a Massachusetts local newspaper. These machines were not yet computerised: like the Monotype caster, they were driven by punched paper 'idiot tapes'.

Typesetting began to be computerised in the 1960s, on mainframe machines such as the RCA VideoComp and the IBM 2680, then on minicomputers at the start of the 1970s. It wasn't easy or cheap to achieve publishing-quality justification and pagination at speeds acceptable to the newspaper industry – the key customer for this innovation. Clark E. Coffee, whose company Graphion developed such a system in 1970, recalls that their computer cost $180,000 – and he had just bought a house in San Francisco for $25,000![7]

7. This and other reminiscences of early typesetting by Clark Coffee can be found at http://www.geocities.com/danhaag/oldtype.tpl.html

But it didn't take long before a wide range of computerised phototypesetting systems became available at a cost that newspapers, printers, even jobbing typesetters could afford. For example, consider the systems developed by Compugraphic: the CompuWriter (1971), the first photosetter to dispense with paper tape input; the VideoSetter (1973), which generated type images digitally on a CRT rather than using a glass negative; the Unified Composer (1974), which brought monitor and floppy disk drive together for the first time in a typesetting machine. These were followed by the highly successful EditWriter (1977) and MCS (1981) systems.[8]

8. Information on the Compugraphic timeline from http://agfamonotype.com/about/timeline.asp?show=agfa

Compugraphic was anything but alone in its pursuit of this profitable market for computerised phototypesetting equipment. Mergenthaler-Linotype, Monotype, Hell, Itek, Berthold, Scangraphic, Harris Intertype, Miles 33 and AM VariTyper are other names that come to mind.

So, at the start of the 1980s, there were dozens of proprietary text-handling computer systems – both word-processors and typesetting machines – using non-standard text encoding schemes, wildly different methods of mark-up, even different ways of writing a floppy disk or tape.

It was not untypical for an author to work on a wordprocessor, print the text out on paper, and send it to the publishers who would then need to have it re-keyboarded by the typesetting equipment operator – all for the lack of a reliable way to move text from one machine to another electronically.

In retrospect, this chaos focussed the minds of thoughtful publishers, and in time led to the invention of generic mark-up – a key turning-point in giving computers a better 'handle' on textual information.

Text: rich in information – poor in structure

Textual information is easy for people to understand, but it has long been 'computable' only in the crude sense of calculating its line-breaks and page-breaks for display and printing purposes. The meaningful structure of text is not the kind of structure that a machine can get a 'handle' on unless 'handles' are added – by handles, I mean added entities, such as the field-boundaries of a database, which allow machines to identify and process their contents.

However, the requirements of word-processing and typesetting introduced a new element: **mark-up**. People wanted to centre blocks of text, embolden words, force page breaks and so on; this was achieved by embedding special encoded control-sequences within the text stream. The control mark-up vocabulary could be as simple as the 'dot commands' of WordStar, or as fully-featured as TEX, the typesetting language invented 1977–79 by Donald Knuth of Stanford University.

Most of these mark-up languages could be described as **procedural** – they specify the procedure that the computer is to perform, such as changing the font size or moving onward to a tab position. Procedural mark-up is great for typesetting, but it doesn't do anything for a computer's ability to grapple with the text content in a meaningful way. For that to improve, we need to use **logical mark-up** that denotes the *function* of the text content elements. One example of logical mark-up is the LATEX system developed on top of TEX by Leslie Lamport in 1983, where the author of a publication may write:

```
\section{Section Title}
```

…and the text is later passed through processing software that applies a predefined format to that section title, and auto-numbers it if required.

However, a more significant development was already under way within the publishing industry, and that was the related idea of **generic coding**, which led us to SGML, HTML and XML.

GenCode: a road out of chaos

In the late 1960s, the Graphic Communications Association in the USA was struggling to solve a major technical problem then faced by book publishers – especially publishers of reference and educational books. Because each vendor of typesetting equipment had its own proprietary markup language, and the codes for these were often revised radically for each new generation of equipment, publishers who had archived typesetter files to make the production of revised editions easier found they were 'locked in' to that vendor's typesetting language – or even worse, that the equipment to run the files was no longer available, and the tapes were useless.

A New York book designer called Stanley Rice suggested that publishers should collaborate to devise their own shared mark-up language, which

would be stable for years to come, and which could be translated into a particular vendor's typesetting codes for production purposes.

What Rice proposed was a markup language that would declare in great detail the editorial function of parts of the text such as chapter, heading, quotation and so on: generic coding. The idea was that publishers would use markup only to define *content* and *structure*; the designed *appearance* of the book would result from the set of transformation and formatting rules chosen by the book designer, at the time when when the marked-up text would be converted into typesetting machine codes.

The GCA set up a project group to study the idea; this evolved into a body called the GenCode Committee. The more they thought about it, the more they realised that a single universal catalogue of editorial elements was not possible; different catalogues of codes would be needed for different kinds of books, depending on their subject. A dictionary, for example, would need special codes to show that a string of text is a pronunciation guide in the International Phonetic Alphabet.

Laying down the laws of markup

In 1969, a young computer-savvy Boston lawyer called Charles Goldfarb was hired by IBM and put in charge of a project to develop an integrated suite of office automation software for law practices. Law is a very text-based application, and the goal was to store these texts in a format that not only supported word-processing, formatting and printing, but also that meant that the texts could be searched and retrieved intelligently.

Goldfarb based his work on the GenCode concepts. He reasoned that if markup could identify a block of text as the name of a plaintiff in a law case, you could search your text database for the string 'Brown' as the name of a defendant – rather than as the colour of the getaway car! With colleagues Ed Mosher and Raymond Lorie, Goldfarb devised **GML** – the Generalized Markup Language.

GML was an improvement on the GenCode concept in two important ways. Firstly, it invented the idea of an external reference document – a kind of 'rule book' – that would list all the elements allowed in a particular class of documents. Secondly, GML allowed designers of these document classes to specify how the elements should be used and structured, both in sequence and hierarchically. For example, the 'rule book' for a report might say:

- a **chapter** *must* have a **title** followed by *at least one* **section**…
- a **section** *must* contain a **heading** followed by a mixture in any order of **paragraphs**, **lists**, **quotations** and other **section**s…

This idea of a 'rule book' for a class of documents is what later developed in SGML and XML into the **DTD** or **Document Type Definition**. It neatly solves the problem that had bothered GenCode, of not being able in advance to imagine all the tags that would ever be needed in documents. Quite simply, if you discovered a need for a new kind of document, you'd figure out the list of elements it would contain and write an appropriate 'rule book'.

The flexibility of GML was such that it was enthusiastically adopted by the technical publishing division of IBM, which by the 1980s was creating huge quantities (over 90%) of its technical documentation using GML-based publishing software, running on mainframe computers.

Making meaningful elements

What GML shared with the GCA's GenCode project was the concept of creating meaningful elements within a text file through the use of mark-up. For example, in creating a bibliography you could use this mark-up to define an author's name:

```
<name>
<forename>Anatole</forename> <surname>France</surname>
</name>
```

This makes the structure explicit. And indeed, though you could easily make a tag vocabulary that controls the typeset appearance directly, proponents of generic mark-up insist that it's better practice to encode *semantics* (meaning), rather than the appearance of the document.

This is where mark-up becomes an important information management technology. With appropriately configured searching software, it is now possible to search the text database for the search-string 'France', setting the condition that this *must* be found within the `<name>` element (or the `<surname>` element if you want to be even more specific). And that avoids you being deluged with unwanted search results that include 'A History of France' or 'Published in Marseilles, France'.

When it is time to convert the marked-up text into a printed book, we might process the file to make all authors' names stand out on a line by themselves, in 11pt bold type, with 6pts above and 2pts below, and with the forename in italic. With suitable software we could event transform the order of data, so the name would be presented as '**France, *Anatole***' instead.

How SGML became Standard and Generalized

In 1978 the US standards body ANSI formed the *Computer Languages for the Processing of Text* Committee. It recruited Goldfarb as a member and asked him to develop a computer-readable text description language, based on GML. The GCA supported the ANSI project, and in 1983 recommended that the sixth working draft of the new Standard Generalized Markup Language (SGML) be adopted as a publishing industry strandard.

Early adopters of publishing systems based on SGML markup included the US Internal Revenue Service and the US Department of Defense. Indeed, when the DoD insisted that bids to supply weapons systems to the US armed forces would be considered *only* if the defence contractor could deliver all documentation marked up in SGML (the CALS initiative), it caused a revolution in the technical documentation industry. Companies like Boeing and Lockheed-Martin scrambled to comply, and vendors came to the rescue with SGML-compliant document composition system such as ArborText Adept•Editor, SoftQuad Author/Editor and Grif.

Beginning in 1984, development of SGML also became an international endeavour, through an ISO working group – JTC1/SC18/WG8. The SGML standard was published in 1986, as ISO 8879:1986.

Metalanguages and applications

As is often pointed out, SGML and its successor XML are not simply mark-up languages but 'metalanguages' – sets of rules for how you can go about creating your own purpose-built mark-up languages. In fact, hammering out what kind of elements you need in your SGML-type mark-up language can take a lot of time and trouble, organisational politics (and sometimes bad temper!) until everyone agrees, and the decisions can be recorded formally in a Document Type Definition for that class of document.

Some examples of important industry DTDs are **DocBook**, which can be used to mark up the text for books and technical manuals, and **NewsML**, which is used to share newspaper stories between papers and press agencies, together with the metadata about who wrote each story, when and where, and who holds the copyright. DocBook and NewsML are examples of what are called **SGML applications**.

'Sounds Good, Maybe Later?'

The main drawback of the SGML system was long held to be its complexity, caused in part by the 'liberal' nature of the metalanguage specification. For example, though the use of `<` and `>` as 'delimiters' for markup was recommended as a *norm*, it was left 'legal' for implementers to choose any other characters as delimiters, if it suited them.

Also, when Working Group 8 was arguing about the rules of SGML, some members thought that many documents would be coded by hand in a text editor. To reduce the amount of tag-typing, these designers of SGML allowed a practice called *tag minimization*. For example, you might specify in your DTD that it was OK to leave out the closing tag for an element, if it was obvious from the context that the end of that element could be *inferred*.

(HTML from its very beginning included tag minimization rules. For example, each **list item** `<LI>` begins with a start-tag, but the end-tag is usually omitted.)

The wide range of variability that the SGML specification allows may have been put in place with the best of intentions, but in practice it makes life hell for the poor programmers who have to devise software for parsing and processing the documents. This kept SGML software expensive.

Combined with the cultural revolution in authoring practice that is required to work with SGML, the expense and complexity meant that for many companies, SGML seemed to stand for 'Sounds Good… Maybe Later'.

XML – a more practical solution

One inspiration for XML may have been a book called *SGML for the Web*, by Canadian SGML guru Yuri Rubinsky, a lively and humorous advocate for SGML. His company SoftQuad developed an impressive browser called Panorama that allowed SGML files to be read straight off Web sites and manipulated in a number of useful ways – for example, by choosing among a variety of alternative views of the data. Yuri died of a heart attack in January 1996, and *SGML for the Web* was published posthumously.

Yuri had argued that the tag vocabulary of HTML was too limiting to do any serious organisation of text with it. On the other hand, the major developers of commercial browsers, specifically Microsoft and Netscape, made it clear that they did not *ever* intend to support SGML on the Web – taming SGML's complexity and open-endedness would be too much of an effort, they said.

The World Wide Web Consortium (W3C), an organisation set up to develop standards and recommendations for use on the Web,[9] invited a number of SGML experts to think about the gap between HTML and SGML. The team, led by Jon Bosak of Sun Microsystems, realised that if they could get rid of tag minimisation and radically reduce the variations of practice that SGML permitted, they could produce a simpler standard that would be easier for software to 'parse' a document (i.e. to make sense of its structure) and to 'validate' it (check to see that the tags have been applied according to the rules defined in the DTD.)

9. It may seem strange that I have not related the history of how Tim Berners-Lee and colleagues started the World Wide Web. I'm making the assumption that the story is well enough known, and experience of the Web sufficiently ubiquitous, for me to 'take it as read'.

To give you an idea of how radically the team pruned the thorny bushes of SGML – the draft of the XML specification that was written by Tim Bray and CM Sperberg-McQueen was only 26 pages in length, as opposed to the 500+ pages of the SGML specification; and it took only a few months to do.

Compared to SGML, the XML specification is a lot more restrictive, but in practice these restrictions have no impact on functionality. For example:

- The only permitted 'delimiters' for tag boundaries in XML are < and >.
- Every container element must have a start-tag *and* an end-tag. Tags may never be omitted for 'minimization' purposes.
- As for tags that are 'empty' (like as the IMG tag in HTML, which is self-contained), they are to be explicitly marked up as such with a special end delimiter `/>` – thus, `<img src="foto.jpg" />`.
- Attributes inside start-tags, like the `src="foto.jpg"` example above, must always have their value contained within double quote marks. (In HTML, sometimes they were, and sometimes they weren't.)

As it has been refined, XML has also been able to take advantage of the standards that grew up around HTML to support the development of Web services – such as HTTP for file retrieval, and Cascading Style Sheets for the association of styling characteristics with document elements. It has also spawned related standards such as the XML Styling and Transformation Language (XSLT).

4 Structured information – an elusive goal?

Document styling/information structure

As we have seen, computers were adopted by knowledge workers as tools for the production of information products. Word processing and typesetting were the first steps, followed in 1985 by 'what you see is what you get' style page make-up programs ('desktop publishing').

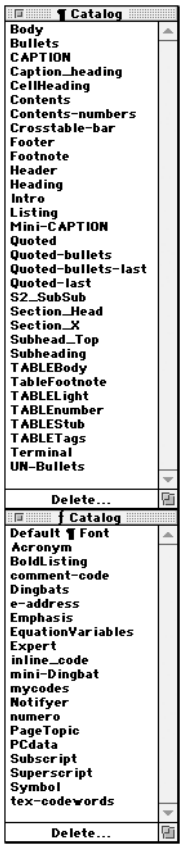


Fig. 1 – Paragraph and Character stylesheets for a FrameMaker document.

It is interesting to recall that Version 1 of PageMaker, the software which can be said to have ushered in the DTP revolution, did not offer the user a way of applying the same typographic styles again and again to blocks of text with a similar function (for example, to ensure that all subheadings were set in bold Helvetica 11pt type centred). It was not until version 2.0 that PageMaker offered such a **stylesheet** function.

Stylesheets are interesting, because they are a kind of halfway house between procedural mark-up and generic mark-up. As an example of this, see Fig. 1 (left), which shows the Paragraph and Character stylesheets that I developed for a document that I wrote – using FrameMaker software – on the problems of typesetting African languages. (Character styles are used to override the default typesetting style of the surrounding paragraph, for a few characters only – as **here**.) Note the following:

- Most of the styles do relate to the logical function of the text which is marked up by them. Good examples are Subheading, Intro and Listing (in the Paragraph stylesheet); and Acronym and tex-codewords (in the Character stylesheet).
- However, there are some styles which were created simply to give the means to tweak the appearance of texts in particular contexts. For example there is the Subhead-Top paragraph style which forces a page break and adds some space above; and Quoted-bullets-last is a variant of the Quoted-bullets style that adds a bit more space after the last bullet in a list.
- It's also significant to note the document elements that *aren't* there. For example in XML or HTML document composition, bulleted 'list-item' elements in a list would be contained in a List element. Here, there are no elements higher than a paragraph level.

Despite these possible shortcomings, document-composition stylesheets are for publishers an attractive way to get *some* structural mark-up added to information products in the process of their creation. Authors and designers are well motivated to use the stylesheets, because they give them the control over appearance which they value. In contrast, many authors resent having to mark up texts for purely structural reasons, if they don't see any benefit to their own work with the document!

Indeed, FrameMaker is a good example because of the relative ease with which its stylesheet mark-up can be translated during HTML export into sensible HTML mark-up via a mapping table, or into XML mark-up using

the existing names of the paragraph and character styles, or into bookmarks with hypertext linking when a Portable Document Format (PDF) file is produced from the publication. For those whose ambitions to produce structured data are more far-reaching, FrameMaker will work with an 'Element Definition Document' that combines SGML/XML structure rules (i.e., a DTD) with a typographic stylesheet.

The waywardness of Word and Web authors

However, information products are not always so well-endowed with a logical, machine-readable structure. I wish I could have a five pound note for each Word document I have been sent which has been typographically styled to present several levels of heading… yet where every paragraph is styled as 'Normal' (with local format overrides). Including a great many such Word documents authored by BCS 'computing professionals'!

The World Wide Web is notorious for the way in which Web designers have used any old HTML tag in defiance of its logical purpose to achieve a desired 'look'. For example, the main heading on a page might be set as <h2> because the designer thinks <h1> looks too big; or <blockquote> is used to cause a block of text to have a wider margin, regardless of whether it truly is a quote or not. There is less excuse for this now that these display attributes can be set using Cascading Style Sheets, but such practices persist.

I remember that one of my earliest experiences of a Web search engine was one run by the OpenText company of Canada, as a demo of their information management software's capabilities. It allowed you to search for text *in the context of a particular HTML tag* such as <h1>. But this proved to be less useful than I thought it would be, because of the waywardness of Web page mark-up in practice. However, more recently I have found it useful, when composing Google searches, to focus my searches with the allintitle: specifier… essentially, a less ambitious variant of the same practice.

'Data-based' information creation

One way to overcome the reluctance of people to apply logical mark-up to the information products they are authoring is to force them to author it in the interface of a database program, or a Web form that has a database behind it. For example, at Harper-Collins, their bilingual dictionaries are 'authored' by teams of writers, who input the dictionary entries into a form interface to a database program. (This also solves the problem of how to get many people to work on a book at the same time.)

To produce a dictionary, the database is made to unpack the fielded data into a text file with SGML mark-up. This SGML file is then imported into a FrameMaker 'structured-document' template, which automatically typesets the dictionary and generates page running-heads, etc.

Similarly, good success has been achieved in the related fields of directory and catalogue publishing, and timetable production, by using a database as an information receptacle and organiser, the database's 'report generator' as a means of producing marked up text, and publishing software to convert

the mark-up to something typographically friendly to the end user. And as an alternative to print production, it is increasingly common to design the transformation process, and use of mark-up, to display the information on demand, as a dynamically generated Web page.

Projects of this sort depend for their success on the combined efforts of IT personnel and information designers (see more about the contribution of information designers on page 39).

5 Grappling with unstructured information

Information Retrieval and the free text search

So, an increasing volume of knowledge has been transformed into information products, and an increasing volume of data and information products is stored in electronic form. (I include here offline resources such as CDs, local online resources such as on hard disk, and networked resources.) Some of this information is structured through database fields or mark-up, but much is not structured at all, or is structured unreliably.

Therefore, all too often, the knowledge worker is thrown back into reliance on a **free text search**: a search for strings such as words or phrases within the information base that the searcher thinks will turn up relevant material. A search on Google is the typical example of this activity, and the frequency with which people rely on free text searching has mushroomed together with the growth of the World Wide Web.

The inverted tree

To search a large quantity of text from one end of the file system to the other is too slow and inefficient. This problem is usually solved by building an 'inverted tree index', which records (by means of file pointers and offsets) each location of each of the many instances of each word in the collection.

The index has to be compiled and maintained, and depending on thoroughness can take up to several times file storage again as the text collection being indexed. But it is helpful in speeding access to information, because the index resource is queried first, which swiftly finds the indexed word or words, and then follows the pointers to pull up the candidate documents.

Inverted-tree indexes are now commonplace in corporate and Internet search systems, and have even come to the desktop. For example, Macintosh users running Mac OS X 10.3 can configure the 'Sherlock' search engine to use spare cycles to maintain an index of all the text content *within* files on a system, and this capability has been upgraded further in Mac OS X 10.4.

Making free text searches more productive

Frankly, most information users have low levels of skill when constructing searches, and the problem can be compounded by inconsistent rules and interfaces between different search engines. (Will searching for villa **AND** Spain retrieve pages that contain both these words, or lots of pages about

Spain that make no mention of villas, or vice-versa?) These are some of the problems in information retrieval that have given rise to discussions of the skills of 'Information Literacy'.

On the 'provision' or IT side, there is a great deal to be studied about how free text searching can be made more productive for users. Here are some of the approaches which have been tried:

- **Phrasal searching** – searches that require the requested words to appear in the retrieved documents as a phrase, for example a search for "house prices".
- **Word-stemming** – this expands the search by additionally using, as search-terms, words that are etymologically related, so that for example a search on the term medicine might turn up references to medical or medication.[10]
- **'Thesauri'** – this refers to what should more strictly called *synonym lists*. A good example of this would be if a search for "automobile trunk" were to come up with documents that mention a "car boot". It does raise the question of who develops these synonym lists, and if there might be automated ways of building them by observing the repeated attempts of users, or by examining historical records of the search terms that had been used to retrieve each document in the collection.[11]

10. The Google search engine now employs word-stemming as a matter of course (2006).

11. Recognition of the fact that different people might use different terms in their search for information is not a new thing. Indeed, one of the traditional skills of the back-of-the-book indexer is to mediate between the vocabulary of the author and that of the reader, building in synonyms to increase the reader's chances of finding information.

Relevance ranking

A related set of techniques are those which take the retrieved documents and present them to the enquirer in what is deemed to be an appropriate order of relevance. This is known as relevance ranking, and once again a variety of strategies are employed:

- **Frequency of use** – This is based on the assumption that if a search term is used many, many times in a document, it is more likely to satisfy the enquirer's needs than if the terms are used just once or twice. In practice, this strategy alone often produces distorted results.
- **Citation frequency** – This simply regards documents that a lot of people make links to as being more likely to be 'good documents'.
- **Location or context** – Depending on how the document is structured, this may be a productive method of relevance ranking. Greater weight may be given to the appearance of a search term if it appears early in the document rather than later. For Web pages, the appearance of a search term within the tags for an <H1> or <H2> element is a sign of its importance to the document as a whole. (Users of Google can enforce a form of locational filtering by insisting that the search term must appear in the title of a page, e.g. allintitle: "Richard III".)
- **Hypertext targeting** – This is a rather sophisticated technique which examines the corpus of documents to see if the search terms are used within a document from which a link is made to another document that also has those terms in it, and then promotes the *target* document

up the relevance ranking. The assumption is that the first document is citing the second one as an authority.[12]

Having excellent algorithms for inverted-tree indexing, free text searching and relevance ranking are part of the weaponry of competitive advantage between Web search sites, and so there are doubtless many other techniques in use which are not fully publicised. Certainly there is much scope here for discussion.

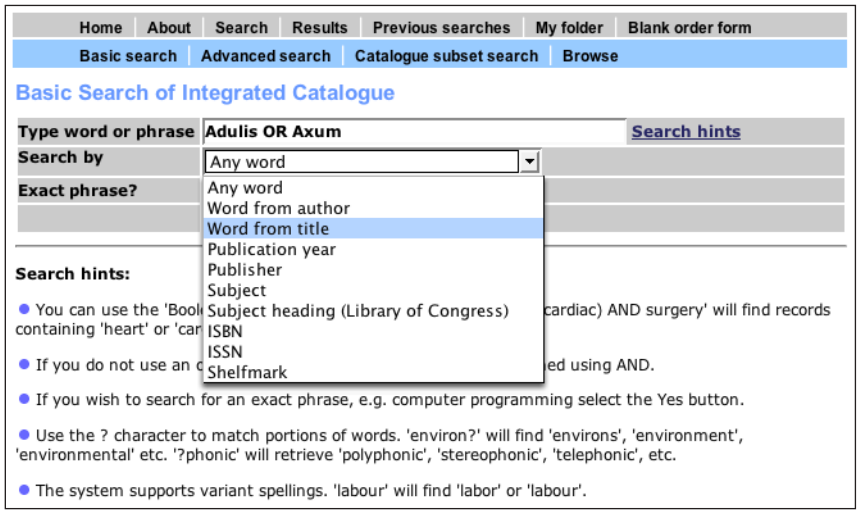
12. But this also can give skewed results. A famous case was of an anti-Semitic Web site which Google placed first in any search for the word **Jew**. That was a result of the anti-Semitic site being often linked to from other anti-Semitic and race-hate sites.

6 Introducing Metadata

What is metadata?

Metadata is 'data about data', but since I have been trying to use words a bit more strictly than is often the case, perhaps I should say that what is usually meant by 'metadata' is well-structured data *about an information product* (or an information resource) – which may or may not itself be structured.

A good example of this is the British Library Integrated Catalogue, which can be found at http://catalogue.bl.uk/. A section of the search interface is shown in Figure 2 below:



In this case, the information resource is in the form of printed books and journals. The metadata about these is in the form of records in a database program, with fields for author, title, year of publication, publisher, subject, ISBN and so on. Of course, when I perform the search for "Adulis OR Axum" as shown above, the search will not be within the information resource itself (the books have not been digitised!), but within the metadata record.

In this case, the purpose of the metadata is to help find an information resource that is *not* stored in computer memory, but even if the resource *is* held electronically, there can be good reasons to generate metadata records for it. I often use a photo library as a teaching example. Here, the resource is non-verbal, and a computer cannot be expected to figure out what the significance of the image is. Let's examine this example more closely.

Metadata in a digital photographic library

I manage my own collection of digital photographs with iView MediaPro. This program was developed especially for managing collections of digital assets such as photographs and drawings, audio and video files and fonts.

To add digital photos to an MediaPro database, I use the Import command – a bit of a misnomer, since what it really does is make a thumbnail image of each photo, record its location in the file system (which can be offline on a CD, or on the hard disk, or network), and automatically capture certain bits of data about the image, such as its filename, file type, dimensions and so on.

To make the collection usefully searchable, I have to add my own metadata. MediaPro makes this easier by providing slots for the names of people in the pictures, keywords, and categories, as shown below:

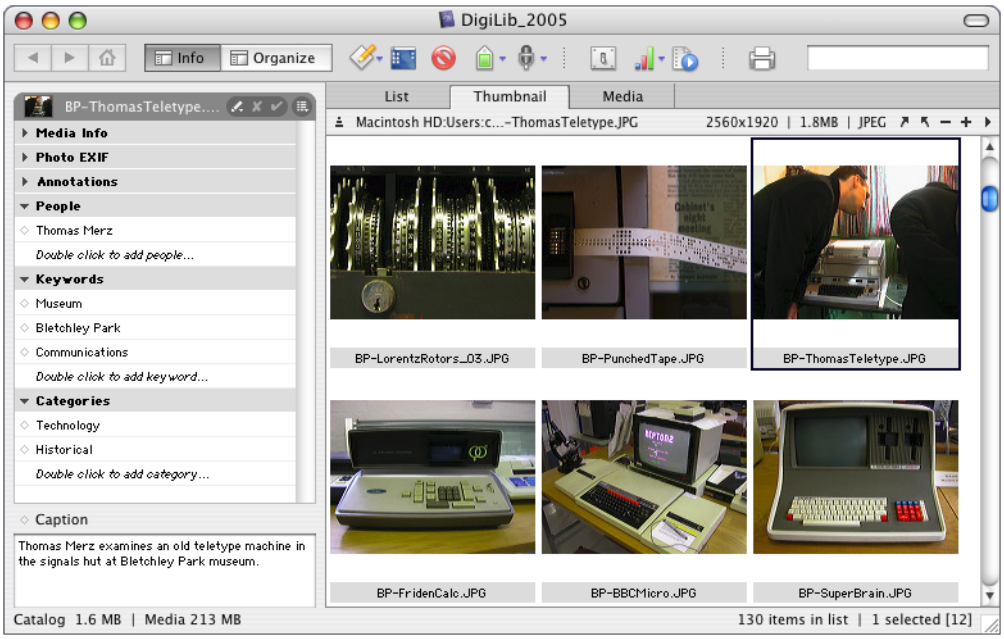


Fig. 3

An iView MediaPro image database, opened to show the **Info** panel where metadata can be added. The photo that has been selected is the one top right. Note the expanded fields for information about People, Keywords and Categories.

Also note (bottom left) that provision is made for a caption.

You will notice that under the Keyword section there is a list of the keywords that I have already applied to the image selected, and a line that says 'Double click to add keywords'. One can either type a keyword, or choose one that has been used already, from a pop-up list. It obviously makes sense to use a controlled selection of keywords – more on that in a moment.

In MediaPro, it has to be said that there is nothing essentially different between 'keywords' and 'categories'.

Metadata issues illustrated by this example

- **One-to-one metadata.** A lot of metadata in a MediaPro database – as in other metadata databases like the British Library catalogue – pairs a named field with a single possible item of content. A MediaPro example would be the Filesize field under the Media Info tab (which is closed in the figure above). A library catalogue example would be the book's title.

- **'Multiple occupancy'.** The Keyword field is different, because several keywords can be assigned to a single image, and this is a very useful thing to be able to do. Similarly, one can assign an image to several different categories. Please note that this is a tremendous advantage over old-style taxonomic/cataloguing systems like the Dewey Decimal System, in which an item had to be assigned to only one category.[13]

- **Sensible taxonomies.** When you set out to sort any collection of information resources into categories, you have to devise categories (taxa) that make sense, in the context of the way people will want to search for and use those information resources. A negative example – Foyle's Bookshop in Charing Cross Road used to shelve its computer books in sections based on the publisher; and within those sections, in order of the author's surname. Doubtless this was of great utility to the staff in checking stock, but if you want to learn how to write Perl scripts you want to find all the Perl-related books on one shelf, no?

- **Controlled vocabularies/category lists.** MediaPro is lovely software for the individual user – beautifully flexible – but this flexibility is a weakness that makes it a rather poor tool for organisational use. The problem is that you can make up any new keyword or category at any time.

  In contrast, Extensis Portfolio is image database software that has several levels of access – Administrator, Cataloguer and Enquirer, in effect. The Administrator can define a set of keywords and categories and 'lock them down', so that a Cataloguer can choose which ones to apply to a newly-acquired image, but can't invent any new ones. This not only maintains the 'sensibleness' of the taxonomies but also means that the list of search terms can be made known to the Enquirer, making searches more productive.

(I can't leave the subject of controlled vocabularies without mentioning an argument I have been having with the *Information Design Journal*, whom I assist editorially. The publisher encourages authors to submit keywords with their articles, but provides no 'pick-list' of keywords, so authors invent their own. To me, this seems like an entirely useless activity.)

13. I think of this as the 'rice vinegar problem'. You think the supermarket doesn't stock it, because you went to the section where the vinegars are on display and it wasn't there. How were you to know that you should have looked in the Japanese Foods section?

Mechanically harvesting metadata

Some data about digital images already resides in the header blocks of file formats, and in the computer file system. It is by accessing these records that MediaPro captures details of the size of files, pixel dimensions, creation date, archiving date and so on.

Additionally, some file formats such as JPEG and TIFF allow other kinds of data to be embedded in them. The Japan Electronic Industry Development Association (JEIDA) exploited this by devising a metadata standard called **EXIF** (Exchangeable Image File Format), and all digital cameras use EXIF to record information about the way in which the picture was taken. Because this metadata is stored in a standardised and documented format, iView was able to develop MediaPro to harvest this metadata automatically when the camera files are taken into the database. The figure below shows MediaPro displaying this metadata for one of the photos in my collection.

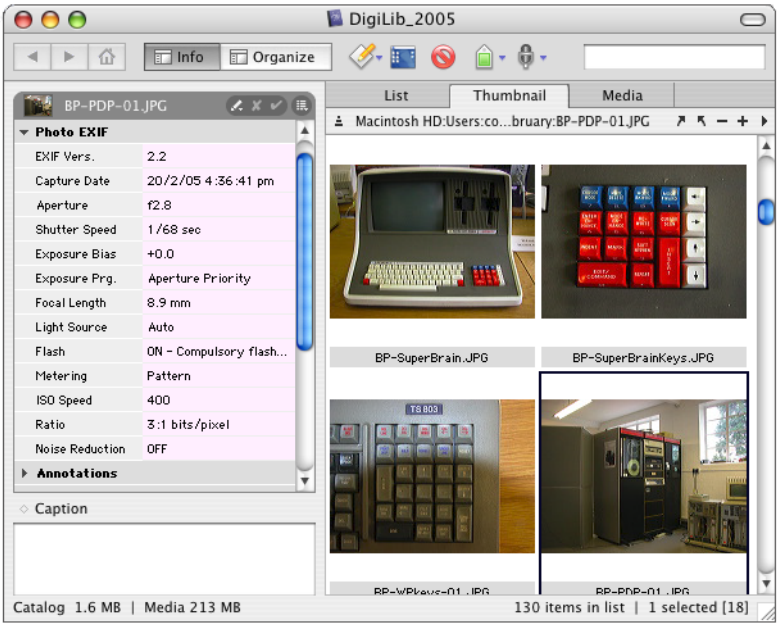


Fig. 4

MediaPro displays the EXIF data that has been gathered from a digital photo imported into the database. This includes the date and time when the photo was taken, the aperture and the shutter speed, and various camera settings.

In addition, the camera uses EXIF to embed an ICC profile in each image, which defines the colour-imaging characteristics of the camera. This data makes it easier for photo-processing laboratories to make good-quality prints from the digital images. Another benefit of standardised schemes for metadata interchange!

Lessons for the application of metadata to the wider world of information products

I chose the example of a photographic database because the subject is close to my heart, but now it's time to see what this example can and can't tell us about the management of other kinds of information/knowledge resource.

Non-textual media: In one way, it's been an excellent example, in that it reminds us that information can be conveyed to us in many kinds of media product, not only in text form. For example, the BBC gives access to previous radio broadcasts through its 'documentary archive',. There is simply no way that a computer can be expected to determine the relevance to a search of the spoken word without some kind of metadata being attached to it.

Controlling categories and vocabularies: Within a tightly-focused domain such as a photographic library, it is understandable how one might develop a sensible set of metadata elements ('file type' – 'photographer' – 'categories' –

'date' – 'location' – 'people') – and also, for some of these elements, a set of permitted labels. It is more difficult to see how this can be generalised into a single global catalogue of metadata elements that could be used uniformly for every kind of digital information resource.

This reprises the old problem that the GenCode committee faced back in the sixties (see page 8) of being able to guarantee an element set that would deliver both *specificity* (the ability to apply markup usefully to each type of publication) and *interoperability* (the ability to be processed in a uniform manner, across limitations of machine type, software, time etc.) As we have seen, SGML and XML solved that problem by offering a uniform manner in which tailored tag vocabularies could be developed and applied to a wide variety of specific 'document types'.

And as I shall explain, this quandary is currently being addressed by some important metadata projects: the *Dublin Core Metadata Initiative*, and the development of the *Resource Description Framework* (RDF) standard, which in turn is linked to a grand objective known as 'the *Semantic Web*'. An elaboration of RDF is the use of *ontologies*, while others are seeking to improve the usefulness of indexes through the use of *topic maps*.

Getting things categorised

As I alluded to on page 13, the creators of information products tend to shirk structuring information in a way that a machine would find helpful, and it takes a big cultural shift within an organisation to make this happen. Exactly the same is true when it comes to getting creators to add structured metadata to their otherwise unstructured products, even though the burden is less.

A partial solution is to include a step which reminds authors of the task; for example, when saving a file in Microsoft Word an author might be prompted with a form asking for keywords, etc. Where authors work within the framework of a Document Management System, as is often the case in technical authoring groups, the system may be more stringent, and refuse to save and store the document unless the required metadata has been supplied.

Auto-categorisation?

While some people urge that we *should* try to structure as much information as possible and classify and organise it, others argue that the application of artificial intelligence can render this unnecessary. Their argument goes that software can 'understand' what documents are about, infer their association with other documents, and bring them to the attention of information users, either in response to a search, or pro-actively.

An eminent example of this kind of system is that supplied by **Autonomy**, founded in 1996 as a spin-off from Neurodynamics, a Cambridge (UK) company which conducted research into adaptive pattern recognition. At the core of Autonomy's application suite is what they call IDOL, the *Intelligent Data Operating Layer*, and DRE, the *Dynamic Reasoning Engine*.

Part of Autonomy's process for making sense of documents is an application of Charles Shannon's Information Theory; as a report on Autonomy written

by the Butler Group explains, this works with the assumption that the most significant words and phrases used in a document are those which are used with *low* frequency. The system also applies Bayesian Inference to figure the probability that a document is relevant to a search being conducted against a document repository; and by observing how users accept or reject the matches offered, the system is able to learn. The Butler report comments:

> *[This] approach to integration… does not rely upon additional technologies. By forming an understanding of the information embedded in data, Autonomy can integrate data sources based on this understanding. This eliminates the need for metadata, middleware, business rules, tagging, or any other constructs to bring data structures together.*

Large claims are made for Autonomy's success in making sense of masses of unstructured data. As a result of their acquisition of SoftSound, Autonomy is even able to extract data from radio broadcasts, or call centre recordings of telephone calls (or wiretaps, if you are feeling paranoid!).

Many large companies have adopted Autonomy's system, including the BBC, Associated Press, Reuters, McMillan, British Aerospace, Ove Arup and Ford. The Ford case is interesting in the context of Knowledge Management, as it is at the heart of a 'learning network' for ongoing education of the Ford workforce, tapping into repositories of information produced throughout the organisation.

It would be useful to discuss the usefulness or otherwise of this approach.

7 Standardising resource discovery metadata

What is 'resource discovery metadata'?

This term refers to the kind of metadata that is attached to digital objects, such as documents, Web pages or images, to enable them to be found more effectively in an electronic search process. The metadata could be embedded inside the digital objects themselves – this is easy to do with HTML and XML documents – or it could be external to the objects, and associated with them in some way (for example, a database of abstracts of articles in scholarly or technical journals).

Where the digital objects are carriers of information, we may call them information objects. And it is important to point out that an 'object' in this case is 'anything that has an identity', anything which can be pointed to, or in a Web context, anything which has a URI (Uniform Resource Identifier).

Please note that a URL (Uniform Resource Locator), or Internet document address, is a rather limited form of URI; where information is more explicitly structured, for example an XML document, subsections of a document can be more formally endowed with their own explicit identity and URI, and of course can have their own resource discovery metadata, too (for example, each entry in a bibliography could have metadata to record when that entry was made, and by whom).

Dublin Core

The **Dublin Core Metadata Initiative** is a metadata standards development project which started in 1995 with a workshop in Dublin, Ohio. The aim of the project was to make it easier to find information resources using the Internet, through three kinds of activity:

- developing metadata standards for resource search and retrieval across different subject areas;
- defining frameworks for the interoperability of metadata sets;
- facilitating the development of community- or subject-specific metadata sets that work within these frameworks.

Version 1.1 of the Dublin Core Metadata Element Set provides fifteen elements as standard. These are defined in the Reference Description (http://dublincore.org/documents/dces/), from which the table below has been extracted for your convenience.

DC Element	Definition	Comment
Title	A name given to the resource.	Typically, Title will be a name by which the resource is formally known.
Creator	An entity primarily responsible for making the content of the resource.	Examples of Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.
Subject	A topic of the content of the resource.	Typically, Subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource. Recommended best practice is to select a value from a controlled vocabulary or formal classification scheme.
Description	An account of the content of the resource.	Examples of Description include, but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content.
Publisher	An entity responsible for making the resource available.	Examples of Publisher include a person, an organization, or a service. Typically, the name of a Publisher should be used to indicate the entity.
Contributor	An entity responsible for making contributions to the content of the resource.	Examples of Contributor include a person, an organization, or a service. Typically, the name of a Contributor should be used to indicate the entity.
Date	A date of an event in the lifecycle of the resource.	Typically, Date will be associated with the creation or availability of the resource. Recommended best practice for encoding the date value is defined in a profile of ISO 8601 and includes (among others) dates of the form YYYY-MM-DD.
Type	The nature or genre of the content of the resource.	Type includes terms describing general categories, functions, genres, or aggregation levels for content. Recommended best practice is to select a value from a controlled vocabulary (for example, the DCMI Type Vocabulary). To describe the physical or digital manifestation of the resource, use the FORMAT element.
Format	The physical or digital manifestation of the resource.	Typically, Format may include the media-type or dimensions of the resource. Format may be used to identify the software, hardware, or other equipment needed to display or operate the resource. Examples of dimensions include size and duration. Recommended best practice is to select a value from a controlled vocabulary (for example, the list of Internet Media Types [MIME] defining computer media formats).

DC Element	Definition	Comment
Identifier	An unambiguous reference to the resource within a given context.	Recommended best practice is to identify the resource by means of a string or number conforming to a formal identification system. Formal identification systems include but are not limited to the Uniform Resource Identifier (URI) (including the Uniform Resource Locator (URL)), the Digital Object Identifier (DOI) and the International Standard Book Number (ISBN).
Source	A Reference to a resource from which the present resource is derived.	The present resource may be derived from the Source resource in whole or in part. Recommended best practice is to identify the referenced resource by means of a string or number conforming to a formal identification system.
Language	A language of the intellectual content of the resource.	Recommended best practice is to use RFC 3066 which, in conjunction with ISO639, defines two- and three-letter primary language tags with optional subtags. Examples include 'en'or 'eng' for English, 'akk' for Akkadian, and 'en-GB' for English used in the United Kingdom.
Relation	A reference to a related resource.	Recommended best practice is to identify the referenced resource by means of a string or number conforming to a formal identification system.
Coverage	The extent or scope of the content of the resource.	Typically, Coverage will include spatial location (a place name or geographic coordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity). Recommended best practice is to select a value from a controlled vocabulary (for example, the Thesaurus of Geographic Names [TGN]) and to use, where appropriate, named places or time periods in preference to numeric identifiers such as sets of coordinates or date ranges.
Rights	Information about rights held in and over the resource.	Typically, Rights will contain a rights management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights. If the Rights element is absent, no assumptions may be made about any rights held in or over the resource.

Weibel and Miller explain the strengths and limitations of the DCMES in an article, *Building Blocks for the Semantic Web* [14]—

> *The Dublin Core Metadata Element Set … can be viewed as the common semantic building block of Web metadata. Its 15 broad categories (elements) are useful for creating simple, easy-to-understand descriptions for most information resources. Most communities need additional semantics to fully describe their resources, however. … The DCMES is the basic block, but other chunks of metadata can be combined with it to form richer descriptions.*

Later in the article, they add:

> *The Dublin Core Metadata Element Set… is intended to support cross-subject search and retrieval. It can be thought of as a simplistic or pidgin metadata language that helps the user navigate through disparate subjects, languages, and cultures. Adoption of the Dublin Core by governments, libraries, museums, archives, publishers, environmental science repositories, print and e-print archives, to name a few, testifies to its success in this role. There are emerging*

14. http://dublincore.org/2000/09/28-xmlcomarticle.html

applications in the commercial sector, as well, with health care organizations and financial industries using the Dublin Core as the basis for organizing and exchanging information.

8 Metadata interchange and interoperability

The problem

There are many ways in which metadata can be stored, as we have seen. It can be held in a database, or written into XML files, or for that matter held on 5" × 7" index cards, printed in a book or carved in stone (but that's not a portable document format!)

It is not enough to standardise the metadata sets. If computers are to process this information for the purpose of indexing, search and retrieval, there also have to be standardised methods for storing metadata in electronic form and accessing it. Various approaches are being explored.

- Where the information is stored in an online database, and is being queried by users on another system, there needs to be a way to ensure that the enquiry interface being used by the searcher 'mates with' the query system of the database being accessed. This would not only help one-off queries, but would help disparate systems to maintain a 'knowledge' of each others' contents on an ongoing basis. **Z39.50** is an example of a standard that facilitates this (see page 25).
- Individual information objects such as documents may have metadata embedded within them, rather like the EXIF standard embeds camera data and colourspace data in TIFF and JPEG photographic image files. Adobe have proposed a mechanism called the **Extensible Metadata Platform** (**XMP**), and have implemented this in their Creative Suite products (including Photoshop, Illustrator and InDesign), and also to a lesser extent in FrameMaker (see page 26).
- Wherever the metadata ends up being stored, it is important that it be in an open format, with well-defined and standardised rules for structure, and an explicit declaration of the metadata elements and their permitted values, so that metadata can be interchanged and processed readily. The clear front-runner here would seem to be **RDF**, the **Resource Description Framework** (see page 27).
- Together, these developments are said to be evolving to an overhauled and more useful World Wide Web, for which the term **Semantic Web** has been coined (see page 29). The ambition here is that semantic mark-up would be sufficiently rigorous that machines would be able to process Web-based information on our behalf, acting as our 'agents'.

Let us now examine these endeavours a little more closely.

Z39.50

Z39.50 is an American national standard for information retrieval that has been widely adopted around the world. The standard specifies rules and procedures for how two systems should communicate, for the purposes of database searching and information retrieval, even though they might run on different hardware and software. In 1998 it also became an international standard – ISO 23950.

In Z39.50 parlance, the system from which a user initiates a search is called the Origin, and the system which is being searched is called the Target. The two systems are linked together in a dialogue known as a *Z39.50 association*. The searcher enters the query into the local system, using the menus and command language of that local system, with which he or she is hopefully familiar. This query then gets translated into a standardised form defined in Z39.50, so the Target system can understand them. Similarly, the results of a search on the Target system are converted into a standardised form for return to the Origin system, and the searcher.

As is explained in a paper by Fay Turner of the National Library of Canada on the web site of the International Federation of Library Associations (IFLANET):

> *No matter what the underlying platform or library software used, systems are able to participate in an information retrieval session providing both systems support the Z39.50 standard. Z39.50 can be used for a wide range of library functions that require database searching, from cataloguing and interlibrary loan to reference. It can be used with biboliographic databases as well as with other types of databases such as those containing the full texts of documents and images.*[15]

15. An Overview of the Z39.50 Information Retrieval Standard by Fay Turner: http://www.ifla.org/VI/5/op/udtop3/udtop3.htm

It is also used in the museum and art gallery community.

Embedded metadata: the case of IPTC Headers

The International Press Telecommunications Council is an industry body for the newspaper business, and they early on realised that there was a need to be able to embed information in photographic images that were being sent e.g. by a photo or news agency to a nespaper picture desk. The focus here was on two image file formats widely used in publishing – TIFF (Tag Image File Format) and JPEG (Joint Photographic Experts Group), firstly because these both support the CMYK model used in colour printing, but also because they are extensible file formats that allow extra information to be added within optional header blocks.

Adobe Systems collaborated with the IPTC on this initiative, which made a lot of sense because Adobe Photoshop is far and away the leading application used in the publishing world for processing images. Therefore, Photoshop's native file format has been extended so that it too can be a receptacle for IPTC information. And, for all three file formats, Photoshop can embed textual fields for (among other things) an author's name, a caption, keywords, location data for where the photo was taken, a copyright notice,

the level of urgency with which a photo must be handled, and a three-letter 'AP Code' which relegates the image to an industry-standard categorisation scheme for news stories and photographs.

Latterly, Adobe has started handling this IPTC information via a different mechanism, namely XMP. But XMP is about much more than just that.

Extensible Metadata Platform (XMP)

Esssentially, what Adobe proposes, and has started to implement in its own applications, is a standardised way of adding metadata 'labels' to binary files, and to key components within those files. For example, if InDesign is used to create a complex publication such as a newspaper, the whole document could get an XMP 'label', but so too could each story and each photograph.

A story might acquire its XMP metadata labelling as it is authored by a journalist using Adobe InCopy, a photo might come from an agency with XMP labelling attached, and they would all be folded into the publication. If the completed publication is then converted to a PDF document, those labels would be preserved in the PDF, readable by any software constructed with the capability to scan the file and recognise and read the labels.

Adobe has published the specifications for a binary structure called the **XMP packet**, sandwiching the metadata between a machine-recognisable standard header and trailer block. As for the metadata block – the sandwich filling – that builds on two other open and well-documented standards:

- the data is written in XML form; and
- it is structured according to the rules of the Resource Description Framework, about which more below.

An Adobe white paper on XMP claims that their Adobe Metadata Framework is 'one of the first major, comprehensive implementations of RDF'. As such it is a major investment for their business:

> *One of the reasons the XMP initiative is the first big XML/RDF metadata project for publishing is the very significant cost of creating the entire development infrastructure. The programming tools and toolkits are an example of costly items that do not make sense if a company has only a single publishing application. In Adobe's case the ability to reuse the implementation code in a broad array of applications justifies the cost of the internal developer tools.*
>
> *Adobe will be providing basic tools for the external community as well, including XMP libraries, SDKs and other resources. It is anticipated that as the technology matures, suites of developer tools will come to market, especially in the area of interfacing XMP to existing enterprise database and media management environments.*[16]

16. 'A Manager's Introduction to Adobe eXtensible Metadata Platform, the Adobe XML Metadata Framework'. Download as a PDF from http://www.adobe.com/products/xmp/pdfs/whitepaper.pdf

As it turns out, if you think that you can take Adobe InDesign software out of the box and author publications with full XMP-embedded Dublin Core metadata, you will be disappointed. Initially, Adobe has concentrated on a metadata set that is beneficial in integrating publication workflows. You can

define quite a lot, however, including: *Document title, Author, Description, Keywords, AP category and AP supplemental categories*. Embedded images acquire additional metadata denoting their dimensions and colourspace, and digital photographs retain their EXIF data.

However, Adobe stresses in the white paper that 'Domain-specific schemas, like IPTC or NewsML… can be described within XMP Packets.' Support for the XMP framework is coming from Documentum, IBM, Kodak, KPMG, North Plains Systems and other companies.

And now, if readers will forgive the somewhat back-to-front method of introducing these strands of metadata technology, it is time to explain the Resource Description Framework of which XMP is an implementation…

RDF, the Resource Description Framework

According to Rael Dornfest, chief technology officer at O'Reilly Media,[17] the ideas for RDF emerged from reflections about an earlier metadata project, PICS – the Platform for Internet Content Selection. This provided a simple metadata format for classifying and rating Web pages, primarily to protect children on the Internet and filter pornographic content. The PICS project gained experience in deploying metadata vocabularies, digital signatures and the like, but lacked a 'namespaces' mechanism to allow all the various independently-managed metadata vocabularies to play together.

17. See 'The Power of Metadata', an excerpt from the O'Reilly book *Peer to Peer, Harnessing the power of disruptive technologies* – http://www.openp2p.com/pub/a/p2p/2001/01/18/metadata.html

The purpose of RDF is to provide a standardised framework or method for describing metadata and interchanging it. To explain how it works, we'll need to review some definitions.

- **Resource** — in RDF-speak, a Resource is anything that can have a **URI** (Uniform Resource Identifier). This includes every single web-page, each of which has a Uniform Resource *Locator*, a limited type of URI that refers to a network location, but also could apply to document components. Indeed, one could create URIs to refer to people, to cars, to books in a library, or to abstract concepts. If it has identity, you can give it a URI! And, of course, you can make statements about it.
- **Property** — a Property is a particular type of resource, one that can be used as a property of other resources. Examples of RDF Properties for publications could include Title or CreationDate. We treat them as Resources in their own right (a) because they have identity and (b) so that they in turn can be assigned Properties.
- **Statement** — an RDF Statement is a combination of three things: a Resource, a Property, and a Value. To give two examples: Hamlet is a type of literary work with value = play; also, Hamlet has a creator with value = William Shakespeare.
- Just to confuse you, these three things are sometimes described in RDF literature as Resource, Property and Value, and sometimes grammatically as Subject, Predicate and Object!

- Please also note that the Value may just be a string of text or numbers, or may be treated as a Resource in its own right. Certainly in a library context we would expect William Shakespeare to be so treated.
- **Statements** can themselves be treated as Resources, and so they too can have Properties. For example, the statement that Lucien French has access rights with the value = 'may upload files to the server' is likely to provoke the response 'Where's the proof of that?' – and that statement would therefore have to have a verification property, which might be the URI of an encrypted digital certificate.
- The RDF standards specify a straightforward way of expressing these statements in the syntax of XML. In fact, RDF defines a specific XML mark-up language, **RDF/XML**, for representing RDF information and exchanging it between machines.

I am still myself in the early stages of grappling with the complexities of RDF, and anyone wanting to know more should study the World Wide Web Consortium's online literature, such as the 'RDF Primer' put together by Frank Manola and Eric Miller.[18] I shall therefore limit myself to general observations about the implications of RDF for knowledge and information management, retrieval and publishing.

18. http://www.w3.org/TR/rdf-primer/

The key thing to realise is that RDF allows communities of interest the freedom to create their own **customised metadata vocabularies**: no-one will be forced to shoehorn their particular type of information into an inappropriate categorisation scheme. However, because RDF standardises the way in which statements about resources are expressed, it enables machines to parse them easily. In a networked environment, machine-readable explanations of Properties can be made available on-line, so that machines can learn about vocabularies they haven't encountered before.

The RDF Primer mentioned above gives a number of case studies of RDF vocabularies that are already helping us tackle information better. For example:

- **RSS** — which some people say means 'Really Simple Syndication' but which more officially is the **RDF Site Summary** standard, has emerged as a technology that allows Web site owners to syndicate parts of their content to appear automatically within the interface of other sites that subscribe to the news feed. Alternatively, individuals may subscribe to the feeds with a desktop news-reader application, and the Firefox browser lets you add automatically updated drop-down menus of headlines from news sites to which you subscribe (in my case, BBC World Service News).
- **Dublin Core** — This metadata standard has already been described on page 22 and following, above. While Dublin Core metadata can be stored in all sorts of forms, such as in a relational database, RDF is an ideal method for encoding it. Thus encoded, it becomes possible to embed the RDF-expressed metadata directly into XML/HTML documents, or in PDF via XMP (see page 26).

- **PRISM — Publishing Requirements for Industry Standard Metadata** is a standard hammered out by magazine publishers and the vendors who support them. The emphasis of this project has been to provide extensive means for categorising subjects, using multiple subject description taxonomies; to perform 'rights tracking' for resources such as photos, the use of which has been licensed from others, and for the onward re-use of content in other editions, media etc; and to ensure that metadata is not discarded as content moves from one stage in the publishing process to another.

Bit by bit, emerging standards like these are giving us tools for describing and categorising all sorts of different types of information in a way that machines can finally 'get a handle on'.

9 Ontologies and the Semantic Web

The vision

The Semantic Web may be the Web of the future, but it is not a different Web from the one we use today. It is the vision of a Web that has been reinforced to make it easier for machines to act as assistants in managing knowledge and information resources. As the Web's original creator Tim Berners-Lee wrote in an early speculative document in 1998:[19]

> *The Web was designed as an information space, with the goal that it should be useful not only for human–human communication, but also that machines would be able to participate and help. One of the major obstacles to this has been the fact that most information on the Web is designed for human consumption, and even if it was derived from a database with well defined meanings (in at least some terms) for its columns, that the structure of the data is not evident to a robot browsing the Web. Leaving aside the artificial intelligence problem of training machines to behave like people, the Semantic Web approach instead develops languages for expressing information in a machine processable form.*

19. 'Semantic Web Roadmap' by Tim Berners-Lee, September 1998. See http://www.w3.org/DesignIssues/Semantic.htm

Later in the May 2001 edition of *Scientific American,* Berners-Lee together with James Hendler and Ora Lassila[20] sketched a more exciting picture of how agent technologies embedded in phones, PDAs and other devices as well as more conventional computers could work on our behalf to make appointments, recommend services, screen interruptions and so on, based on an ability to process machine-readable statements in data.

20. ''The Semantic Web': see http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2

Solutions like this could revolutionise the way we work and get things done. To name but one field of endeavour – healthcare is a field that relies heavily on the accurate transmission of patient information 'statements' between practitioners and services and patients, and the more these statements can be made machine-readable through mechanisms such as RDF, the better the chance we have of improving patient care.

But RDF *itself* seems to pose a problem…

Beyond Babel

XML is a brilliant tool for letting us define customised tagging schemes. Based on that, RDF is a wonderfully flexible tool for constructing descriptive vocabularies of categorisation and definition. But how do we ensure that all these vocabularies work together?

Even when two systems ostensibly describe the same kinds of information, we may find that the systems label that information differently. System A calls 'periodicals' what System B calls 'journals'. To take an example from the *Scientific American* article just mentioned, an 'address' in one database may be where mail can be delivered, including a post box, whereas in a different context an 'address' has to be a physical building, or in a third context it could mean a political speech.

Ontologists to the rescue?

Here is seems best to cite the *Scientific American* article by Berners-Lee, Hendler and Lassila directly:

> *A solution to this problem is provided by the third basic component of the Semantic Web, collections of information called* ***ontologies***. [21] *In philosophy, an ontology is a theory about the nature of existence, of what types of things exist; ontology as a discipline studies such theories. Artificial-intelligence and Web researchers have co-opted the term for their own jargon, and for them an ontology is a document or file that formally defines the relations among terms. The most typical kind of ontology for the Web has a taxonomy and a set of inference rules.*
>
> *The* ***taxonomy*** *defines classes of objects and relations among them. For example, an* address *may be defined as a type of* location*, and* city codes *may be defined to apply only to* locations*, and so on. Classes, subclasses and relations among entities are a very powerful tool for Web use. We can express a large number of relations among entities by assigning properties to classes and allowing subclasses to inherit such properties. If* city codes *must be of type* city *and cities generally have Web sites, we can discuss the Web site associated with a* city code *even if no database links a city code directly to a Web site.*
>
> ***Inference rules*** *in ontologies supply further power. An ontology may express the rule 'If a city code is associated with a state code, and an address uses that city code, then that address has the associated state code.' A program could then readily deduce, for instance, that a Cornell University address, being in Ithaca, must be in New York State, which is in the US, and therefore should be formatted to US standards. The computer doesn't truly 'understand' any of this information, but it can now manipulate the terms much more effectively in ways that are useful and meaningful to the human user.*

21. Op. cit. – Emphases in this passage are mine. (CT)

Ontologies in practice

Professor Ian Horrocks explained in his BCS Roger Needham lecture in December 2005 how ontologies are already being of practical use in various disciplines. For example, medical practice uses SNOMED (Systematized Nomenclature of Medicine), which is a controlled vocabulary for indexing medical records; and GALEN, which is a computer-based multilingual system which links together the clinical terminologies of different European languages though a common reference model that is written in the GALEN Representation And Integration Language (GRAIL). Ian Horrocks also mentioned engineering ontologies in use by NASA, Lockheed-Martin and General Motors.

An ontology for the Web

Ian Horrocks also traced efforts to develop an ontology language for the Web. He said that **RDF Schema**, which is the metadata schema for RDF, has the characteristics of an ontology language because it organises the resources it references into classes, provides for a hierarchy of classes, and records the properties of resources and classes. But…

> *The problem with RDF Schema is that it is very weak; it doesn't allow you to express many things. Many features are missing that people building complex ontologies describing domains like medicine, biology and so on would need… [so there are ontologies] that RDF can't express. The other funny thing about RDF is that it has a kind of 'higher-order' flavour with not very standard semantics, which makes it difficult to provide reasoning support.* [22]

22. Transcribed by Conrad Taylor from an audio recording of the 2005 Needham Lecture by Ian Horrocks.

Efforts to redress these deficiencies on both sides of the Atlantic led to the a couple of prototype ontology languages which were merged and developed into the **Web Ontology Language** (**OWL** for short). This is now an official recommendation of W3C, the World Wide Web Consortium. To quote from the main W3C page about OWL: [23]

23. See http://www.w3.org/2004/OWL/

> *OWL is a Web Ontology language. Where earlier languages have been used to develop tools and ontologies for specific user communities (particularly in the sciences and in company-specific e-commerce applications), they were not defined to be compatible with the architecture of the World Wide Web in general, and the Semantic Web in particular.*
>
> *OWL uses both URIs for naming and the description framework for the Web provided by RDF to add the following capabilities to ontologies:*
>
> - *Ability to be distributed across many systems*
> - *Scalability to Web needs*
> - *Compatibility with Web standards for accessibility and internationalization*
> - *Openness and extensibility*

> *OWL builds on RDF and RDF Schema and adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. 'exactly one'), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes.*

Ian Horrocks described OWL as having benefitted from about fifteen years of research into **description logics**. Description logics (DL) are a family of formalisms in Knowledge Representation (KR), mainly distinguished from each other by the logical operators they use. The description logic behind OWL is known by the acronym *SHOIN*, which indicates these key features:

- (*S*) – an extension of the standard basic DL known as 'ALC', which has incorporated the ability to characterise certain roles/properties as being transitive.
- (*H*) – Hierarchy not only of classes but also of roles – such as 'has a daughter' is a sub-property of 'has a child';
- (*O*) – The ability to name the members of a class, including singleton classes;
- (*I*) – The ability to show that roles are each other's inverse, such as 'is child' is the inverse of 'has child';
- (*N*) – The ability to define a numerical restriction on the relationships (via properties) that members of a class can have.

Thus, according to Ian Horrocks, from the point of view of Description Logics, 'An OWL ontology is just a web-friendly version of a *SHOIN* Knowledge Base.' [24]

24. Ian Horrocks comments: 'My remarks relate mainly to OWL DL – OWL Full allows the DL syntax to be "abused" in a way that takes it outside the world of DLs, and in fact outside the world of First Order Logic.'

(Correspondence with author.)

10 Topic Maps: another way to link knowledge to information

Coming from a different direction

While the Resource Description Framework initiative arose from within the computing community – specifically, the World Wide Web Consortium in their quest for the Semantic Web – the idea of Topic Maps has been fostered within the ISO standards community. Steve Pepper of Oslo-based company Ontopia comments that both RDF and Topic Maps are attempts to tame and harness information resources in an age of 'infoglut' by applying knowledge representation techniques to information management. However, there are differences of emphasis and 'direction':

> *Topic mapping has its roots in traditional finding aids such as back-of-book indexes, glossaries and thesauri. RDF has its roots in formal logic and mathematical graph theory. Topic mapping is knowledge representation applied to information management from the perspective of humans. RDF is knowledge representation applied to information management from the perspective of machines. This accounts for some of the critical differences between the two.* [25]

25. 'Ten Theses on Topic Maps and RDF' by Steve Pepper: http://www.ontopia.net/topicmaps/materials/rdf.html

In another briefing paper, *The TAO of Topic Maps*, Steve Pepper explains why the useful skills that have been applied for centuries in indexing books are needed more than ever to deal with the vast quantities of information resources in which human knowledge is now represented.

> *Up until now there has been no equivalent of the traditional back-of-book index in the world of electronic information. True enough, people have marked up keywords in their word processing documents and used these to generate indexes 'automatically', but the resulting indexes have remained firmly within the paradigm of single documents destined to be published on paper. The world of electronic information is quite different, as the World Wide Web has taught us. Here the distinction between individual documents vanishes and the requirement is for indexes to span multiple documents, and in some cases, to cover vast pools of information, which in turn calls for the ability to merge indexes and to create user-defined views of information. In this situation, old-fashioned indexing techniques are pitifully inadequate.*
>
> *The problem has been recognized for several decades in the realm of document processing, but the methodology used to address it — full text indexing — has only solved part of the problem, as anyone who has used search engines on the internet knows only too well.*
>
> *The main problem with full text indexes is their lack of discrimination. They index everything: Imagine creating a traditional back-of-book index by taking every single word in the book, removing a couple of hundred of the most obviously useless ones, and then including every single usage of those that remain. Even with some intelligence to allow for inflected forms the result would be of no practical use whatsoever. Mechanical indexing cannot cope with the fact that the same subject may be referred to by multiple names (the 'synonym problem'), nor that the same name may refer to multiple subjects (the 'homonym problem'). And yet this is basically how a web search engine works (no wonder you always get thousands of irrelevant hits and still manage to miss the thing you are looking for!).* [26]

26. 'The TAO of Topic Maps' by Steve Pepper: http://www.ontopia.net/topicmaps/materials/tao.html

In contrast, a traditional index is a carefully constructed guide to knowledge. It is a list of the topics covered in the book, and it points to the occurences of those topics in the book (i.e. where the reader will find them). Typical useful features of an index, which Topic Maps seek to replicate, are:

- The synonym problem is addressed by listing topics under a range of names under which readers might expect knowledge topics to be found: for example 'integrated circuit' or 'chip'. Sometimes this is dealt with by redirection entries. ('Chip – *see* integrated circuit.')
- Associated topics provide the reader with a wider net of entry points into the knowledge resource. ('*see also* microprocessors.')
- Sub-entries also point out associations between topics, especially where hierarchies or contexts are useful to honing the reader's search. ('Bronze Age. *China* – p. 24; *Persia* – p. 27; *Greece* – p. 32')

- The homonym problem is dealt with by disambiguation entries so that for example 'bus' (a form of wheeled transport) can be distinguished from 'bus' (a subsystem in computer architecture that transfers data or power between components).
- Entries are also 'typed' in various ways, usually in a book index by certain typographic conventions: for example by using italics to distinguish between Hood (the Admiral, Lord Samuel), and *Hood* (the battle cruiser, sunk in 1941 in the Battle of the Denmark Strait).

Thus **topics** in an index have **associations** of various kinds between them, and the index points to the **occurrences** of these topics within the body of work being referenced. It is this structure that allows an index to act as a kind of 'map' to the knowledge in a book.

The Topic Maps standard

The idea of electronic Topic Maps arose from the work of the Davenport Group which devised **DocBook** as a standard SGML DTD for the creation and maintenance of technical documents. Their concern with ways of merging indexes between documents led to the research work that has led to the definition of the Topic Maps standard, **ISO 13250**. This in turn has led to an effort to realise the 'topic maps paradigm' in the form of a standardised expression in XML – the **XML Topic Maps (XTM)** specification.

As you might expect, the central concept in Topic Maps is – the Topic. The role of a topic within a topic map is to represent a subject. Subjects can be anything at all – perhaps something as physical as Mt. Everest or Tony Blair, or an electronic artefact such as a Web page, or this paper you are reading now (perhaps on screen, perhaps on paper). As a 2001 paper on the XML Topics Maps (XTM) specification explains:

> *Because not all subjects are electronic artifacts … we cannot provide an address for the subject. Instead, we provide an electronic surrogate for the subject, which (being electronic) can have an address. This surrogate we call a topic. Every topic acts as a surrogate for some subject. We say that the topic 'reifies' the subject — or makes the subject 'real' for the system. The creation of a topic that reifies a subject enables the system to manipulate, process, and assign characteristics to the subject by manipulating, processing, and assigning characteristics to the topic that reifies it. When we need an address for the subject, we give the address of a topic which reifies it, and acts as its surrogate within the system.* [27]

27. XML Topic Maps (XTM) 1.0: http://www.topicmaps.org/xtm/

This relationship between subjects and topics – and the possibility that two or more topics might be reifying the same subject – means that Topic Maps have to be extremely rigorous in defining the identity of the subjects of topics. It is only through the practice of such rigour that it becomes possible to exchange and merge Topic Maps.

Subjects which are electronic artefacts can be identified with a URI. Subjects which are *not* electronic artefacts, such as Mt. Everest, are identified through the use of a **subject indicator**, which is an electronic resource and is there-

fore in turn addressed via its URI. One extremely useful feature of the XTM specification is the concept of a **published subject indicator** (PSI), defined thus in the XTM 1.0 specification:

> *A published subject is any subject for which a subject indicator has been made available for public use and is accessible online via a URI. A published subject indicator is therefore any resource that has been published in order to provide a positive, unambiguous indication of the identity of a subject for the purpose of facilitating topic map interchange and mergeability.*

Because PSIs are 'out there', with unique URIs, it has become possible to link them to information resources by adding metadata to the knowledge objects within those information resources. It doesn't take much imagination to see that if more and more information resources on the Web were explicitly declared to be about certain subjects through the use of PSI-referencing embedded metadata, they could be very accurately and appropriately indexed for information discovery purposes.

Names and scopes

Topics in XTM usually have names. There is provision for variant names as well as base names. This allows the same topic to be named differently in different languages (Mt. Everest, or Qomolangma), or for different historical periods. A crucial feature of Topic Maps is the use of **scopes**, which among other things define which of these names are valid in particular contexts. As Martin Bryan has explained to me in personal correspondence:

> *So, for example, the concept of 'gay' has a different meaning in the context of the 18th century from that it has at the start of the 21st century. Hence we need two concepts, both with the matching names, but with different subject identifiers – different occurences but the same title – qualified by scope statements to indicate that 'gay (18th century literature)' equates to ''high jinks (Edwardian literature)' and not to 'gay (21st century literature)' or 'sodomite (18th century literature)'.*

In fact, scopes manage the applicability of not just topic names, but also the other two kinds of topic characteristic defined in XTM: **occurrences** of the topic (any information that is defined as being relevant to that topic), and **roles** for the topic (which defines how it is involved in associations with other topics).

Definition of a Topic Map

Having introduced some of the concepts behind topic mapping, we can now understand the formal definition of a Topic Map:

> *A topic map is a collection of topics, associations, and scopes (collectively called topic map nodes) … The purpose of a topic map is to convey knowledge about resources through a superimposed layer, or map, of the resources. A topic map captures the subjects of which resources speak, and the relationships between resources, in a way that is implementation-independent.*

A Topic Map may exist in some application-internal form, for example in topic mapping software, or in a serialised interchange format expressed in XTM or some other syntax.

Out-of-line linking in Topic Maps implementations

One of the useful features of topic mapping methods is that they make possible the use of out-of-line linking and addressing mechanisms. This means that an individual, organisation or company can make very useful indexes for resources to which they don't have 'write access' to modify the information resource.

An interesting example of Topic Maps in e-learning which demonstrates this advantage is provided by the **BrainBank Learning** action–research project with ten girls and six boys aged 13–14 at a school in Alsvåg in Norway, implemented with Ontopia's OKS suite, as explained in a paper *BrainBank Learning – building personal topic maps as a strategy for learning* presented at the XML 2004 conference:

> *Users enter the application through individual accounts. Topics (keywords) that the learner meets during education activities are entered and described using BrainBank Learning. The topics can then be connected by describing associations between them (Topic Maps provides excellent support for this). Thus the learner is creating his own associated network of topics and this represents his documented knowledge. This way of documenting in the learning process is good for the learner's understanding of the area of study (placing knowledge in a context), as well as navigating and overview of the acquired knowledge later on.*
>
> *To further describe topics and associations in BrainBank Learning, digital resources such as documents, pictures, movie clips and sound clips can be attached to the topics. (These resources can be either linked to or uploaded to and stored in BrainBank).*
>
> *BrainBank Learning stimulates the learning process as the learner continuously reflects through and updates his own knowledge and stores it in BrainBank. This is because he has to discriminate received information to extract the essence of the information to document it in BrainBank Learning, and also by relating new information to already existing knowledge by associating new topics to existing ones (and describing the relation between them).* [28]

28. http://www.idealliance.org/proceedings/xml04/abstracts/paper21.html

For advanced applications, naturally it takes a great deal of knowledge of the domain in order to construct valuable Topic Maps. But these maps of whole domains of knowledge can themselves be valuable resources for anyone who needs to organise a body of information about this domain. And as Steve Pepper explains:

> *This … opens up new business opportunities for creating and selling 'portable topic maps' that can be overlaid on multiple information pools. For traditional commercial publishers, producing well-crafted topic maps could be a new way of leveraging their existing knowledge and experience and combating the threat to their*

existence posed by the vast amounts of information now available for free.

As for their potential role in managing the knowledge assets of companies and other large organisations, he adds:

> *The ability to encode arbitrarily complex knowledge structures and link them to information assets indicates a major role for topic maps in the realm of knowledge management: Topic maps can be used to represent the interrelation of roles, products, procedures, etc. that constitute corporate memory, and link them to the corresponding documentation.* [29]

29. 'The TAO of Topic Maps' by Steve Pepper: see note 23 for URL

RDF versus Topic Maps?

What then is the relationship between the approach taken by the Resource Description Framework and that of Topic Mapping? While Steve Pepper is at pains to point out the differences, he is also careful to discount any idea that they are competitors. He says that at Ontopia, they have come to regard the two approaches as complementary, and that their software tools use RDF to assist in the automated generation of topic maps.

> *RDF is resource-centric, whereas topic maps are subject-centric. In RDF one starts with information resources and attaches metadata structures to them; in topic maps, the primary focus is the subjects that the information is 'about'. So in one sense RDF and topic maps have diametrically opposed points of view. (To some extent, this difference in focus parallels that between document languages, such as the Anglo-American Cataloguing Rules, AACR, and subject languages, such as the Library of Congress Subject Headings, LCSH, in the domain of library science.) However, 'resource' in RDF and 'subject' in topic maps can be regarded as synonyms, since information resources can (also) be 'subjects' in topic maps; and 'resources' in RDF do not have to be addressable information resources – so the difference is dialectical rather than diametrical.* [30]

30. 'Ten Theses on Topic Maps and RDF' by Steve Pepper: see note 24 for URL

And how does the relationship between ontologies and topic maps appear from the Description Logicians' side of the valley? They would point out that the OWL initiative, especially OWL DL with its emphasis on formalism, is an attempt to guarantee that knowledge representations are computable. Ian Horrocks again:

> *I don't think that [Topic Maps] are directly comparable with RDF and OWL. Rather I think that Topic Maps could be an* application *of OWL/RDF – i.e. OWL/RDF (or perhaps some successor to them that provides some additional expressive power with respect to the meta-classes that are used by Topic Maps) could be used to write Topic Maps. This would make their meaning more precise, allow additional constraints to be specified, and make them more accessible to machine processing.* [31]

31. Personal correspondence with author.

11 Managing knowledge about knowledge management

Making the implicit, explicit

In this paper I have concentrated chiefly on the ways in which computing technologies, techniques, languages and standards are evolving to help us get a better 'handle' on the information resources and information products in which we store human knowledge.

I believe we mustn't lose sight of the fact that knowledge is something that lives in human minds, and that minds exist in a social context. Knowledge is sought after to achieve human ends and ambitions, and the management of knowledge and information is an essentially human endeavour.

In fact, as I was writing the later parts of this paper, addressing the subjects of ontologies and topic maps, I came to the realisation that construction of any knowledge management artefact has to proceed from an intimate knowledge of the domain to which the ontology or topic map refers – be it medicine, genetics, government, computing, business or whatever.

Much of this knowledge is *implicit*, and often there is nothing quite like a formal knowledge/information management project (such as hammering out a metadata schema, ontology, topic map or DTD) for forcing domain experts to pull out their assumptions and ways of structuring their view of the domain into the open, where they can be debated.

What does this mean for our discussions inside the BCS about knowledge and information management? Well, 'knowledge and information management' itself is a domain, and also an overlap of several other domains, and we would do well to consider a broad view of who the 'domain experts' would be to move such a discussion forward constructively.

Throughout the computing community – and beyond

In starting 'the KIMtec discussion' in 2005, a couple of dozen people within the British Computer Society expressed a willingness to share perspectives on this domain that have been nurtured within our own Specialist Groups – Electronic Publishing, Information Retrieval, Internet, Data Management, Sociotechnical, Artificial Intelligence and other SGs. In preparation for the 6th March 2006 day of discussion, we also attracted interest from BCS headquarters and from elsewhere.

But information and management is not just a computing issue (to return to my starting point).

Important and central as ICT is becoming in information management, we should reach out to other communities of practice whose knowledge and skills and contributions would be essential; and so I urge everyone to look over the Information Technology parapet to espy other communities with whom we might want to dialogue in any ongoing discussion.

- **Librarians**, as I remarked on page 4, have long had responsibility for managing information resources and have decades of experience in doing so with computers. As information resources have become less book-like, there has been a tendency for these people to define themselves as 'information professionals', especially when working in a business context. I think that members of CILIP, [32] the Chartered Institute of Library and Information Professionals, may be interested in dialoguing with us, as would members of KIMNET, the Knowledge and Information Management Network of Aslib. [33]

- **Information Designers** are practitioners who design how information is presented and made accessible to people. ID started as a specialism within graphic design, with links to technical authoring, expanding into design of Web sites and interactive interfaces to information. ID practice incorporates concern for what information people want, how they use it in social and institutional contexts, and how usable it is. For some of us (I count myself as an Information Designer), it seems right to be concerned with how information sources are structured, as well as the display forms that present information to people. [34]

- **Publishers**, the community that the Electronic Publishing Specialist Group is in touch with – including some subsidiary communities of skill such as editors and indexers – have long been a repository of knowledge about how to make information resources usable. At a recent discussion meeting [35] which the EPSG Committee held with several external experts to consider the future direction of publishing (and of the Group), it was noted that boundaries are becoming very blurred between publishing and information management.

To this short list we might add all of the domain experts in fields such as medicine, science, government, scholarship and so on who are looking to ICT tools and methods to assist them in managing the knowledge at the heart of their practice. There should therefore be quite a lot of people with an interest in learning about and discussing ICT techniques for managing knowledge, information, data and metadata; because these techniques are likely to become increasingly important to how society operates.

Conrad Taylor
February 2006

I wish to acknowledge the help I have received from comments on previous drafts of this paper, especially those from Martin Bryan, Berin Gowan and Ian Horrocks.

32. http://www.cilip.org.uk/

33. For a link to KIMNET on the Aslib website, see http://www.aslib.co.uk/members/kimnet/index.htm

34. This much is evident on the main portal site for Information Design: http://informationdesign.org

35. Discussion documented here: www.epsg.org.uk/directions/