ιδεοGRAPHY

Getting a handle on it:

achieving efficiency by improving the structure, management and design of information

Conrad Taylor

This paper was written to accompany a lecture, illustrated with slides, given by the author at a conference in November 1997 in Stockholm organised by the training company Populär Kommunikation.

It has been placed for international distribution on Conrad Taylor's Web site: **www.conradiator.com**

Getting a handle on it:

achieving efficiency by improving the structure, management and design of information

Conrad Taylor, ιδεοGRAPHY *— and Popular Communication Courses*

Whenever human society has has made advances in the complexity of its organization, it has required advances in the organization of its base of knowledge. In those ancient times when the pyramids were built, as Lewis Mumford has pointed out, the civilizations of Egypt, the Indus and Mesopotamia already possessed impressive large-scale machines – made of human components! The high degree or organization of those societies would not have been possible without concomitant developments in record-keeping, and methods for storage and transmission of knowledge between co-workers, and between generations.

This pattern has been repeated throughout history, with a different emphasis in each age. In the Middle Ages, warfare and religion required most organization; the Renaissance saw developments in accountancy, politics and diplomacy. In the 17th and 18th centuries, long-distance seafaring and colonial conquest led to advances in the organization of geographical knowledge; in the 19th century, the 'captains of industry' presided over the increase in the technological complexity of manufacturing processes, which required knowledge to be accumulated and passed on to armies of specialists such as mechanics, operatives, miners and engineers, often through long apprenticeships.

The ocean of information we need to do our jobs

Whatever the appeal of the 'lean organization', one thing that defies trimming is the amount of information we need to be able to do our jobs. If anything, it's getting worse:

- Products are becoming more complex, both to make and to use. Consider the car, mobile phone, camcorder and computer; or food packaging machinery, printing presses, drilling rigs, aircraft and telephone switchboards. Some people need the information to help them to make these products, or to maintain and repair them, while others need the information necessary to use them.
- Some products are simple to make and to use, but come from a complex process of research, development and testing, all of which needs to be documented. The pharmaceutical industry gives many examples. An anti-depressant medication may be a simple tablet;

Revision: 2 November, 1997

but developing it takes decades of cumulative research, molecular modelling, clinical trials, extended assessment of interactions with other medications and medical conditions, correspondence, submissions of documents to licensing authorities, and so on.

- Many product and service offerings are delivered in customised ways; the ability to deliver custom products is one way companies compete. Such companies face a challenge in providing their sales force or agents with accurate, up-to-date information about the features and options with which their products are available.
- Consumer awareness and legislation makes it more important for companies to be aware of product liability law, and the standards to which their products (and the information which describes them) must comply.
- Information is a commodity in its own right, and an important component of other products and services. Take transport and tourism; their services would be useless without information on routes, times, connections, prices, and room availability. Also, most businesses, particularly utilities and financial services, find that customers want access to more detailed breakdown of transactions.

The result is a flood of information, and organizations are experiencing only partial success in managing it. However, simplistic calls to 'trim' information must be regarded with suspicion. I am reminded of the response which Mozart is said to have made to a royal patron, who complained that his music had 'too many notes'. Very well, said the composer – which ones shall I take out?

But something has to be done to make information, and the knowledge which organizations possess about their products and processes, more manageable and more readily available to the people who need it. Achieving this takes technology; but also much more besides.

Three kinds of skill involved in handling information in an organisational context

I have come to recognise three kinds of professional activity which go on around the processing of information in organisations:

Information technology — This is the most recognised form of work around information: it gets the big budgets, interest and support at Board level, and lobbying from external vendors who offer continual improvement. It's true that IT professionals have been successful in getting computers to organise such information assets as can be tabulated into fixed-length fields and numbers (especially accounting, order processing and stocktaking), but less successful in organising information which is primarily linguistic, or which describes the properties and behaviours of real objects and social processes – that is, the *knowledge* of the organization.

Information design — This is my preferred term for all the thought and work that must go into ensuring that information is presented in a way that's useful to those who access it. The term encompasses skills in writing, editing, typographic design and diagrammatic representation, and has latterly extended to include skills in hypermedia and interface design, and a commitment to usability testing of information products. Recognition of the importance of these skills has been growing since the 1950s, but is patchy; technical authors have won some professional recognition, for instance in computing and the aerospace industries, but it is still rare for the contribution of information design to be recognised and advocated at Board level in enterprises.

Information management — Concern for systematic management of information resources of organisations first arose among librarians and archivists in public service, or employed by professional bodies and larger companies. As technologies for information storage and retrieval have become digital, on-line data has also become a concern of this group. However, research indicates that few organisations have an effective policy towards the management of their information assets.

The efficient use of information requires the contributions of all three viewpoints and traditions. We need these three kinds of expert to talk to each other, and to understand each other's contribution to the efficient use of information. Perhaps there should be more people trained to work across the boundaries of these fields.

Recognition for information products

The term **information products** has recently arisen as a term for *all* the visible, discrete products through which information is presented to people for their use, whatever its medium of delivery: print on paper or screen, fax sheets, Web sites, multimedia, signs and more. The idea of conscious organisation and presentation is implicit in the definition of an 'information product': a pile of boxes of records of clinical trials can't be called an information product, but an analytical report compiled from this information source *is* an information product.

Examples of information product:

- **For customers:** catalogues, product data sheets, Web sites, price guides, timetables, user guides, assembly instructions, manuals.
- **For internal use:** personnel directories, documentation for office automation systems, on-line help, directories of suppliers, repair/ maintenance manuals, databases of customers, quality assurance process documentation, Intranets.

One trend in the development of information products is that formerly they were very concrete, and looked the same to all recipients: books, reports, and other printed materials. Many are now *query systems*, producing unique and ephemeral products for each different user.

As for the skills required to develop information products, I would argue that each has three dimensions – a technical one, a design one and a managerial one – requiring input of those who know how to manage information and design its structures and interfaces, as well as those know how to make the technology of information do its stuff.

'Cognitive artifacts' and their forms of representation

Humans have always augmented their abilities by inventing artifacts: tools, containers, clothing, shelters. Psychologist Donald Norman points out that we augment our mental abilities in a similar and even more spectacular way, by inventing what he calls **cognitive artifacts**. Some of these are mental in nature: language and number are the most important. We have also created many *physical* cognitive artifacts – objects that store information for us, or allow us to manipulate it, by representing it in another form. Examples include tally-sticks, the abacus, maps drawn in sand, written language, the clay tablet, paper, the address book, Post-It Notes™, and of course software.

Norman makes a useful distinction between cognitive artifacts which represent information on their *surface*, and those which represent at least part of the information *internally*. Pocket notebooks, diaries and address books are good examples of pure surface artifacts; the data is represented in ink and pencil marks, and there's a physical representation of the way in which information is organised. A date in June is near the middle of the diary, and the early notes in a notebook are at the front. Such an artifact is easy to use, (it certainly needs no user manual!), but has obvious limitations: the diary has limited space to write in for each day, if you have lots of Scottish friends the 'M' page of the address book fills up too fast, and it's hard to make links between ideas written in different parts of a notebook.

Compare these artifacts to the electronic personal organizers which for many people have replaced their diary, notebook and address book. Information is represented *internally*, in computer memory. When the machine is switched off, there is no surface representation of the stored information; when it is switched on, the information is represented on the screen in a *mediated* form, determined by the software.

In business and government too we are moving more towards the use of cognitive artifacts which store their contents internally, digitally. The reason is clear: when information is stored in a computable way, you can do more with it. You can see different views of data, search within it, access it over a network and send it to a remote location with a click of the mouse. At least, that's the theory.

Wanted: a new kind of database

A well-tested method of making information searches more productive is to put the information in a relational database, which stores data in 'fields': for a bibliographical application, these fields might include author's first name, surname, title of book, publisher, date of publication, ISBN number and so on. Fields are stored in spreadsheet-like tables, and relationships can be built between the tables.

Big supermarkets in Britain have encouraged most of their customers to accept a 'loyalty card'. Now your name and address is stored in a table of their database. Another table stores each product, with its barcode number and price. The check-out operator swipes your card before she passes the purchases in front of the bar-code scanner – this is how you gain 'points'. And what the supermarket gains is… an intimate history of your shopping habits.

Without a doubt, relational databases have transformed how businesses deal with their *data*. But they have not been able to help us deal with *knowledge*. That is because, to quote Tim Bray of Textuality, 'Knowledge is a text-based application'; it does not fit into the fixed-length fields of a relational database. To store and index, search and retrieve text-based information, we need a new kind of database.

Fumbling around with free text searches

My first experience of on-line databases was in 1990, as a subscriber to a pioneering commercial on-line service, GeoNet, which at that time was not part of the Internet. Through my local GeoNet service provider, the Manchester Host, I could access repositories of on-line information, including back issues of several newspapers and databases of commercial information such as Dialog and Datastar.

The World Wide Web had not yet been unleashed, and the commercial Internet did not exist. This data was in plain 7-bit ASCII text; it was accessed through a command-line dialogue with the remote host, and each host had its own conventions for contructing queries. GeoNet usefully provided a common-denominator search interface, **fIND**, which translated behind the scenes into the specific query languages of the commercial services one could access through it.

The trouble with these databases was that they were basically only file systems; the text within them had no structure. They were probably stored in relational databases, the text itself an undifferentiated slab of ASCII, plus some field-stored indexing data such as title, author, date and keywords. A search of these free-text databases invoked the most primitive search engines, mindless bumbling robots with no ability to discriminate whether a search-string of 'Major' indicated the name of a politician, a rank in the army, a kind of musical key or an adjective giving priority. Try as one might with the Boolean tools of

AND and OR and NOT and the seductively tempting NEAR, one got far too many false hits. I also often wondered how many relevant documents had slipped through the net. (And if this sounds like searching the Web for information, you are right; we seem to have made little progress!)

Dreams of artificial intelligence

In his book *Being Digital*, Nicholas Negroponte of the MIT Media Lab sketches a future in which we will be able to send software agents out on our behalf to trawl the worldwide seas of data, and bring back a shimmering catch of information for us. He also suggests that such agents would be able to learn what we are interested in, even assemble a personalised newspaper (he calls it *The Daily Me*!) which matches what our 'news-agents' construe to match our concerns.

Negroponte's hopes seem founded on significant progress in artificial intelligence. I am not convinced that such capabilities are imminent. As recent books in both conventional and computational linguistics conclude, the ability to fathom the meaning of language, trivial for a child, is a serious challenge for computers.

Inverted trees, and topic indexing

A free-text database is inherently *dis*organized, and can suffer from poor performance because you do not know, when searching 10,000 articles with an average of 2,500 words each, that the word 'polysilicate' is unfortunately in the last paragraph of the last article…

The performance of free text searches can be improved by compiling an 'inverted tree index' of document collections. A simple application of this method is to scan the whole database, extracting from it a list of all the unique words. These words are stored in an index file in alphabetical order, together with 'pointers' to where in the texts these words occur. When searching for the word 'polysilicate', the search engine can move swiftly to the index entry, retrieve the pointers, and move directly to search only the articles indicated by the pointers. This is even more efficient when *two* words are entered as search terms; the index entries for both words can be compared, and only those articles which match both sets of pointers are returned for consideration.

Some points to note about inverted-tree indexing, and the additional capabilities which some companies, such as Verity, have brought to it:

- Indexes can grow very large; some indexing software filters out a list of 'stop words' on the grounds that it is not useful to know how many articles include the word 'and'. But regardless of the size to which the index file grows, it is always faster to search this way because it has structure and logical (*e.g.* alphabetical) sequence.
- It is usual to let someone search on a word fragment, in which case a search on 'carcin' would turn up references to 'carcinoma' as well as 'carcinogen'. But some search engines offer a 'word stemming'

feature, on the grounds that even when searching on 'carcinogen' you might like to turn up references to 'carcinoma' as well. Word-stemming requires that the search engine has reference to an etymological dictionary.

- The Verity software includes a feature known as a 'topic engine', which accesses a reference database of how words relate together within domains of discourse. This allows a search on 'cancer' to offer references to 'melanoma', even without an etymological link.
- There's a powerful commercial argument in favour of electronic document repositories based on searchable extracted indexes: they can handle a heterogeneous collection of file types without requiring them to be converted to a common format. So long as the software has filters that let it peek inside and unscramble the text content, however the creator software has encoded it, an index can be built. Thus a single index and search engine technology could cope with a document repository with content in Acrobat PDF, HTML, MS Word, FrameMaker, Lotus Notes and ASCII text files. (Viewing the content is another matter, of course.)

The intimate metadata of SGML

I cannot recall precisely how or when I first heard of the Standard Generalized Markup Language, SGML. I do know that in December 1991 I wrote a private 'ViewsLetter' for fellow users of the Manchester Host system, in which I recommended:

> SGML (also known as ISO 8879) is an emerging international method of 'marking up' a computer-resident document so that the structure of its distinct textual 'entities' (which could be headings, machine part numbers, Latin names of plants… you name it…) can be interpreted and expressed visually in different ways, depending on the software used to access it. I believe SGML could be the way to revolutionise our organization of (and access to) on-line databases…
>
> Let it suffice to say for now that once a computer file has been marked up, it can simultaneously be part of a free-text database, yet remains searchable by fields; alternatively, it can be viewed as a typographically fully formatted, automatically paginated, printable document.

To check out this opinion, in May 1992 I went to the SGML conference *International Markup '92,* organized in Amsterdam by the Graphic Communication Association. It felt like I was the only information designer present amidst 400–500 computer scientists… But I did return strengthened in my conviction that SGML markup has fantastic potential to let us add intelligent 'handles' to our information.

- SGML tagging converts a text file into a nested structure of entities which correspond to the logical parts of a document. It clearly marks the sections, the headings, the paragraphs, and significant elements within the paragraphs. If the document contains an Abstract, then it is explicitly marked as such. This means that you can search in the Abstracts alone; or, having discovered a relevant article by whatever kind of search, you can return the Titles and Abstracts and Authors to the enquirer.

- SGML permits you to choose among a variety of 'tag vocabularies' (tagsets, defined in a Document Type Definition, a DTD) – many have been created for various industrial and scholarly applications. Alternatively, you can create your own DTD, to be able to wield a tagset that lets you identify unambiguously whatever kinds of text entity are important to you.
- Although it is not a trivial task, it is possible to build software that will transform SGML-tagged text in various ways, re-arranging it and applying rules-based formatting to it for delivery via a variety of media types, be it e-mail, print, CD-ROM hypertext or whatever. (Indeed, there is now a formidably comprehensive way of specifying these transformations and formatting requirements in a software-independent way, through DSSSL, the Document Style Semantics and Specification Language, ISO/IEC 10179; I attended a workshop at the 1992 GCA conference at which Sharon Adler and Anders Berglund talked about the draft of this specification, which only became a full ISO standard in 1996.)

Following my return from the GCA event in 1992, at a conference of Manchester Host users, I ran a workshop on how SGML principles could transform on-line information. Which is ironic, because I was at that time quite unaware that, starting in 1989, Tim Berners-Lee had had similar ideas… ideas which became the World Wide Web!

Learning from the Web

When I look through the overhead transparencies which I prepared for that Manchester Host workshop in 1992, I am amazed how much my proposals looked like the World Wide Web. I suggested a client-server model, in which client software would be able to interpret simple embedded SGML mark-up into formatted text on screen. I also suggested on-line graphical forms, with radio buttons and pop-up menus, which on completion would send a marked-up stream of text as a query to a search engine on the host system.

The trick I missed, which is the trick that drove the spectacular growth of the Web, was *hypertext.* Mine was a client-server database model, with little concept of jumping from host to host, while Berners-Lee's model was of a decentralised *web* of data pages, the principal structure of which was caused by explicitly anchored hypertext links from one Web page to another with no central repository.

What have we learned from the World Wide Web? Well, we have learned that a worldwide information system can be built with incredible speed when there is a simple language to construct the files in (HTML), and when the browsers are capable, easy to use, produce fairly attractive results, and are cheap or free.

We have also learned about limitations caused by the 'statelessness' of HTTP, the Hypertext Transfer Protocol which underlies transactions on the Web. It's like asking the concierge of a hotel for help; after each answer he turns away, and when you tap him on the shoulder again, he's already forgotten who you are. We are also learning about the limitations of one-way hypertext linking; Web pages are too ignorant of their context, and only the author can fashion those links, which makes it difficult for knowledge workers to assemble their own link structures around other people's documents.

And we are straitjacketed by the limited tag vocabulary of HTML, which defeats efforts to build more intelligent indexes and contributes to the hopelessness of searching for information on the Web.

XML: Revenge of the forty-somethings?

We live in strange times. I remember standing at the microphone on the floor of the Seybold San Francisco conference several years ago, with the late Yuri Rubinsky of SoftQuad, giving John Warnock of Adobe a hard time for his apparent rejection of SGML.

This autumn John was the first keynote speaker at Seybold, singing the praises of SGML – especially a revised and slimmed-down version of SGML called the **Extensible Markup Language**, or **XML**. It's little wonder that Paul Trevithick of Bitstream called XML 'the revenge of the forty-somethings'. (You could almost hear the chorus of 'We told you so…' from the SGML crowd.)

In brief, here are some of the promising characteristics of XML (and the companion standards which are in development for it under the aegis of the World Wide Web Consortium):

- Like SGML (but unlike HTML), XML lets organizations create their own tagsets, so they can add structural markup relevant to their kind of information. (That's why XML is 'extensible'.) There is a proposal for a standard way of defining and registering these 'tag vocabularies', what the elements in it are for, and how they should normally be processed and displayed: it's called the Document Object Model (DOM).
- Compared to SGML, XML is highly prescriptive and restrictive in defining how tags should be distinguished from text, and it forbids the use of 'implied' (omitted) tags. This makes it much easier to build software to parse, process and display XML files; and the XML standard itself is easier to understand (it's about 45 pages only).

- This reform also means that it is possible to check whether an XML file is 'well-formed' (all elements having a proper start and end tag, for instance) without having to check its 'validity' (whether the tags used correspond to those in the Document Type Definition). This should make it easier to author XML files and produce them by conversion from other software formats.
- When SGML was developed, it did not include a well-developed model for linking and hypertext, but XML has learned from the best practices of HyTime, the SGML companion standard for time-based media and hypertext. A sophisticated linking system (XLL, or XML-Link) is being developed which will 'tell' XML documents where they belong in the context of other documents, will allow documents to be embedded inside each other by reference, and will allow multi-destination links. It should even be possible to build separate files of 'out-of-line links' between documents.
- A style and transformation language for XML is under development, dubbed 'XSL' for the Extensible Style Language. Having a transformation element that allows files to be converted in various ways before formatting and display, XSL is more powerful than Cascading Style Sheets for HTML, which handles formatting only. The concepts in XSL draw on concepts developed for DSSSL.
- XML's standard character set is defined as Unicode, ISO 10646, which opens it up to use for all the world's languages.

What will be interesting is to see how big software companies (some of whom have vested interests in their own formats, which they would like to see become *de facto* standards) will embrace XML; and what they see its role as being. My impression is that John Warnock, for instance, would like XML and SGML to be secret underlying enabling technologies, of which neither authors nor information users need ever be aware, with final product being delivered through on-the-fly formatting to *e.g.* Acrobat PDF.

Building the Infobase

At Seybold SF 1996, my friend Susan Cato, then the publishing manager of the Optical Society of America, spoke about the OSA's strategy for managing its information resources and making them available in various forms – on line and on paper – for their member organizations. At its heart was a determination to add a high level of structural markup to information using SGML.

Susan Cato dubbed the resulting repository as an 'Infobase', insisting that the information was not 'repurposed' from its original editing environment; it had always been 'depurposed' – that is, edited without prejudice towards any one particular form of use.

If the organization's culture permits this approach to the important knowledge assets of the enterprise, it's possible that having those knowledge assets in SGML or XML means that many detailed 'handles' get embedded within the information. And therefore:

- If inverted-tree indexes are built from these files, with the right technology it should be possible to cross-tabulate free text content *and* structure – for instance so that a repository-wide search for the text string 'Major' or 'Bear' could be restricted to the names of constellations only.
- There are many ways in which such 'handles' can be used to attach XML 'out-of-line' hypertext link mechanisms — especially if, in authoring, each tag is assigned an ID reference attribute.
- The structural mark-up, when associated with a stylesheet, will allow the files to be transformed appropriately for print or on-line delivery. Transformation could include hiding some sections or adapting to the character set of the target output; we are talking here of far more than just 'style' in the typographic sense.

Of objects, databases, brokers and clients

Another area where progress is taking place is in developing databases that are better suited to containing the complex document formats in which linguistically-based knowledge is most sensibly kept. As we have already seen, relational databases offer great benefits in the handling of fielded data, but it may be a new class of database, the *object* database, which is better suited to the storage of structured documents.

Essentially, we need database software which is fully aware of the structure of what it has been asked to contain, and can make queries within that structure. We also need to have a query language so that we can ask those document objects to be interrogated, and a layer of 'middleware' on the server side which can receive those queries, operate on the object database, find the results, and then post results back to the client software.

A recent issue of *Byte* magazine contained an article suggesting a mechanism for how this might be done. Client software (it could be a Web browser) would ask to interrogate the contents of a document repository. This interrogation might itself require special middleware on the client side. One way this could be provided is on a temporary basis, as a Java program downloaded from the information site, and therefore customised to interact with the server-side middleware.

Written in Java, the client-side interface would be capable of a much richer query preparation process than an HTML form could provide, and could do things like validation and help without having to make another round trip to the server. Once the query is complete, it can be sent off to the server-side middleware. Here too XML might find a role, because it provides a very rich model for bundling up transactions.

The model just outlined assumes broad public access to an information server using the kind of thing which would ordinarily today be recognised as a computer. But what I find extremely interesting is the prospect of information querying and information distribution through small handheld devices or embedded information artifacts which are not now recognised as computers, and perhaps never will be. The VCR which knows that the program it was asked to record has been delayed 15 minutes, the bus-stop which knows when the next bus is coming, the mobile phone which alerts you if the price of the Deutschmark has moved five pfennigs; who knows what will be possible in the future?

Capturing knowledge is the hard part

For many organizations who have made the move to capturing their knowledge base in a highly structured way, the technological challenge has indeed been considerable; but when the veterans of such campaigns gather to compare notes, you find that the biggest revolutions that were needed were cultural and political.

Consider the work of a knowledge worker such as a technical writer, a lawyer, a consulting engineer. People like these are the ones who create the linguistically-based knowledge of an organization and cast it into words. They have beencomfortable working in a situation in which the information products they were creating had to make sense only to humans. As WYSIWYG-style word processors came along, they adopted them gladly, adding formatting to headings and bullet points, from which another human could infer the structure. In fact, seeing the structure 'dressed up' also helped them as writers and editors.

When it comes to making document structure *logical and explicit* using a mark-up language, how will they react? Technical authors may acquiesce; other kinds of knowledge worker may say, 'That's not my job!' Yet who is better qualified to assign structure and meaning to text elements than the subject experts themselves? This is a challenge which organizations must address.

Representation, and interface design

Donald Norman recognises the gritty politics of this problem, and in his book *Things that make us smart* he discusses the role played in this by the new cognitive artifacts (*e.g.* editing software) which people must learn in the interests of efficiency. But — efficiency for whom?

> From a person's *personal* point of view, artifacts don't make us smarter or make us have better memories; they change the task. From the *system* point of view, the person + artifact is more powerful than either alone. Performance of the *system* of person + artifact is indeed enhanced, but that of the individual person is not…

> An artifact is not a simple aid. That is, you can't just go out and find some cognitive artifact, and there you are, better at something. Nope, most cognitive artifacts present you with yet another thing to be learned, another manual to be read, another course to be taken, or another period of slow, tedious learning to endure.
>
> …[W]hat is it that makes some artifacts effective, others not? Could we develop a science of artifact design that would tell us how to make better artifacts, perhaps ones that were easier to learn and use?

In English we call people 'Luddites' if they show a negative attitude to new technology. But the Luddite activists of 19th-century England did not destroy mechanised looms because they hated technology; they did it because they were suffering as a result of mechanisation. Similarly, if we expect knowledge workers to learn new working techniques and take on new responsibilities without any benefits to them, efforts to change will be resisted – perhaps even sabotaged.

It is encouraging to see improvements in the user interface of some of the latest structured editing tools, such as FrameMaker+SGML and ArborText Adept•Editor 7.0. Not only is an effort being made to ease the burden of structural mark-up and the addition of metadata; I can envisage writing teams becoming downright enthusiastic about some of the editing power that is fed back into the hands of writers – such as being able to grab hold of the structure of documents, and drag-and-drop whole sections into new locations.

So, what are our responsibilities?

- **Information managers** should take the lead, building an assessment of the role which knowledge plays in the organisation, who is in a position to create and codify it, who needs it, and in what form, and how it should be delivered, or made available for searching.

- **Information technologists** need to work with information managers to create the best fit between the needs of the organization and the various technologies and standards as they evolve. There needs to be a solution for handling data formats in which knowledge is contained at present (for instance, word-processing files, PDF, Web pages *etc.*), but also a rational long term strategy about the data formats to which the organization should migrate.

- **Information designers and technical authors** continue to be responsible for the appearance, wording and usability of the information products on which they work now, but should also be aware of what they can contribute in to the Brave New World of depurposed information and documents-on-demand. I promise you, if you leave it to the IT department, you won't like the result!

- And **publishing software companies**; their job is to provide us with better and easier to use tools. And perhaps our job is to help them to help us, by telling them what we need!