Contents

What is parallel publishing?

Not a generally-used term, but quite useful

A limited definition

Publishing the same thing in at least two different media at the same time

A practical definition

Edit once, publish many

or

Faster websites

A complete definition

The creation of

- two or more
- ‘ format instances’ or ‘ expressions’
- from a single data set
- [automagically]

Technically

Use of master data which

- is unambiguously structured
- uses an ‘ open standard’ for data description
- contains information about the meaning of the content
- does not contain information about the appearance or look of the content

such that

- separate rules about layout and appearance can be applied to suit different media or content, and
- documents can be intelligently searched, classified, linked and ordered

Why would anyone want to do it?

- To save money
- To save time
- To reduce errors
- To create presentations which cannot be made in conventional ways
- To create presentations based on ‘ slices’ of master data
- To create nervous breakdowns among authors, editors and designers

To save money

For example, with conventional methods;

Print & Web	design print	prepress costs
	design web	prelaunch costs
Print & Web & WAP	design print	prepress costs
	design web	prelaunch costs
	design WAP	prelaunch costs

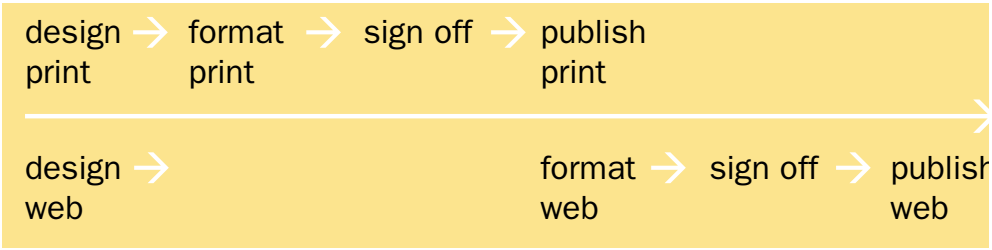
With parallel publishing methods:

Print & Web	design print	development and prepress
	design web	
Print & Web & WAP	design print	development and prepress
	design web	
	design WAP	

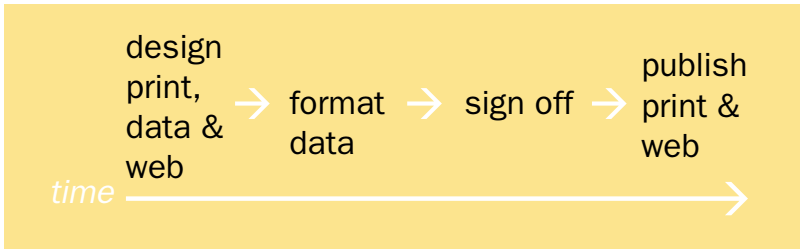
With parallel publishing, the more you do, the more you save.

To save time

the old way

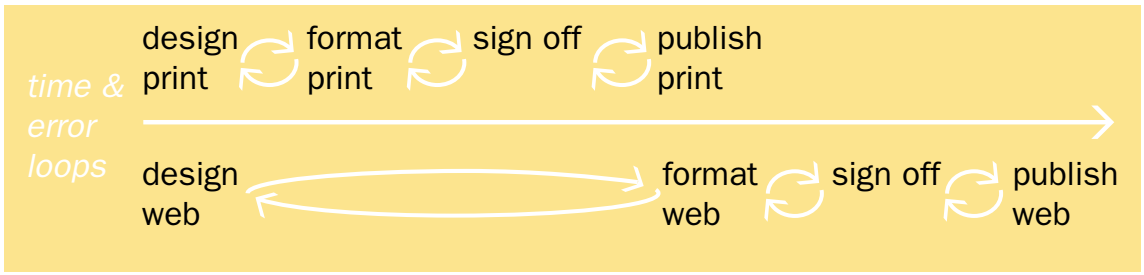


the new way

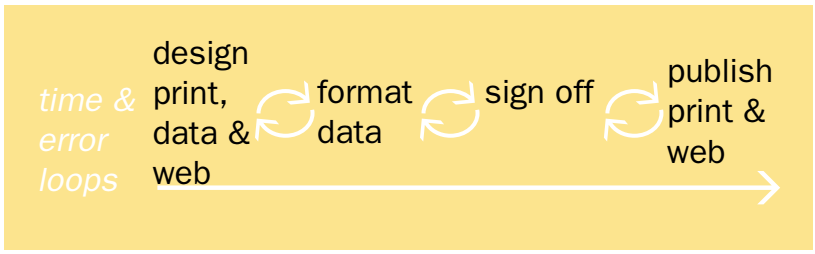


To reduce errors

the old way



the new way



To create presentations which cannot be made in conventional ways

To create presentations based on slices of master data

- summaries
- tailoring websites for particular segments of individuals
- user choice: order presentation by subject or by date

To create nervous breakdowns among authors, editors and designers

Not a joke:

- new methods
- new roles
- old roles performed in new ways

= 500 years of learning to cope in inherently stressful jobs is under threat

What kinds of projects?

Parallel publishing methods suit publishing projects which involve any of these:

- publishing in more than one medium
- ‘ instant’ publishing
- big information projects
- frequently-changed information
- many versions or personalisation required
- searchable and linkable results needed

Publishing in more than one medium

... but not everything that's dual-published. These methods suit

- journals – but (usually) not magazines
- annual reports – but not annual reviews
- information sheets – but not brochures
- a series of leaflets – but not a ' series' of one leaflet

– systematically-designed media

' Instant' publishing

... where you can prepare but then need to get it out fast

- Reports and enquiries
- Parliamentary reports
- Tax materials
- Results, prizes, honours lists
- Annual reports

Big information projects

Any large body of information

- Dictionaries
- Encyclopaedias
- Curricula
- Prospectuses
- Phone books
- Regulations

Frequent changes

... either many editions or a lot of faffing about before publication

- statistical returns and reports
- news releases
- consultation drafts of legislation, regulation, recommendation, etc
- phone books, address lists
- almost anything produced by any committee anywhere at any time

Many versions or personalisation required

... for different audiences, interaction methods, or media

- ' congratulations, Ms Smith, you have already been chosen'
- Tax/benefit/achievement forms and statements
- Learning materials/teacher editions

Searchable and linkable results needed

(requires tedious lecture about BLOBs)

The method suits sites heavy on metadata and intelligent linking.

A well-structured document is a smart document

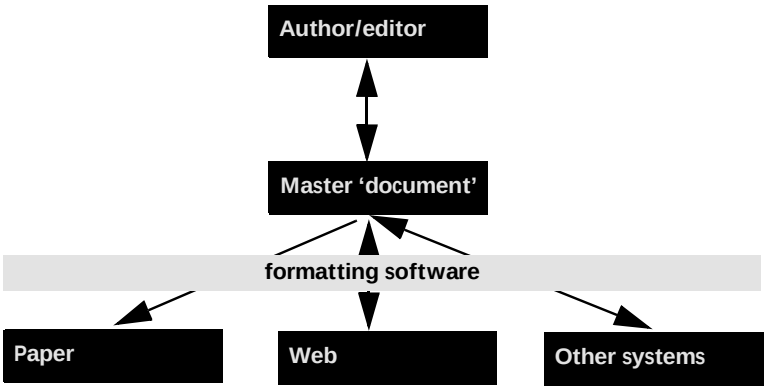
Summary

- Parallel publishing means automated publishing for web, print, and other media from a single master source
- It involves ' intelligent documents'
- It suits big/long projects

How parallel publishing systems work

- Conversion-based
- Document-centred
- Database-centred
- Object repositories

Common characteristics



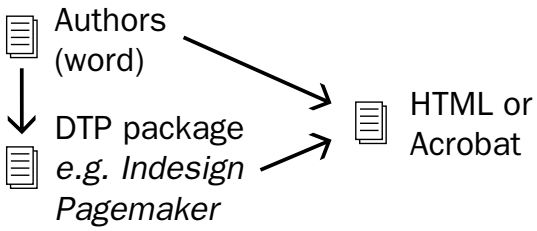
Conversion-based

Take a paper document and export it into a web-based presentation. 2 main ways:

1. put pictures of it on the Web
2. translate it into a more appropriate format

Put pictures of it on the web

- Acrobat
- InDesign
- MS Office – especially Word and Excel



Translate it into a more appropriately-formatted version

- InDesign
- PageMaker
- MS Office (via Web Publishing Wizards)

Plus points

- Where you can do it at all, it's easy
- Where authoring is tightly controlled, fast way of publishing less-considered information
- ' Save as web' ideal for Intranets

Minus points

- Visual organisation which suits paper doesn't suit web
- Longer documents are difficult
- No subtle control possible
- Can't transform documents to suit medium

Document-centred

Uses a ' design-neutral' intermediate or master ' language' such as XML or SGML.

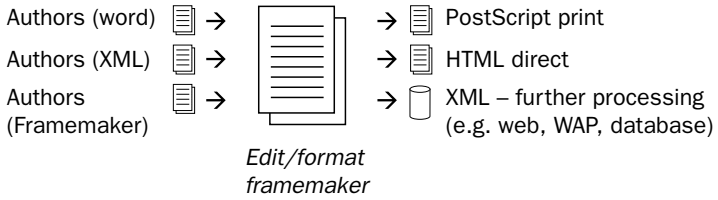
Generally behaves quite like DTP

Key differences are that the application

- separates layout from structure
- can check that a document conforms to external rules about the structure of a class of documents – specifically, to a formal Document Type Definition

Usually some WYSIWYG editing, e.g.

- Interleaf Quicksilver
- FrameMaker+SGML
- Ventura Publisher
- Adept Author/Publisher



Plus points

In production, it can look like a print document, so...

- familiar/reassuring to editors/typesetters
- can be used by authors
- proofing done conventionally
- some systems can create high-quality print

Minus points

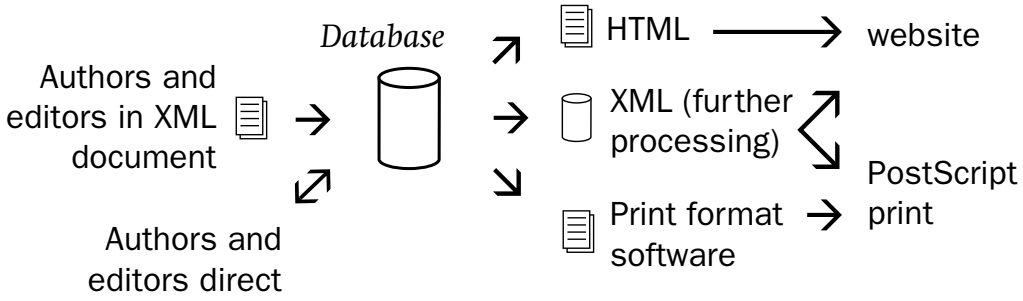
- usually needs other software to help make good websites
- limited ' selective output' for print
- authors need training
- version management/workflow tools expensive or geared to all-on-one-site publishing

Database-centred

Database used as the master format, surrounded by input (editing) and export (publishing) filters, or exporting to suitable publishing software

Oracle, Informix, IBM DB2 and others are relational database management systems (RDBMS) used in many web-publishing systems such as MediaSurface and StoryServer.

Lotus Notes/Domino is a ‘ flat’ database and web publishing system.



Plus points

- ideal for ‘ table-based’ information such as catalogues, price lists and statistics
- straightforward to build web-forms for authors/editors
- straightforward to export HTML and/or XML
- easy to make ‘ live transformations’ on data reflecting changes instantly
- well-understood by IS professionals

Minus points

- Not suited to ‘ document-like’ documents (!)
- Output to print often difficult: bespoke formatting software or separate print-publishing process may be required
- Remote authors need live web connection
- For most purposes, web-form authoring is unpleasant

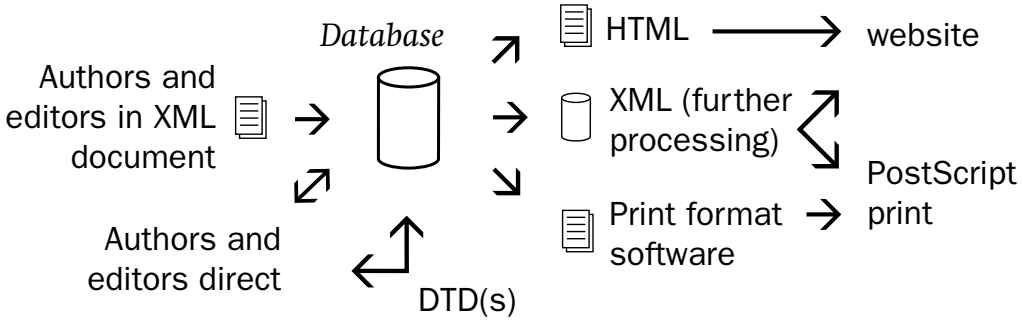
Object repositories

Object database used as the master format, surrounded by input (editing) and export (publishing) filters, or exporting to suitable publishing software.

Object databases deal with ‘ tree’ data structures as well as ‘ table’ data structures. This is useful in dealing with XML.

Normally contain ‘ workflow’ tools for revision control

Oracle IFS, Zope, Frontier, POET



Plus points

- Object databases are cool
- System design is simple – in theory
- Usually suited for workflow management
- Work well with ‘ document-oriented’ XML

Minus points

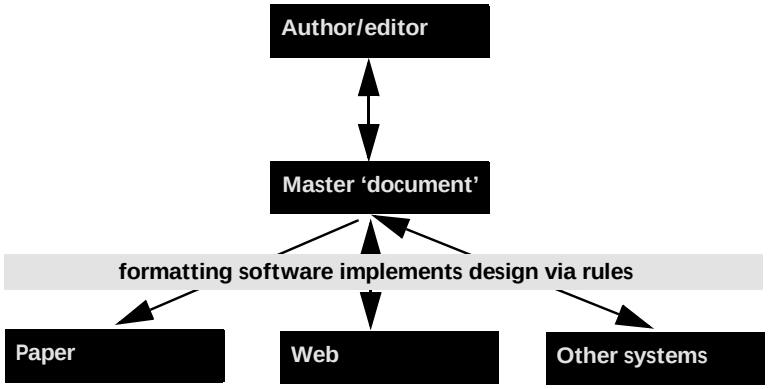
- Not many products to choose from
- Can be expensive
- Tools are scarce or young, therefore
- Implementing projects can be risky/expensive

Summary

There are lots of ways of doing parallel publishing. All of them are quite technical

How you separate content and design

Different methods and technologies, but the principles are the same:



Procedural markup

Typesetting, PostScript, most HTML coding:

<Times; Medium; size=9pt; weight=medium; linefeed=13pt>Here is some <bold=on>important<bold=off> news<lf>

- everything is procedural in the end
- specific to one ' rendering' situation. e.g. web, imagesetting

Stylesheets

Most word-processing and office document tools

Document markup	Stylesheet
<Newsflash>Here is some <bold=on>important <bold=off> news<lf>	Newsflash= Times; Medium; size=9pt; weight= medium; linefeed=13pt

Stylesheet information is applied to the document to render it

- allows change of appearance throughout document by changing stylesheet rather than every piece of text
- stylesheet usually part of document
- allows separation of ' content' from ' design' in theory, but not often in practice

Structural or generalised markup

Stylesheets plus enforcers

Document markup	Document description	Stylesheet/FOSI
Descriptname <Newsflash>Here is some <emph>important </emph> news</Newsflash>	Newsflash (TEXT | emph)* emph (TEXT)	Newsflash= para; Times; Medium; size=9pt; weight= medium; linefeed=13pt emph=character; bold
		Newsflash= H1 emph=red

- Stylesheet and description separate: used by many documents
- One document can use many stylesheets
- Enforced structure required for auto layout

Formatting in parallel publishing

- is done by software not by people
  (though real life is full of sordid compromises)
- requires documents to conform to an expected pattern

Design in parallel publishing

- is the creation of the rules that control formatting
- just as it always has been

Markup systems – xml and sgml

- What XML/SGML are – all you need to know about the technology
- How something so simple can be so powerful
- XML support in software
- XML support in EDI and business-to-business communication standards
- XML and information management in government
- when XML will be everywhere

Markup – all you need to know about the technology

eXtensible Markup Language – XML – a new standard based closely on SGML

Standard Generalised Markup Language – SGML – an ISO standard for describing information since 1986, based closely on IBM GML

HyperText Markup Language – HTML – is another child of SGML, conceived in passion and optimism but made bitter and twisted by a difficult high-pressure childhood. Now a surly and contrary adult.

Tags allow computers to recognise information

People mostly recognise kinds of information easily. We use

- context
- position
- size
- weight

Computers need help.

Every piece of information in a file which uses XML has an explicit tag.

‘ Markup’ is what it sounds like

Both XML and SGML use ‘ tags’ which surround information so that a computer can recognise what is between the tags.

This is an invoice total 2345.00 and that’s the end of the invoice total

One tag tells the computer where a piece of information starts, and the other tells it where that information ends.

Tag syntax

To separate tags from the information you need signs.

XML uses less-than and more-than signs – so `<total>` is an XML tag.

It also needs to know where the total begins and ends. XML repeats the tag at the end of the information with the addition of a slash to show that it is the end. So a complete tagged total might be expressed as

<total>£19.95</total>

Tag-and-content makes an **element**

Information about information

Tags can contain other information used by the computer but not necessarily appearing in a printed version of the document.

‘ Attributes’ are ‘ meta’ information, about anything you like. (‘ meta tags’ in web pages use attributes)

Attributes are usually used to help find and process information which doesn’t appear to the reader.

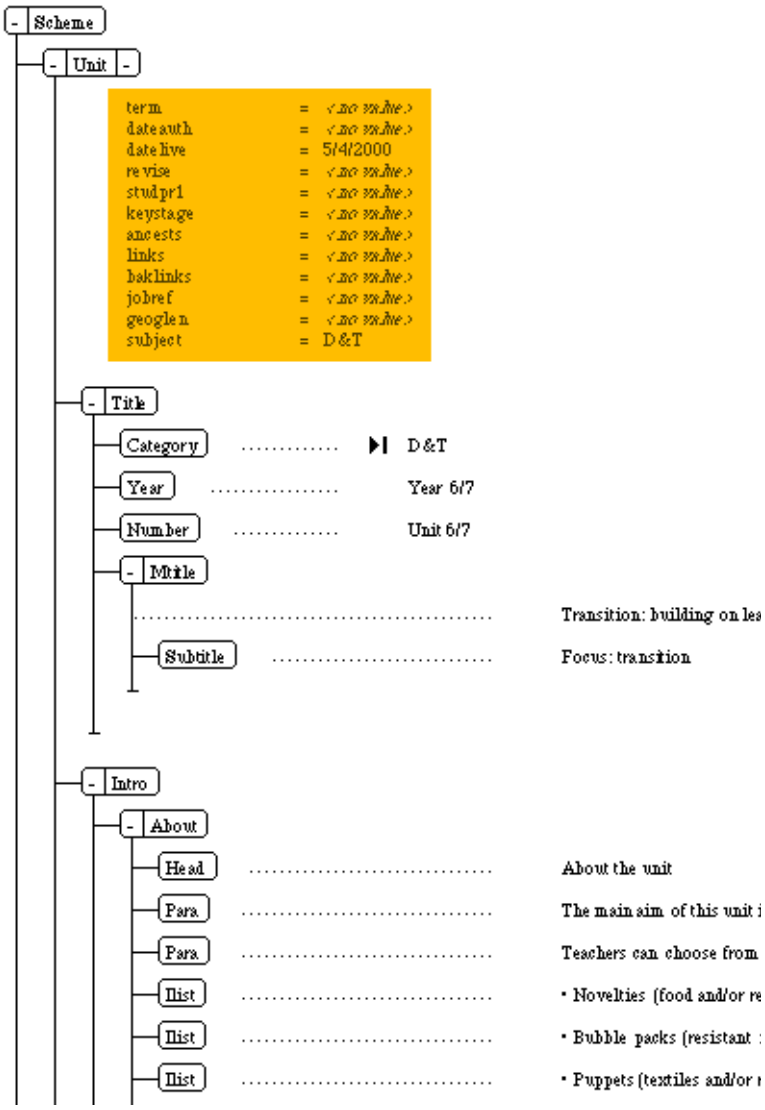
<total currency = “sterling” >19.95</total>

allows the computer to insert a £ sign in displays – or to do all calculations in Euros and translate to sterling during printing

Tags within tags

Tags can enclose

- text
- ‘ binary’ data - e.g., pictures, sound, software
- more tags



Contents lists – DTDs and schemas

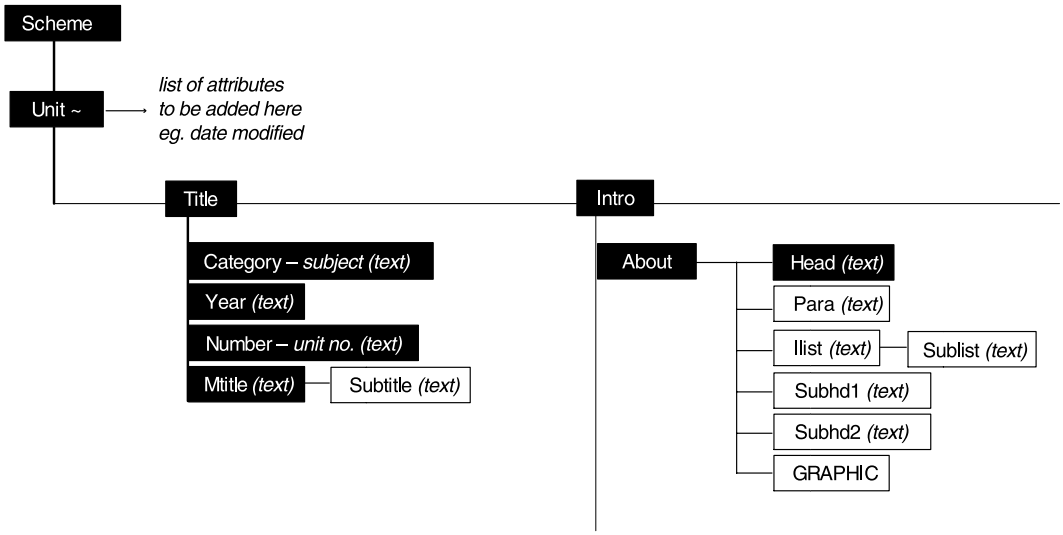
A computer needs to know what sort of information to expect in a document.

XML has rules for describing what tags can appear – and where they can appear – in any particular type of file.

An XML schema or DTD (document type definition) describes a class or type of document.

A schema called `invoice`, for example, might contain a list of the tags used in any invoice. The tags would probably include things such as `item`, `quantity`, `price`, `tax`, etc.

Most XML files are associated with a schema or DTD.



All the schemas you want

The XML standard does not define any particular schemas or DTDs – it just says how schemas should be written.

Some schemas are internal to a project or organisation

Others are industry- or world-wide.

UK government is committed to adopting standard schemas (none actually exist yet).

How something so simple can be so powerful

‘ Compatibility’ is the key to parallel publishing.

XML delivers compatibility – by creating a format for information files that is readable by any XML-aware software application.

Different applications may want to do different things with the same information

XML defines content not appearance

XML support in software

Most of the world’s software builders have committed to XML for file formats.

- Microsoft Internet Explorer 5 reads XML files. Microsoft is spending ‘ more than the Apollo programme’ in building XML-based ‘ Next-generation Windows services’.
- Quark Avenue supports XML [import and] export
- IBM has committed enthusiastically to XML
- Lotus supports XML in Notes/Domino
- Apple has built XML into its new operating system, OS X
- Oracle has committed to XML as a data format for its database products, and has delivered many XML components
- Sun is launching developer tools which integrate XML and its Java language seamlessly.
- Adobe supports XML in FrameMaker document processing system and plans support within its Acrobat document-transfer format.
- AOL’s forthcoming Navigator 6 browser will open and process XML files.

XML standards work – general

If you're making a single publication and website you only need standards within your project

If you may want to exchange information with other organisations or parts of your organisation, you need to agree some standards for naming different kinds of data.

This may mean having many tedious meetings

Or you may be able to adopt industry, national or international standards

Relevant standards and projects

- UKP.eb – UK government and industry consortium to develop best practice
- ebXML – A joint project between OASIS and the United Nations CEFACT to develop e-business XML standards.
- DocBook – long-standing standard for documents in defence and aerospace
- eBook – attempt to create a generic XML schema for paper and electronic books
- BizTalk – Microsoft-led clearing house for XML schemas

XML and information management in UK government

One of the ' Strategic building blocks' in the e-government strategy is information management, and its goals are:

' 30. The public sector needs to move towards managing the information it holds as a corporate resource to benefit citizens and business and to improve the effectiveness and efficiency of government itself. This will require

- definition and adoption by public sector bodies of corporate standards for data entities that are common across the public sector; for example, citizen name and address. CITU will publish these in the interoperability framework and will monitor their adoption
- definition and adoption by public sector bodies of common meanings for common data descriptions, so that information accessed by public servants or the public can be understood and used effectively
- a framework for departments and agencies to implement electronic records management systems
- a common policy on the use of metadata.'

– *e-government: a strategic framework for public services in the Information Age*

Interoperability

' There is [...] a strategic decision to adopt XML as the main standard for data integration. This strategy includes provision of XML schemas for use throughout the public sector.'

...

' Compliance to the UK GIF policies and standards are mandatory for government systems.'

– *e-government Interoperability Framework*

Progress on standards

The elements of standards, based on existing international standards such as EDIFACT exist.

In UK government a working group exists and will issue its first recommendations shortly for brief consultation.

A very few detailed standards exist – mostly for things like names and addresses.

Potential for standards

Huge.

Requires management and co-ordination

A lot of people are building standards now

By 2002 most kinds of data exchange will have XML standards

By (guess) 2004 most data exchange will use XML

Most web browsers will read XML instead of HTML

Summary

- XML is a language for describing information
- XML does not usually describe the layout or appearance of information
- Types or classes of information are described in DTDs or schemas
- UK government is committed to XML and says it is developing DTDs/schemas

Doing design for web and print

Designing the rules for formatting uses different

- mechanisms
- tools, and
- disciplines

from designing pages for print of ' pages' for web, but the underlying skills are unchanged

Ways of putting design in

The designer needs to be concerned with:

- document structure
- style rules

Of these, manipulating style rules is the key design skill

Document structure

Simple (fast, cheap) parallel publishing projects should ' map' the document content structure to intended layouts/formats.

Ideally, the designer should aim to

- get involved early in the project
- press for structure where
  - 1 tag = 1 style
  - the order of information stays the same
  - page breaks/web trees can be prompted by tags
- information which won't be visible is ' hidden' as attributes

Style rules

The main focus for design attention.

Three main kinds of rules:

- Application-specific rules
- Standards-based style rules
- Processing applications

Application-specific rules

Where the software which controls editing/formatting has its own rules about applying layout to structured content:

- FrameMaker+SGML
- Interleaf Quicksilver
- Quark Xpress+Avenue
- Microsoft Word

These applications generally

- export directly to print, web, and XML
- have their own ways of doing things
- can be used by the designer in WYSIWYG mode

Standards-based style rules

Rules for other software to use when formatting, e.g.:

- CSS – Cascading Style Sheet
- XSL – eXtensible Style Language
- XSLT – eXtensible Style Language and Transformation

These are specifications rather than bits of software. They

- need other bits of software to actually make anything
- are web-focused
- are clunky to use: modifications to a style take time to test

Processing applications

Custom software which processes a document entirely automatically and produces

- a web site
- PostScript ready to print.

Writing this software is not trivial. It means

- changes are hard to make
- testing is timeconsuming
- it all gets very expensive

Best avoided – though some processing applications are usually needed, especially when the specification keeps changing during the project. Perl and Python

How parallel publishing changes things for...

- Authors
- Editors
- Designers
- Printers
- Web administrators
- IS/IT Professionals

Authors

Authors don't have to work with structured documents but there are significant advantages if they do:

- Reduces costs of formatting dramatically
- All ' instances' in a series are made to the same standard
- Authors can include metadata
- Authors can engage in standards-making – making clear their needs to make better DTDs

Some authors love structured documents. Some hate them – generally because:

- The writing tools are unfamiliar or expensive (or both)
- The writing method doesn't suit them – it's more like filling in a form than writing an essay

Editors

Editors in structured-document workflows almost always work electronically. Changes to the way they work are in two places:

- Editing text
- Correcting proofs

Editing text

All existing editorial tasks

- sense
- grammar
- effectiveness
- spelling
- house style

exist, to be joined by

- ensuring the structure is used ' legally'
- ensuring that ' metadata' exists
- ensuring that links/cross-references are inserted

Correcting proofs

Very difficult for editors: all existing methods have drawbacks

- Working on paper: needs strong understanding of the structure and impact on other formats
- Working WYSIWYG on screen: generally see impact in only one format
- Working in ' forms' on screen: useless for everything except getting the text right.

Designers

Designing for parallel publishing is like designing for anything else:

- using the medium to communicate effectively and stylishly
- working within the production technology

the major differences are

- when design should happen – at or near the project start
- where design happens: in filters/rules
- may be unfamiliar output technology – web, print, WAP
- ' simultaneous' design may mean small tight teams

Printers

Currently, Quark Xpress or PageMaker or InDesign are unlikely to be the way printers (imagesetters) receive files from parallel publishing projects. More likely to be

- raw PostScript
- Acrobat (PDF)
- FrameMaker binary

Observation: imagesetting people have got used to working smoothly with Quark. In general they need to get better at dealing with ' high-end Acrobat workflows'. It's the future.

Web administrators

Most site administrators will enjoy dealing with parallel publishing output: it should arrive in the right state to be published easily. Keys to success are:

- get involved early with the team in designing the process
- specify clearly what you need – and what you'd like to have
- do not accept the need for ' hand-coding' – it should be done automatically

IS/IT Professionals

Understanding XML as a data-exchange medium is not the same as understanding XML as a document-exchange medium. IS professionals sometimes find it hard to understand this.

Parallel-publishing projects are IT-intensive, and are sometimes (often?) seen as IS-driven.

But the task is to present information to help people understand: the key activities are marketing, authoring and design and they must lead.

What products do this stuff? Are they any good?

Design and authoring software for XML-centred document design

Task requirements

New software for new ways of making documents:

1. Design and creation of DTDs and schemas: data structure
2. Creation of databases/repositories or other systems for managing ' canonical' sources
3. Creation of design ' templates' for paper and Internet (XSL and CSS) publishing
4. ' Authoring' – term from technical documentation – a combination of writing, editing, illustration and layout
5. Export to XML, PostScript, HTML, MS Word etc

Design and creation of DTDs and schemas

XML Authority

- Excellent, inexpensive tool for creating and managing DTDs and schemas. £400

Near & Far Designer

- Excellent, long-established visual tool for creating and managing DTDs and schemas. Relatively easy to learn. £800

Creation of databases/ repositories or other systems for managing ‘ canonical’ sources

Interleaf BladeRunner

- Complex enterprise-level asset-management tools. BladeRunner comprises a suite of tools for managing information assets, starting from a Word plug-in to help make structured texts through enterprise servers and repositories to implement e-commerce solutions. Native XML support throughout the workflow. Some print (PostScript) support. From $100,000

Vignette StoryServer

Oracle IFS

MediaSurface

Creation of design ‘ templates’ for paper and Internet (XSL and CSS) publishing

Adobe GoLIve and Macromedia Dreamweaver are both excellent web page design packages. Less than £300.
Either can successfully be used for

- fast prototype design of web pages to test layout and interaction ideas – before successful ideas are incorporated into XSL/CSS controls for production templates
- user-friendly editing of CSS (Cascading Style Sheet) files

FrameMaker+SGML and Interleaf Quicksilver also incorporate template-building features

‘ Authoring’

SoftQuad Xmetal

- Excellent, highly configurable, and relatively inexpensive XML editor. Look and feel approaches a word-processor. Very crude paper-publishing abilities. Setup requires Java, Visual Basic or C++ programming. £400

XML Instance

- A ‘ pure’ and very capable XML document editor. No layout abilities at all. Does not look or feel like a document editor or word-processor. $99

Stilo WebWriter

- Good XML editing tools. No design ability. £200

Arbortext Adept

- Excellent editors and design management tools. Includes Word plug-in for creating structured texts. Hard to learn. Weak print-formatting. Modules $1000–$20,000 (Adept Editor 8 $1300)

Adobe FrameMaker+SGML

- The ‘ +SGML’ variant of FrameMaker has many faults but is still probably the best structured layout and editing tool for parallel-publishing tasks available. Excellent editor. Best design-focused tool. XML support still partial – no XML import ability. Good Word import/export. £1200

Interleaf QuickSilver 2345

- Native XML editor with some WYSIWYG features. Some print (PostScript) support. Word import/export. This is the main competitor to FrameMaker. It has better XML support but less ability overall. £800 per year per seat (not sold outright).

Export to XML, PostScript, HTML, MS Word etc

Combination of tools already mentioned and/or

- FOSI-builders
- Perl/Python/Java languages used to process source into destination
- some web servers have inbuilt ability to turn XML into HTML ' on the fly'

Summary

- Quite a lot of software exists to help make structured documents for parallel publishing
- Most of it is unfamiliar to most designers, authors and editors
- A lot of it is new (may not work very well)