Document Name			
Document Title	Significant Properties Testing Report: Structured Text		
Work Package			
Author(s) & project role	Lynne Montague, Project Offer		
Date	12 February 2010	**Filename**	structuredtext-significantproperties-v11
Access	☑ Project and JISC internal		☑ General dissemination

Document History		
Version	**Date**	**Comments**
1.1	12 February 10	GK copied text into revised template
1.0	15 April 09	Final version of Significant properties of structured text report
0.3	23 March 09	First draft of Significant properties of structured text report
0.2	10 March 09	Draft 0.3 of Significant properties of email testing report by GK
0.1	10 January 09	First draft of structured text property list by GK

Contributors

The following people have made direct or indirect contribution to this report: Adrian Brown, Tim Gollins, Stephen Grace and Gareth Knight.

Intended Audience

This document is written for use by the InSPECT project team, the JISC community and those interested in digital preservation.

Table of Contents

Project Overview

Significant properties are those aspects of a digital record that must be preserved over time in order for the Information Object to remain accessible and meaningful. The InSPECT Project is funded by JISC to investigate methods for maintaining the authenticity of digital resources across digital environments and transformation processes. It has produced a framework for the analysis of significant properties and created a set of reports that outline its application to four object types – audio recordings, raster images, structured text and e-mail – that will contribute and advance strategies for the characterisation and maintenance of significant properties over time.

Purpose of the report

This report examines the notion of significant properties as it applies to structured text documents. It seeks to identify the significant properties of structured text that must be maintained by examining each of its constituent elements and analysing its designated function. It goes on to examine strategies that may be utilised to maintain access to structured text assets in the long-term. Finally, it outlines a set of experiments that were performed by the project team to identify and evaluate tools that may be utilised to convert significant properties from one form to another.

1. Introduction

1.1. Overview of structured text

Structured text is a term that can be used to describe a broad range of different types of content, encoded using a number of methods. It is electronic data that contains text, represented by alphabetic, numeric and punctuation characters, accompanied by information that indicates its description or appearance. The key characteristic that distinguishes structured and unstructured text is the presence of markup that provides additional information about the text. The concept of markup comes from the publishing industry where traditionally manuscripts were marked up using a language of instructions in order for the document to by typeset for printing[1]. With the global use of the Web, formats such as HTML and XML are perhaps the most widely known examples of structured text today but other examples would include source code and email messages.

Structured text may be created for two purposes:

- Presentation – Markup intended to describe the display of textual content. It may be used to infer the structure or layout of textual content, e.g. text rendered in bold or a large font may indicate a title or column heading and italicised information may indicate emphasis or particular display conventions, such as indicating the author of a work.

- Description – Markup intended to indicate the semantic meaning of text, but not the method in which the information may be utilised. It is an exercise for the software application or researcher to decide on the method with which markup is handled. For example, software may extract text that is encased in a <creator> for use in the creation of a coversheet, or may attribute different display characteristics (bold, italics).

Presentation and descriptive markup languages separate information into logical structures. However, the principle for defining categories of information differ – presentation markup is primarily intended to affect the visual representation of a page (e.g. text emphasis, page layout); descriptive markup separates information categories into the appropriate semantic meaning. A digital Record may contain presentational markup, descriptive markup, or a combination of both.

Many representation formats can be considered to be compound objects that are comprised of a primary Component and several associated secondary Components, e.g. images, sounds, etc. The Information Content contained in the compound object may be presented using a number of methods – through the primary Component in isolation; through a combination of the primary and one or more Secondary Component; or through the Secondary Component in isolation.

[1] http://en.wikipedia.org/wiki/Markup_language

1.2. Overview of Standards

Many technical specifications and standards relevant to the storage and preservation of the different types of structured text have been developed and it is not intended that this document provide a comprehensive look at all relevant ones but rather a brief look at some of the main ones.

The Text Encoding Initiative[2], or TEI, is a consortium which has developed a standard to guide the representation of texts in digital formats. It has developed a set of encoding guidelines for machine-readable texts, particularly in the fields of social sciences, linguistics and humanities. Other resources on related teaching projects, software and publications are also provided through the TEI website. An encoding scheme in a formal markup language is specified within the TEI guidelines.

The Data Document Initiative[3], or DDI, is an international body aiming to establish a standard to govern social- science-based technical documentation. Specifically, this initiative aims to enable the use of social science datasets through a standard, written in XML, which will govern the content, presentation, transport, and preservation of documentation for these datasets. The standard should encourage interoperability, richer content, multiple types of output from one codebook, online analysis of DDI documents and more precise searching.

The World Wide Web Consortium[4], or W3C, is a body that aims to develop and implement specifications, guidelines, software and tools in order to encourage technological interoperability on the Web. It also acts as a forum for the exchange of ideas on related areas such as commerce, communication and information with the aim of reaching a shared understanding of how these areas influence the Web. The core of their work revolves around the writing of technical specifications which define how particular technologies should be used and implemented. Once these have gained W3C consensus these become recommendations and are regarded as Web standards. Amongst these standards are specifications for the various versions of HTML and XHTML that we will be using for testing in this project.

1.3. Application of the Performance model

To determine the significant properties of a digital Record, a consistent, formal method of identifying the important aspects is required. The National Archives of Australia (2002) has developed a 'Performance Model'[5], which has been adopted by the InSPECT Project. The Performance model establishes the concept of the 'essence' of a digital record that contains the "characteristics that must be preserved for the record to maintain its meaning over time". The principle of the model is that the process of rendering the Information Object in a form that can be understood by a user requires some interaction between the underlying data object and interpretative software. The model is comprised of three components:

1. Source: the encoded data object that contains the text, still images, moving images, or other content for interpretation;
2. Process: the method in which the encoded data is interpreted, e.g. a software tool, an algorithm;
3. Performance: the recreation of the Information Object in a form that can be understood by the user.

The central premise of the Performance model is the distinction between the raw, uninterpreted data, defined as the Source, and the interpretation of the data as a Performance. Although this is a useful metaphor, its application for structured text documents will vary, as distinguished by the content type and the rendering method. During the analysis it was recognized that, when applied to certain types of structured text (e.g. XML documents that do not possess associated instructions on the preferred method of recreation), the Performance Model metaphor is unhelpful unless a distinction between the Source and Performance can be made. Many types of structured text may be 'performed' using several methods. The purpose of our analysis is to describe the performance of structured text in a particular environment. It does not, and indeed cannot, describe every type of performance that can be made of structured text. To illustrate, an XML-encoded text may be presented to the user as an RSS feed, processed and converted to an audio stream, and/or represented in several XHTML-compliant web pages that contain different types of information (figure 1). If a theatre Performance metaphor is applied, it may be compared to the recreation of a script by one or more actors in different theatre environments.

[2] http://www.tei-c.org/index.xml
[3] http://www.ddialliance.org/index.html
[4] http://www.w3.org/
[5] http://www.naa.gov.au/Images/An-approach-Green-Paper_tcm2-888.pdf

A structured text document is composed of markup that encapsulates fragments of text. Through the use of certain tags, the creator is able to specify the meaning of the text and how an interpreter should handle it. In isolation, the text and semantic markup located in an XML document contains the Information Content to be preserved. However, it does not indicate the method in which it has been, or should be, presented to the user. In order to record details of the performance, the digital archive must describe the rendering method that has been used and the relationship structure that is visually established. It was recognized that the importance of certain properties was relative to the performance method. For instance, presentation formats such as HTML may contain a diverse set of structured and unstructured information that possess complex, and often poorly defined inter-relationships.

1.4. Representation Information

The Reference Model for an Open Archival Information System (OAIS)[6], introduced the concept of representation information i.e.

> 'The information that maps a Data Object into more meaningful concepts. An example is the ASCII definition that describes how a sequence of bits (i.e., a Data Object) is mapped into a symbol.'

It is important at this stage to clarify the difference between the concepts of representation information and significant properties. To apply the performance model, the representation information is involved at the process stage in interpreting the source data object and rendering it as an information object or performance. The significant properties are the characteristics or essence of this information object or performance that need to be preserved over time, regardless of technological changes, to maintain its meaning. It is these significant properties that we are assessing in this project rather than the representation information used to interpret them.

[6] http://public.ccsds.org/publications/archive/650x0b1.pdf

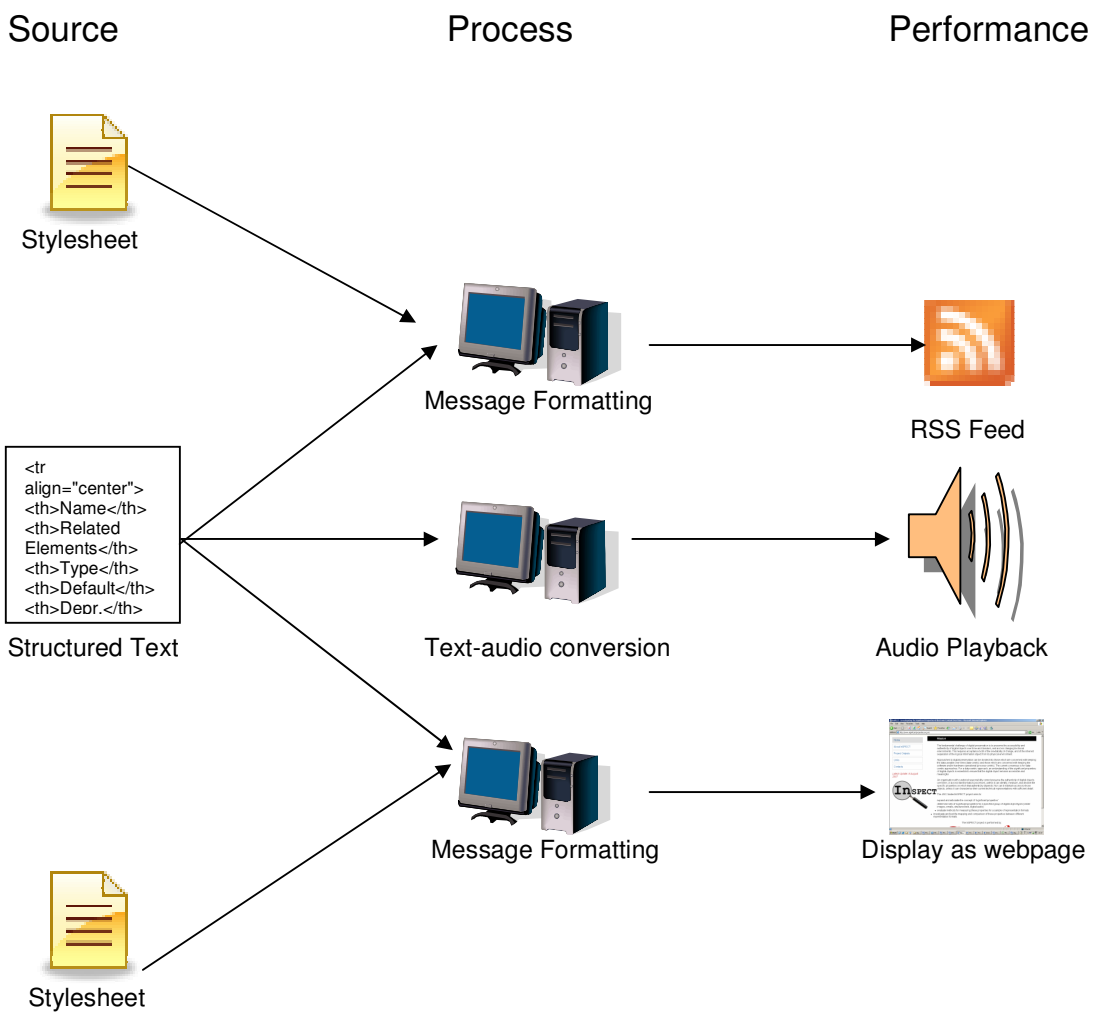


Figure 1. An example of the application of the Performance Model to structured text

2. Testing requirements

2.1. Significant properties that must be maintained

2.1.1 Introduction

The identification of properties of a digital object that are worthy of preservation is not a simple task that can be analysed based upon a set of universal rules. A set of rules defined for one category of digital object may prove to be too restrictive when applied to unusual variations, or inappropriate for other object types. Instead, the InSPECT Project team has developed a methodology to identify factors that establish the authenticity and integrity of the Information Object through a combined technical and epistemological approach.

During the process of investigating the creation, storage and use of digital objects it was found that the classification of significant properties was influenced by four key elements:

1. The form that the creator has chosen to express an intellectual or artistic idea and the method that they have used to communicate information
2. The function for which the digital object has been created to perform or the aims and objectives that its use will achieve.
3. The method in which information is encoded and stored in a digital environment, influenced by the encoding format and data standards in use.

4. The interpretation of the audience – the intended recipient of the digital object or an unknown future user – that is accessing the information to achieve an objective.

The challenge for the curator or archivist is to identify the characteristics of a digital object that enable them to fulfil the required function of maintaining the authenticity and integrity of that object throughout the preservation process. It is possible that that person will be able to answer some, but not all, of the questions needed to be asked. For example what information did the creator of a structured text object intend to communicate and who was the intended viewer of the object? For example, it may be that the structural layout, colour or font of the text within the markup is not important to the creator as long as the main meaning of the text is conveyed. However, in some circumstances it is possible that characteristics such as layout, colour and font are specifically picked as an artistic choice and that these are a vitally important, inherent part of the meaning of the object for the creator.

An item of structured text that is the target of analysis is unlikely to contain all the necessary information to answer these questions, unless extensive metadata is received with it.

2.1.2 Assessment of Significant Properties

To develop a list of the properties that may be significant for establishing the authenticity and integrity of structured text, the evaluator reviewed several specifications and standards that are widely used for the storage and description of structured text. The assessment of the significant properties of structured text in this document is based primarily on the analysis of the latest W3C HTML 4.01 specification[7] as it is felt that this is the most comprehensive standard which adequately specifies the types of generic characteristics found in structured text. Both the elements and attributes within the standard were reviewed. The list of significant properties defined in this section is not intended to be definitive as the number of attributes and elements that can conceivably be included in a structured text document are limitless due to extensible formats such as XML. Rather this section is an indication of the major types of characteristics found in structured text objects of the sort within the test samples, and an illustration of how they are regarded in terms of significance.

2.1.2.1 Parameters of project

For the purpose of analysis, this project examines the requirements of structured text containing a mix of presentation & semantic markup. It considers the preservation requirements of compound objects that consist of textual information (Primary Component), and a combination of textual and other information (Primary and Secondary Component). The third method of presenting the performance, as detailed above, may include a range of additional factors, dependent on the type of information contained in the Secondary Component, so is considered to be out of scope. This document will include some consideration of HTML and XHTML-based markup. It does not include a discussion of binary text documents that, although broadly similar, have other preservation requirements that must be considered. It also excludes an analysis of structured text files that contain dynamic content that may change, based on interaction with the user.

2.1.2.2 W3C HTML 4.01 specification

2.1.2.2.1 Document Header

'The HEAD element contains information about the current document, such as its title, keywords that may be useful to search engines, and other data that is not considered document content. User agents do not generally render elements that appear in the HEAD as content. They may, however, make information in the HEAD available to users through other mechanisms.'[8]

Name	Description	Element	Significant for preservation?
Character encoding	The standard and version number to which the	<meta http-equiv="content-type" content="text/html;	No This may be considered a type of Representation

[7] http://www.w3.org/TR/html401/cover.html#minitoc

[8] http://www.w3.org/TR/html401/struct/global.html#edef-BODY

	document conforms	charset=ISO-8859-5">	Information rather than a significant property (see 1.4 above).
Title	The title is a property of the document that may be used by a creator to provide a short description of the page content[9]	<title></title>	Yes
Creator	A meta element that enables an author to specify one or more creators.	<meta name="author" lang="eng" content="Gareth Knight">	Yes
Date	A meta element that indicates the date of creation and modification.	<meta name="date" content="2009-01-05T08:48:39+00:00	Yes It helps establish the provenance of a message
Keywords	A meta element used to specify keywords associated with the document	<meta name="keywords" lang="en" content="significant properties, representation information">	Yes. If used correctly to indicate key terms associated with the document topic, keywords may be useful for location and retrieval.
Rights	A meta element that indicates the copyright status of the document.	<meta name="copyright" content="&copy' 2009 Gareth Knight">	Yes. Establishes the rights holder(s) of the intellectual content and layout.

2.1.2.2.2 Document Body

'The body of a document contains the document's content. The content may be presented by a user agent in a variety of ways. For example, for visual browsers, you can think of the body as a canvas where the content appears: text, images, colors, graphics, etc. For audio user agents, the same content may be spoken.'[10]

Name	Description	Element	Significant for preservation
Body background	Attribute that specifies the page background to be displayed – an image or colour	Background = URI Bgcolor	Yes in certain circumstances Although the background may be utilised as a constituent component in creating the identity of the web resource, it is considered unlikely (except in a limited number of examples) that the background display will have a direct contribution to the intellectual content of the document. However, there are instances where it may be considered part of the intellectual content and would be significant. An example would be a 'draft' stamp, or an artistic decision about the type or colour of background used.

[9] See http://www.w3.org/Provider/Style/TITLE.html for further information.
[10] http://www.w3.org/TR/html401/struct/global.html#h-7.5

Body text colour	Attribute that specifies the foreground colour for text on the page	Text=[colour]	Yes in certain circumstances. The significance of the text colour is ambiguous and may vary between research disciplines. Web Accessibility Initiative guidelines specify that information should not be communicated through colour alone for accessibility purposes. However, it is recognised that many authors use colour artistically and to convey meaning e.g. using red to indicate a negative number.
Body link	Attribute that specifies the colour of text marking unvisited hypertext links (for visual browsers)	Link=[colour]	Yes in certain circumstances The colour of a hypertext link is not considered to be significant aspect of the intellectual content. However, it is recognized that it may have an artistic role for a small number of documents as above.
vlink	Attribute that sets the colour of text marking visited hypertext links.	Vlink = [colour]	Yes in certain circumstances The colour of a hypertext link is not considered to be significant aspect of the intellectual content. However, it is recognized that it may have an artistic role for a small number of documents as above.
alink	Attribute that sets the colour of text marking hyperlink text when visited by the user	Alink=[colour]	Yes in certain circumstances The colour of a hypertext link is not considered to be significant aspect of the intellectual content. However, it is recognized that it may have an artistic role for a small number of documents as above.
Div	A block-element in a page that indicates a division or section. It is a generic language/style container		Yes Communicates the context and structure of the text for interpretation by a reader
Span	An inline element in a page that indicates a division or section.		Yes Communicates the context and structure of the text for interpretation by a reader

2.1.2.2.3. Text markup

Name	**Description**	**Element**	**Significant for preservation**
Language	This is a language code that identifies a natural language which is spoken, written, or used for communication of information among people in another		Yes This communicates the language of the text for interpretation by a reader

	manner. These codes do not include computer languages		
Paragraph	The enclosed text indicates a linear set of text that is distinct from other paragraphs		Yes Communicates the context of the text for interpretation by a reader
Line break	A character that indicates the end of a line. Text that appears after the line break will appear on a new line	 	Yes Communicates the context of the text for interpretation by a reader
Preformatted text	The enclosed text is "preformatted" – white space remains intact and word wrap is disabled.	<pre></pre>	No The significance of preformatted text is ambiguous. The tag is a characteristic of HTML that is often used to define a specific text layout. However, it does not perform a function that is distinct from other types of text markup in the sample web resources analysed. It is recommend that preformatted text is examined and character encoding line breaks are converted to markup line breaks
Headings 1-6	A heading element may be used to indicate the logical internal structure of a document.	 	Yes Communicates the context of the text for interpretation by a reader
Emphasis	Indicates key words within the document		Yes Its significance is ambiguous – it may be utilised to indicate key concepts in the document text or for presentational purposes. The InSPECT team have taken the former viewpoint.
Bold	The enclosed text is formatted in bold		Yes Its significance is ambiguous – it may be utilised to indicate key concepts in the document text or for presentational purposes. The InSPECT team have taken the former viewpoint.
Italics	The enclosed text is formatted in italics		Yes Its significance is ambiguous – it may be utilised to indicate key concepts in the document text or for presentational purposes. The InSPECT team have taken the former viewpoint.
Centre	The enclosed text is formatted to the centre of the page.	 <DIV align=center>	Yes in certain circumstances Its significance is ambiguous and will vary from case-to-case. Text centring is commonly used for presentational purposes only and the risk of affecting the intellectual interpretation of the content is low. However, it may have significance for artistic works.
Underline	The enclosed text is underlined	<u></u>	Yes Its significance is ambiguous – it may be utilised to indicate key concepts in the document text or

			for presentational purposes. The InSPECT team have taken the former viewpoint.
Strong emphasis			Yes Its significance is ambiguous – it may be utilised to indicate key concepts in the document text or for presentational purposes. The InSPECT team have taken the former viewpoint.
Strikethrough	The enclosed text is struck through. The markup may be used to visually indicate that information has been deleted or modified.	<s></s>	Yes
Font	Defines the font in which text should be displayed, the size and colour	 	Yes in certain circumstances The font is not considered to be an essential element of a web page. However, it may be important for published papers and other documentation. It could in some contexts convey meaning or artistic intent, or may be a conscious decision made by a web designer for e.g. ease of use.[11]
Horizontal Rule	A horizontal line that is visually rendered on the screen[12].	<hr>	Yes The <hr> tag may be used by authors to provide a visual distinction between information as with line break above.
Inserted text	Denotes that the enclosed text has been inserted as a modification of an earlier version	<ins> </ins>	Yes The <ins> may be useful as a primitive form of version control in HTML documents.
Deleted text	Denotes that the HTML document has been modified and the enclosed text has been deleted from an earlier version[13]	<del> </del>	Yes The <del> may be useful as a primitive form of version control in HTML documents.
Samp	Denotes sample output, such as from a program or script.	<samp> </samp>	Yes This communicates the purpose of the text for interpretation by a reader
Cite	Denotes a citation or a reference to a source[14]	<cite></cite>	Yes This communicates the purpose of the text for interpretation by a reader
Dfn	The defining instance of an enclosed term	<dfn></dfn>	Yes This communicates the purpose of the text for interpretation by a reader
Code	Indicates that enclosed text is	<code></code>	Yes

[11] For example see http://webdesign.about.com/od/fonts/a/aa080204.htm.
[12] http://www.w3.org/TR/html401/present/graphics.html#edef-HR
[13] http://www.w3.org/TR/html401/struct/text.html#edef-del
[14] http://www.w3.org/TR/html401/struct/text.html#h-9.2.1

	software code		This communicates the purpose of the text for interpretation by a reader
Keyboard	Indicates text to be entered by the user	<kbd>	No Structured text files that contain dynamic content that may change, based on interaction with the user, are outside the scope of the project.
Abbreviation	Indicates that enclosed text is an abbreviation	<abbr>	Yes This communicates the purpose of the text for interpretation by a reader
Acronym	Indicates the enclosed text is an acronym.	<acronym>	Yes This communicates the purpose of the text for interpretation by a reader
Quotations	The enclosed text is a quotation	<q> (short quotations) <blockquote> (long quotations)	Yes This communicates the purpose of the text for interpretation by a reader
Subscript / Superscript	The enclosed text is displayed smaller than other text and is displayed slightly below or above it.	 	Yes This communicates the purpose of the text for interpretation by a reader
Address	Denotes contact information for the page creator or other contact[15]	<address> </address>	Yes This communicates the purpose of the text for interpretation by a reader
Button	Inserts a push button	<BUTTON name="submit" value="submit" type="submit"> Send<IMG src="/icons/wow.gif" alt="wow"></BUTTON>	Yes This communicates the purpose of the text for interpretation by a reader

2.1.2.2.4 Table and List elements

Name	Description	Element	Significant for preservation
Unordered list	A list of items that may be interpreted in any order but which shares a common basis.		Yes This communicates the purpose of the text for interpretation by a reader
Ordered list	A list of items that must be displayed in a pre-defined order.		Yes This communicates the purpose of the text for interpretation by a reader
List item	An distinct item in a list		Yes This communicates the purpose of the text for interpretation by a reader
Definition List	A list that consists of two parts: a term	<dl> <dt></dt> </dd></dd>	Yes This communicates the purpose of the text for

[15] http://www.w3.org/TR/html401/struct/global.html#h-7.5.6

	and description	</dl>	interpretation by a reader
Table caption	A short description of the table's purpose[16]	<caption> </caption>	Yes Indicates the purpose of the table which may be useful for interpretation.
Table caption alignment	Attribute that specifies the position of the caption with respect to the table	Align=top|bottom | left | right	Yes in certain circumstances May be an artistic decision as with centring above
Table summary	The purpose or structure of the table		Yes Indicates the purpose of the table which may be useful for interpretation.
Table directionality[17]	The direction of text displayed in the table. The default is left-to-right.	<table dir=""> </table>	Yes in certain circumstances Although it may assist the aesthetic appearance of the table, the text direction should not affect its underlying meaning. However, in some circumstances it may be an artistic decision as with centring above
Table Border	The visual appearance of a border that appears around a table, including colour and size.	<table border>	Yes in certain circumstances Although it may assist the aesthetic appearance of the table, the text direction should not affect its underlying meaning. However, in some circumstances it may be an artistic decision as with centring above
Table width	The visual width of a table		Yes in certain circumstances Although it may assist the aesthetic appearance of the table, the text direction should not affect its underlying meaning. However, in some circumstances it may be an artistic decision as with centring above
Table row	Rows convey structural information	<tr></tr>	Yes Indicates the logical structure of the information contained in the table.
Table Headers	Table headers communicate information about the cell that may be useful for visual or non-visual representation[18].	<th></th> <thead></thead>	Yes
Table footer	Table footers communicate information about the cell that may be useful for	<tfoot>	Yes

[16] http://www.w3.org/TR/html401/struct/tables.html
[17] http://www.w3.org/TR/html401/struct/tables.html
[18] http://www.w3.org/TR/html401/struct/tables.html

	visual or non-visual representation		
Cell Spacing	An attribute that indicates the spacing between cells	Cellspacing = length	Yes in certain circumstances Although it may assist the aesthetic appearance of the table, the cell spacing should not affect its underlying meaning. However, in some circumstances it may be an artistic decision as with centring above
Table Cell padding	An attribute that indicates the spacing within cells	Cellpadding = length	Yes in certain circumstances Although it may assist the aesthetic appearance of the table, the cell padding should not affect its underlying meaning. However, in some circumstances it may be an artistic decision as with centring above
Table cell scope	The set of data cells for which the header cell provides header information.	Scope	Yes
Table cell abbreviation	An abbreviated form of the cell's contents.	abbr	Yes Provides contextual information that may be useful for screen-readers
Table cell axis	Comma-separated list of related headers	axis	Yes
Table row span	The number of rows spanned by the cell	rowspan	Yes Communicates information on how the information contained in the cells inter-relate.
Table column span	The number of columns spanned by the cell	colspan	Yes Communicates information on how the information contained in the cells inter-relate.
Table cell wrapping	A Boolean attribute that indicates that the cell should not be wrapped when visually rendered.	nowrap	Yes in certain circumstances Although it may assist the aesthetic appearance of the table, the cell wrapping should not affect its underlying meaning. However, in some circumstances it may be an artistic decision as with centring above
Table cell width	An attribute that indicates the recommended cell width	width	Yes in certain circumstances Although it may assist the aesthetic appearance of the table, the cell width should not affect its underlying meaning. However, in some circumstances it may be an artistic decision as with centring above
Table cell height	An attribute that indicates the recommended cell height	height	Yes in certain circumstances Although it may assist the aesthetic appearance of the table, the cell height should not affect its underlying meaning. However, in some circumstances it may be an artistic decision as with centring above

Table cell alignment	The alignment and justification of text in a cell	Align (horizontal) Valign (vertical)	Yes in certain circumstances Although it may assist the aesthetic appearance of the table, the alignment should not affect its underlying meaning. However, in some circumstances it may be an artistic decision as with centring above
Table ID	A document-wide identifier		Yes May be useful for maintaining internal navigation
Table lang	Language		Yes
Table: Column Group	An explicit group of two or more columns	<colgroup> </colgroup>	Yes This communicates the purpose of the text for interpretation by a reader

2.1.2.2.5 Relationship

Name	Description	Element	Significant for preservation
Image	The element displays a referenced image at the location specified in the document.[19]	<IMG src="sitemap.gif" alt="HP Labs Site Map" longdesc="sitemap.html" >	Yes Indicates the relationship between a document and associated objects to be displayed in-line.
Link	A 'media independent' link found in the header that denotes relationships between one or more pages	<LINK rel="Next" href="Chapter3.html">	Yes The link may be significant, if the page is one of several pages that are held by the archive. However, it may be insignificant if the page is stand-alone.
Applet	The element displays a referenced image at the location specified in the document.[20]	<APPLET code="Bubbles.class" width="500" height="500"> Java applet that draws animated bubbles. </APPLET>	Yes Indicates the relationship between a document and associated objects to be displayed in-line.

2.1.2.2.6 Frames

Name	Description	Element	Significant for preservation
Frame	The element defines the contents and appearance of a single frame or subwindow	<FRAME src="contents_of_frame 1.html">	Yes Communicates the context and structure of the text for interpretation by a reader
Frameset	The element specifies the layout of the main user window in terms of rectangular	<FRAMESET> </FRAMESET>	Yes Communicates the context and structure of the text for interpretation by a reader

[19] http://www.w3.org/TR/html401/struct/objects.html#edef-IMG
[20] http://www.w3.org/TR/html401/struct/objects.html#edef-IMG

	subspaces.		

2.1.3 Summary

The suggested list of significant properties of structured text that need to be maintained, within the scope and definition of the InSPECT project is:

1. Title
2. Creator
3. Date
4. Keywords
5. Rights
6. Div
7. Span
8. Language
9. Paragraph
10. Line break
11. Headings
12. Emphasis
13. Bold
14. Italics
15. Underline
16. Strong emphasis
17. Strikethrough
18. Horizontal Rule
19. Inserted text
20. Deleted text
21. Samp
22. Cite
23. Defined Terms (DFN)
24. Code
25. Abbreviation
26. Acronym
27. Quotations
28. Subscript / Superscript
29. Address
30. Button
31. List Elements
32. Table Elements
33. Image
34. Link
35. Applet
36. Frame
37. Frameset

3. Methodology

3.1. Measurement Challenges

The identification and recording of the characters and markup in the Record itself is an effective language-independent method of measuring the significant properties of a digital Record. However, two problems may be identified that limit the assessor's ability to gain a detailed understanding of the property:

1. *Malformed tags* - Malformed tags are one of many common errors found in structural text, particularly HTML files, that may affect the assessor's ability to measure the document structure. The term refers to the incorrect expression of opening or closing tags in a file, e.g. an opening paragraph tag is defined, but the closing tag is missing, or tags are improperly nested (e.g.). This may present problems when attempting to record the document structure.

2. *Special characters* – Many character encodings and markup languages reserve certain characters for use in particular circumstances and specify that any other use in a text document is prohibited. Common examples include left (<) and right (>) brackets, ampersands (&) that are used for the definition of HTML elements.

However, there is often an alternative method of expressing the character that can be rendered, e.g. < for left bracket, & for ampersand, etc. Although the representation of such characters is not an issue, they present problems if the digital archive is measuring the success of a file conversion by counting the number of characters contained in the Record.

The value of measurements extracted from structured text in their submitted format may be questioned if it is likely that the Record is affected by the issues identified. A software application may misinterpret the relational structure of the document, or miscount the characters. The digital archive may be required to normalize the content prior to the creation of a canonical list and the measurement of the Record properties. Software code[5] exists to correct the majority of malformed tags. However, the process is automated and may change the rendering of certain characteristics. Similarly, special characters may be normalized to reduce the likelihood that anomalies will occur. The W3C has developed the 'Canonical XML' standard that may serve as a method to reduce the complexity of a Record, by reformatting text content. By normalizing an XML document, the encoding method is changed, white space is removed, default attribute values are added, special characters are reformatted to systemlegal characters, and comments are stripped.

For the purposes of this project, this normalisation process was not undertaken before characterising the test samples in order to highlight some of the difficulties that do occur during the process.

3.2. Representation Formats

Representation format is a general term that describes the method in which information is stored. In its abstract form, a representation format may be applied to many types of information. Restrictions on the type and extent of information are imposed when handling representation formats intended for a specific purpose. To provide a simple example, a representation format for image data is unlikely to be able to contain audio. Limitations may be imposed, even if information is stored in a representation format of the correct type. Specific properties of the information content may be degraded or removed when it is stored in a representation format.

3.2.1 Common representation formats

There are hundreds of different types of markup languages. In fact the number is unlimited because due to the extensible nature of XML, specific XML languages are being developed all the time. This section aims to give a brief overview the most widely used; HTML, XHTML and XML. [21]

- **Hypertext Markup Language (HTML):** HTML, which is based on the markup language SGML, is the universally understood, principal markup language used for publishing on the

[21] http://www.w3.org/TR/html4/intro/intro.html

Web. HTML documents have a structure containing a HEAD section with a title and information about the document contained within in, and a BODY section which contains the content of the document. The basic building block of an HTML document is the 'element' which can be structural or presentational. Elements usually have a starting tag containing the element's name, an ending tag which begins with a forward slash, and some content in between e.g.

<element-name>content</element-name>

However, there are exceptions to this format with some elements not needing a starting tag and some elements being empty and thus not needing an end tag. An element can also have an attribute with a value within its starting tag e.g.

<element-name title="Hypertext Markup Language">

A Document Type Definition or DTD will reference, in computer-readable language, the formal specification that applies to a HTML document i.e. the syntax and grammar of the HTML allowed in a particular document. The DTD is used to state whether the HTML document is valid i.e. conforms to the permitted content allowed by the DTD. Within the HTML 4.01 specification there are 3 DTDs, strict, transitional and frameset, which support different elements. The strict declaration includes all elements and attributes that have not been deprecated (outdated) or are not in the frameset definition; the transitional declaration includes all elements and attributes, including those that have been deprecated; the frameset declaration includes everything in the transitional one plus frames. Most of the elements allowed in transitional but not in strict relate to presentational elements. This is to encourage the separation of the presentation from the main document and into a separate style sheet in strict HTML 4.01 and is why many presentation elements have been deprecated. Whilst deprecated elements are still supported currently, they may become obsolete in later versions of HTML.

- **Extensible Markup Language (XML):** XML is a W3C recommended markup language designed to allow the software- and hardware-independent sharing of data. Generally, information about how to display data within an XML document will not be found within the document itself but rather, within a separate, referenced style sheet.

  Although XML data is written and stored in plain text, it is recognisable by many different types of application which means that data can be shared by incompatible systems. Unlike with HTML, tags are not predefined with XML and must be created by the user. This has lead to many new XML-based languages being developed to deal with specific types of data, for example, the TEI Encoding Language. In addition to developing guidelines for the representation of text in digital forms, the TEI has developed a specific encoding scheme in a formal markup language. In its latest version this uses XML syntax with almost 500 elements in order to be able to adequately encode documents from any time period or in any language.

  The concepts of well-formedness and validity apply to both HTML and XML documents. A document is well-formed if complies with the syntax rules of the particular specification. For example, tags should be properly nested, i.e. closed in the correct order, in both HTML and XML for the document to be regarded as well-formed. However, if this doesn't happen in an HTML document most HTML browsers will be very forgiving and display the document anyway whereas an XML application will reject the document in these circumstances.

  For HTML or XML documents to be declared valid they should comply with the relevant DTD (or schema in the case of XML). Again, for HTML, an invalid document will still be readable by a browser whereas an XML document will not be displayed by a browser if regarded as invalid.

- **Extensible Hypertext Markup Language (XHTML):** XHTML is HTML reformulated in XML in order to obtain interoperability between HTML and other XML languages, enable the use of XML tools and increase functionality. It conforms to the XML syntax and like HTML, XHTML 1.0 has strict, transitional and framesets versions.

For this project, HTML 3.2, HTML 4.1 and XHTML 1.0 were the formats chosen for testing as these are all supported by the JHOVE tool which was chosen to do the file characterisation.

3.3. Software tools

3.3.1 Requirements

The criteria for identification and selection of the software tools needed for this project were based upon those suitable to extract the significant properties and migrate and characterise the representation formats identified in the research part of the project. .

General criteria for the selection of software tools were:

1. *Task:* Able to identify some or all properties of an Information Object that are considered to be significant;
2. *Task:* Able to extract significant properties of source format and store them in an open, well documented destination format;
3. *Environment:* Can be compiled or operated on a number of computing operating systems;
4. Distribution: Are publicly available as a full product or in demo form for testing;
5. *Legal:* Provide clear guidance on the licence for use of the software in a production environment. Particular preference given to open source licence models;
6. *Documentation:* Are well documented.

3.3.2 Software tools available

The ability to identify, extract and convert the significant properties of a structured text file requires a combination of different software tools. Whilst there may be a variety of different suitable tools available for this, due to the computer security restraints inherent in working within a government department, the types of product freely or easily downloadable for use are limited and it was necessary, within the time available, to chose products already available to the project team. Therefore Macromedia Dreamweaver (version 8) was chosen to undertake the conversion tasks and JHOVE (version 1) was chosen for the characterisation tasks. The formats chosen as the representation formats were those that are supported by JHOVE.

- **Dreamweaver:** is a software package for the design, development and maintenance of standards-based web sites. Versions 1.0 – 8.0 were developed by Macromedia but the latest versions, CS3 and CS4, were developed by Adobe. It enables the forward and backward conversion of websites between XML and HTML 4.01 formats. However, it did not allow backward conversion to HTML 3.2 although HTML 3.2 documents could be saved in XML and HTML 4.01.

- **JHOVE:** JHOVE (JSTOR/Harvard Object Validation Environment) is an identification, validation and characterisation tool developed by JSTOR and Harvard University Library. These actions involve being able to identify files of particular specified formats, state whether particular object examples of these formats are well formed and valid, and determine the specific properties of a particular object in a supported format. It has modules to support these actions for arbitrary byte streams; ASCII and UTF-8 encoded text; GIF, JPEG2000, JPEG and TIFF images; AIFF and WAVE audio; PDF, HTML, and XML. Output from these modules is available in text and XML formats. It includes both a command line and GUI version, with the latter being used in this project.

4. Experiment

4.1. Sample data to be analysed

To demonstrate the identification, extraction and conversion of properties in a production environment the project team obtained data samples from several sources which were used as the basis for analysis. Prior to data selection, it was established that the data should represent real-world examples, i.e. structured text created in a production environment, as opposed to that created in a controlled environment for analysis purposes.

It was originally intended that all files for testing would be gathered from the UK Government Web Archive. However the availability of suitable HTML 3.2 documents was limited and it was not possible to find sufficient suitable files to build a working set of test data. Further, the project team had difficulties using the 'open url' function of the JHOVE characterisation tool which meant that websites had to be saved locally in order to be analysed by JHOVE. This in turn created problems with any websites containing GIFs, (which many of the located HTML 3.2 websites did), as these could not be rendered properly after saving. It appeared that this may have been due to a link between saved image files and the HTML files being broken and this could not be fixed within the time available. The project team also attempted to use the HTTrack open source offline web browser tool to harvest and analyse websites but was unsuccessful in getting the tool to work. This problem could also not be rectified within the time available.

Learning from this, the final test data was assembled from websites located using a mixture of random internet searching using Google Web and using suitable sites located within the UK Government Web Archive. All websites were then saved locally and it was checked that they could be rendered adequately before any experimentation took place. This process of locating suitable files proved to be time consuming. After considerable time spent searching for ostensibly suitable material in the right format, each image went through a format identification process in JHOVE in order to formally identify the format and to clarify which version of the format the structured text file used.

The final test set is made up of a mixture of websites as follows:

3 X HTML 3.2
3 X HTML 4.01
3 X XML 1.0

NB. Unless stated otherwise, further mention of these three formats refers to these format versions.

4.2. Testing Environment

All software testing was performed on a CompaqEvo D510 SFF fitted with a Pentium 4 1.80 GHz CPU, 1GB RAM and installed with Microsoft Windows XP Professional (version 2002) Service Pack 2.

4.3. Experiment testing

4.3.1 Initial Characterisation

At the same time as having the format formally identified, during the finalisation of the test data, each of the test structured text files outlined in 4.1 above, were characterised, using JHOVE. This characterisation process determines a set of properties as pre-defined by the relevant JHOVE module, and gives a value for each of these properties where present. JHOVE states that these properties are 'the format-specific significant properties of an object of a given format'.[22] However, it should be noted that the use of significance here is not defined and differs from that defined in the InSPECT project. The JHOVE concept of significant properties includes technical information such as byte order and compression scheme which would be outside of the InSPECT definition of significance because they are properties which apply to all digital objects and not just structured text.

The property values obtained from this characterisation served as the basis for comparison with our structured text files once they were migrated in the next stage of the experiments.

4.3.2 Migration

The intention was that each of the test objects would be migrated twice, from its original format to each of the other test formats. However, in experiments 2 and 3 it was found that the original HTML 4.01 and XML 1.0 files could not be backwardly migrated to HTML 3.2 using Dreamweaver. It was not possible to locate a suitable alternative tool to do this. In addition, it was noted that each HTML 4.01 and XML 1.0 file could be saved as both strict and transitional versions and so this was done where applicable.

[22] http://hul.harvard.edu/jhove/using.html

4.3.3 Post-migration characterisation

Once each structured text file was migrated, each of the new format versions was characterised using JHOVE and the output used as the basis for comparison with the original file to see how well properties were retained through migration.

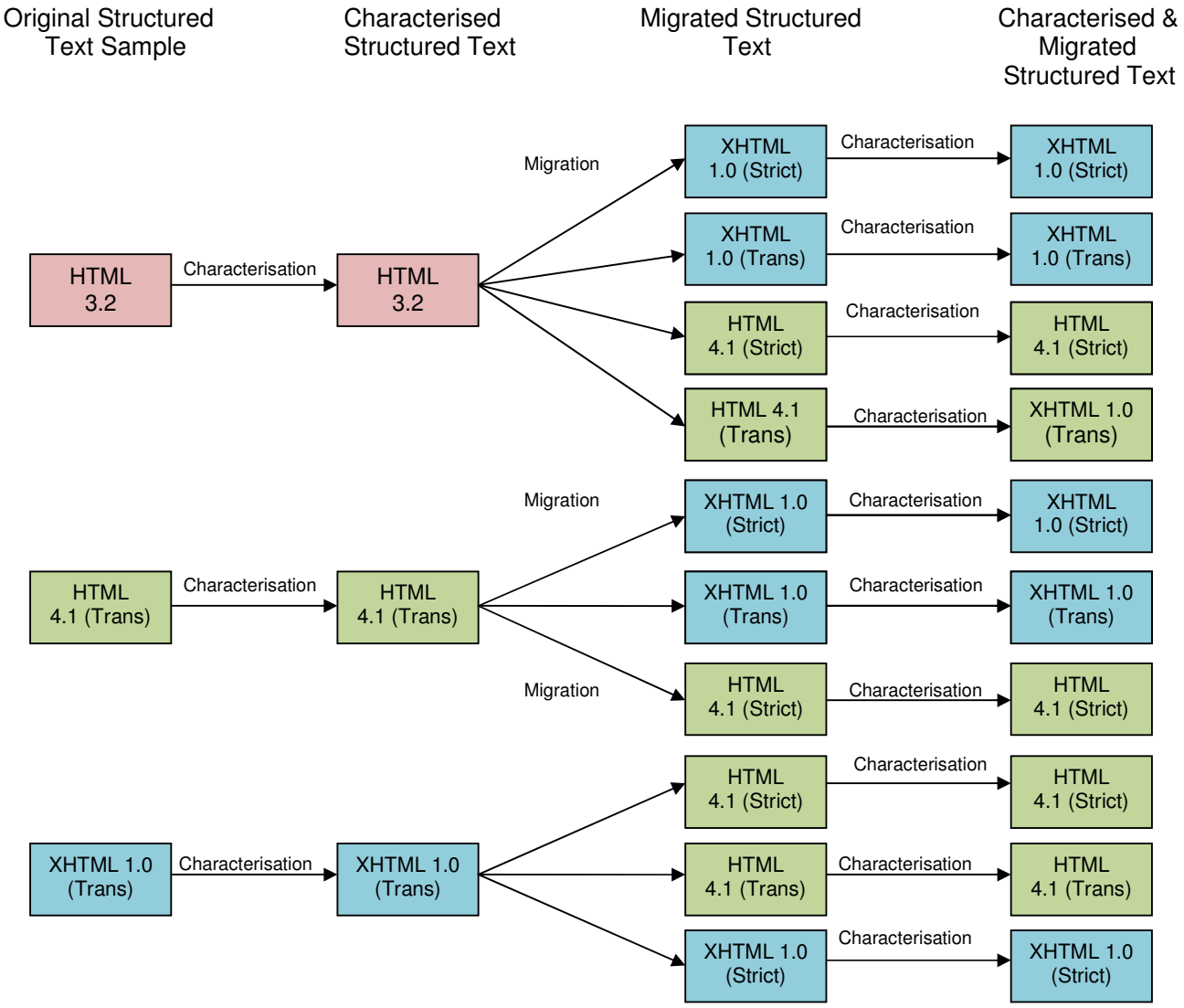


Figure 2. Illustration of experiment procedure

4.3.4 Visual assessment of converted images.

Once the automated parts of the process were carried out, a visual assessment of the structured text files was carried out. Internet browsers Mozilla Firefox (version 2.0.0.20) and Internet Explorer (version 6.0.2900.2180.xpsp_sp2_gdr.080814-1233) were used to open each file so that the evaluator could visually compare them.

4.4. Experiments

4.4.1 Experiment 1: Convert HTML 3.2 to HTML 4.1 and XHTML 1.0 using Dreamweaver

The first experiment involved converting the collected HTML 3.2 sample files to HTML 4.1 and XHTML 1.0 using Dreamweaver.

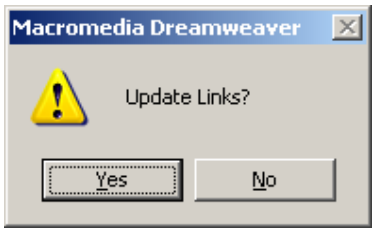
4.4.1.1. Initial Characterisation

In order to compare and measure the properties of the file before and after conversion, the initial step was to characterise the original HTML 3.2 file using JHOVE. This simply involved selecting the HTML-hul module within the JHOVE 'Edit' menu and then opening the file from the JHOVE 'File' menu. This provides a file analysis which was then saved in both text and XML format (the two available options in JHOVE) and screen shots of the JHOVE output were taken as this sometimes proved the easiest was of viewing the output.

4.4.1.2 Migration

Dreamweaver was then used to migrate the HTML 3.2 files to both the HTML 4.1 and XHTML formats in both strict and transitional forms. To do this, 'Convert' was chosen from the 'File' menu and the desired format was picked. This process allowed the formats to be saved in both strict and transitional types and so this was chosen to see, what, if any, differences this would highlight.

Each new, migrated file was then saved, with the following option to update links. Yes was always chosen.



Screengrab 1. Option presented by Dreamweaver when converting to a new format.

4.4.1.3 Second Characterisations

The migrated HTML 4.01 and XHTML files were then characterised using JHOVE, as in section 4.4.1.1 above, by choosing the HTML-hul and XML-hul modules respectively. These characterisations were used as the basis for the comparison of properties between the original HTML 3.2 and the migrated HTML 4.01 and XHTML files in order to see how the specified properties were converted.

4.4.1.4 Results – Significant Properties identified by JHOVE for original and migrated structured text files 1-4

NB - Size, status and message information was left in the results for interest but are not defined as significant properties within the InSPECT project.

	Structured Text File 1				
Metadata identified by JHOVE	**XHTML 1.0 Transitional**	**XHTML 1.0 Strict**	**HTML 4.01 Transitional**	**HTML 4.01 Strict**	**HTML 3.2**
Size	9397	9385	9302	9290	9267
Status	Not well-formed	Well-formed but not valid	Well-formed but not valid	Well-formed but not valid	Well-formed but not valid
Messages	1 Error	1 Error	22 Errors	55 Errors	55 Errors
Primary Language	-	-	-	-	-
Other Languages	-	-	-	-	-
Title	-	-	Title: <!-- This document was created with HomeSite 2.5 --> <!DOCTYPE HTML PUBLIC "-//W3C//DTD	Title: <!-- This document was created with HomeSite 2.5 --> <!DOCTYPE HTML PUBLIC "-//W3C//DTD	<!-- This document was created with HomeSite 2.5 --> <!DOCTYPE HTML PUBLIC "-//W3C//DTD

			HTML 3.2 Final//EN"> <HTML XMLns:v="urn:schemas-microsoft-com:vml" XMLns:o="urn:schemas-microsoft-com:office:office" XMLns="http://www.w3.org/TR/REC-html40"> <HEAD> <link rel="File-List" href="gps_files/filelist.XML"> <TITLE>Ohio University, plantbio	HTML 3.2 Final//EN"> <HTML XMLns:v="urn:schemas-microsoft-com:vml" XMLns:o="urn:schemas-microsoft-com:office:office" XMLns="http://www.w3.org/TR/REC-html40"> <HEAD> <link rel="File-List" href="gps_files/filelist.XML"> <TITLE>Ohio University, plantbio	HTML 3.2 Final//EN"> <HTML XMLns:v="urn:schemas-microsoft-com:vml" XMLns:o="urn:schemas-microsoft-com:office:office" XMLns="http://www.w3.org/TR/REC-html40"> <HEAD> <link rel="File-List" href="gps_files/filelist.XML"> <TITLE>Ohio University, plantbio
Meta Tags	-	-	-	-	-
Links	-	-	4	4	4
Images	-	-	6	6	6

	Structured Text File 2				
Metadata identified by JHOVE	**XHTML 1.0 Transitional**	**XHTML 1.0 Strict**	**HTML 4.01 Transitional**	**HTML 4.01 Strict**	**HTML 3.2**
Size	2426	2414	2310	2298	2281
Status	Well-formed but not valid	Well-formed but not valid	Well-formed but not valid	Well-formed but not valid	Well-formed but not valid
Messages	6 Info 7 Errors	6 Info 27 Errors	11 Errors	23 Errors	12 Errors
Primary Language	-	-	-	-	-
Other Languages	-	-	-	-	-
Title	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <HTML> <HEAD> <TITLE><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <HTML> <HEAD> <TITLE>Letter Home	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <HTML> <HEAD> <TITLE><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <HTML> <HEAD> <TITLE>Letter Home	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <HTML> <HEAD> <TITLE><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <HTML> <HEAD> <TITLE>Letter Home	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <HTML> <HEAD> <TITLE><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <HTML> <HEAD> <TITLE>Letter Home	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <HTML> <HEAD> <TITLE><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <HTML> <HEAD> <TITLE>Letter Home
Meta Tags	-	-	-	-	-
Links	2	2	2	2	2
Images	-	-	-	-	-

	Structured Text File 3				
Metadata identified by JHOVE	**XHTML 1.0 Transitional**	**XHTML 1.0 Strict**	**HTML 4.01 Transitional**	**HTML 4.01 Strict**	**HTML 3.2**
Size	6433	6421	5940	5928	5892
Status	Well-formed but not valid	Well-formed but not valid	Well-formed but not valid	Well-formed but not valid	Well-formed but not valid
Messages	6 Info 23 Errors	6 Info 138 Errors	6 Errors	62 Errors	16 Errors
Primary Language	-	-	-	-	-
Other Languages	-	-	-	-	-
Title	Oleg. K. -- HTML 3.2 by Example	Oleg. K. -- HTML 3.2 by Example	Oleg. K. -- HTML 3.2 by Example	Oleg. K. -- HTML 3.2 by Example	Oleg. K. -- HTML 3.2 by Example
Meta Tags	4	4	4	4	4
Links	-	-	-	-	-
Images	-	-	-	-	-

4.4.2. Experiment 2: Convert HTML 4.01 to HTML 3.2 and XHTML 1.0 using Dreamweaver

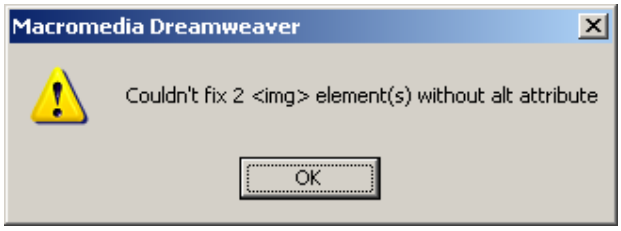
The second experiment involved converting the collected HTML 4.01 files to HTML 3.2 and XHTML using Dreamweaver.

4.4.2.1 Initial Characterisation

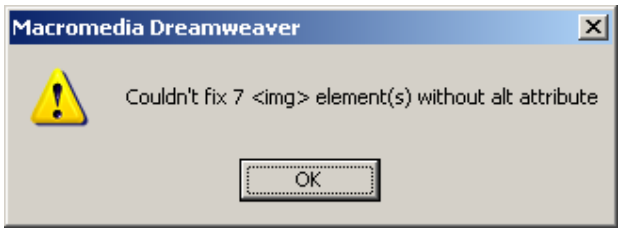
As with the previous experiment, in order to compare and measure the properties of the file before and after conversion, the initial step was to characterise the original HTML 4.01 files using JHOVE. This simply involved selecting the HTML-hul module within the JHOVE 'Edit' menu and then opening the image from the JHOVE 'File' menu. This file analysis was then saved in text and XML formats and screen shots of the JHOVE output were again taken.

4.4.2.2 Migration

The aim was that Dreamweaver would then be used to migrate the HTML 4.01 files to both the HTML 3.2 and XHTML formats using the 'convert' option, as in the previous experiment. However, the backwards conversion from HTML 4.01 to HTML 3.2 was not supported by Dreamweaver. Unfortunately, an alternative tool for this migration was not able to be found. Therefore the experiment went ahead converting the HTML 4.01 files (which were transitional) into HTML 4.01 strict and XHTML strict and transitional. When converting to XHTML transitional, files 4 and 5 produced the following messages respectively.



Screengrab 2. Message produced on conversion of file 4 from HTML 4.01 transitional to XHTML transitional.



Screengrab 3. Message produced on conversion of file 5 from HTML 4.01 transitional to XHTML transitional.

All migrated files were then saved in their new formats.

4.4.2.3 Second Characterisations

The migrated HTML 4.01 and XHTML files were then characterised using JHOVE, as in section 4.4.2.1 above, by choosing the HTML-hul and XML-hul modules respectively. These characterisations were used as the basis for the comparison of properties between the original HTML and the migrated HTML and XHTML files in order to see how the specified properties were converted.

4.4.2.4 Results - Significant Properties identified by JHOVE for original and migrated structured text files 5-8

NB - Size, status and message information was left in the results for interest but are not defined as significant properties within the InSPECT project.

	Structured Text File 4			
Metadata identified by JHOVE	**XHTML 1.0 Transitional**	**XHTML 1.0 Strict**	**HTML 4.01 Transitional**	**HTML 4.01 Strict**
Size	24068	23966	23869	23811
Status	Well-formed but not valid	Well-formed but not valid	Well-formed but not valid	Well-formed but not valid
Messages	102 Error 6 Info	224 Error 6 Info	56 Error	172 Error
Primary Language	-	-	-	-
Other Languages	-	-	-	-
Title	ARCHIVED CONTENT] Centrex - Developing Policing Excellence	ARCHIVED CONTENT] Centrex - Developing Policing Excellence	[ARCHIVED CONTENT] Centrex - Developing Policing Excellence	[ARCHIVED CONTENT] Centrex - Developing Policing Excellence
Meta Tags	3	3	3	3
Links	29	29	29	29
Images	32	32	32	32

	Structured Text File 5			
Metadata identified by JHOVE	**XHTML 1.0 Transitional**	**XHTML 1.0 Strict**	**HTML 4.01 Transitional**	**HTML 4.01 Strict**
Size	29671	29448	29292	29296
Status	Not well-formed	Not well-formed	Not well-formed	Not well-formed
Messages	1 Error	1 Error	1 Error	1 Error
Primary Language	-	-	-	-
Other Languages	-	-	-	-
Title	-	-	-	-
Meta Tags	-	-	-	-
Links	-	-	-	-
Images	-	-	-	-

	Structured Text File 6			
Metadata identified by JHOVE	**XHTML 1.0 Transitional**	**XHTML 1.0 Strict**	**HTML 4.01 Transitional**	**HTML 4.01 Strict**
Size	43890	43750	43238	43188
Status	Not well-formed	Not well-formed	Not well-formed	Not well-formed
Messages	1 Error	1 Error	1 Error	1 Error
Primary Language	-	-	-	-
Other Languages	-	-	-	-
Title	-	-	-	-
Meta Tags	-	-	-	-
Links	-	-	-	-
Images	-	-	-	-

4.4.3 Experiment 3: Convert XHTML 1.0 to HTML 3.2 and HTLM 4.01 using Dreamweaver

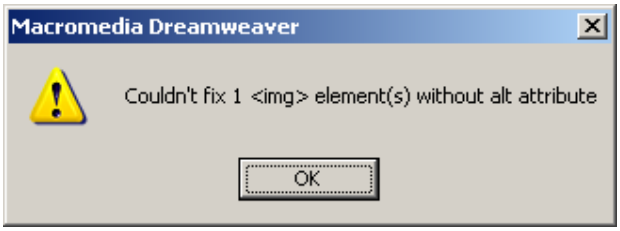
The final experiment involved converting the collected XHTML sample sites to HTML 3.2 and HTML 4.01 using Dreamweaver.

4.4.3.1. Initial Characterisation

As previously the original files to be migrated, in this case XHTML, were characterized using JHOVE in order to compare and measure the properties of the file before and after conversion. This involved selecting the XML-hul module within the JHOVE 'Edit' menu and then opening the image from the JHOVE 'File' menu. This file analysis was then saved in text and XML formats and screen shots of the JHOVE output were again taken.

4.4.3.2 Migration

Dreamweaver was again used to undertake the XHTML file migrations but as pointed out in experiment 2, files could not be backwardly converted to HTML 3.2.Therefore the experiment went ahead converting the XHTML files (which were transitional) into XHTML strict and HTML 4.01 strict and transitional. When converting from XHTML transitional to strict, file 1 produced the following message.



Screengrab 4. Message produced on conversion of file 7 from XHTML transitional to strict.

All migrated files were then saved in their new formats.

4.4.3.3. Second Characterisations

The migrated HTML 4.01 and XHTML files were then characterised using JHOVE, as in section 4.4.3.1 above, by choosing the HTML-hul and XML-hul modules respectively. These characterisations were used as the basis for the comparison of properties between the original XHTML and the migrated XHTML and HTML 4.01 files in order to see how the specified properties were converted.

4.4.3.4 Results - Significant Properties identified by JHOVE for original and migrated images 9-13

NB - Size, status and message information was left in the results for interest but are not defined as significant properties within the InSPECT project.

	Structured Text File 7			
Metadata identified by JHOVE	**XHTML 1.0 Transitional**	**XHTML 1.0 Strict**	**HTML 4.01 Transitional**	**HTML 4.01 Strict**
Size	36024	37266	37401	37389
Status	Not well-formed	Not well-formed	Well-formed but not valid	Well-formed but not valid
Messages	1 Error	1 Error	8 Errors	142 Errors:
Primary Language	-	-	-	-
Other Languages	-	-	-	-
Title	-	-	-	-
Meta Tags	-	-	19	19
Links	-	-	62	62
Images	-	-	69	69

	Structured Text File 8			
Metadata identified by JHOVE	**XHTML 1.0 Transitional**	**XHTML 1.0 Strict**	**HTML 4.01 Transitional**	**HTML 4.01 Strict**
Size	17632	17679	17557	17545
Status	Not well-formed	Well-formed but not valid	Not well-formed	Well-formed but not valid
Messages	1 Error	3 Error 6 Info	1 Error	1 Error
Primary Language	-	En	-	-
Other Languages	-	-	-	-
Title	-	Environment Agency - Home	-	-
Meta Tags	-	21		
Links	-	19		
Images	-	8		

	Structured Text File 9			
Metadata identified by JHOVE	**XHTML 1.0 Transitional**	**XHTML 1.0 Strict**	**HTML 4.01 Transitional**	**HTML 4.01 Strict**
Size	26029	25754	25584	25572
Status	Not well-formed	Well-formed but not valid	Well-formed but not valid	Well-formed but not valid
Messages	1 Error	32 Error 5 Info	13 Errors	34 Errors
Primary Language	-	En	En	En
Other Languages	-	Cy	Cy	Cy
Title	-	[ARCHIVED CONTENT] Home Office | Welcome to the Home Office	[ARCHIVED CONTENT] Home Office | Welcome to the Home Office	[ARCHIVED CONTENT] Home Office | Welcome to the Home Office
Meta Tags	-	23	23	23
Links	-	54	54	54
Images	-	9	9	9

4.5.3 Visual inspection of results.

A visual inspection of the image files in Firefox and Internet Explorer resulted in the following obvious differences being noted in the images. This was a superficial visual inspection by the project team where the evaluator was not an expert and it may be that further differences would be noted by a professional in the web design field.

Structured Text File	Visually discernible differences in conversions
Structured Text File 1	None
Structured Text File 2	None
Structured Text File 3	None
Structured Text File 4	None
Structured Text File 5	None
Structured Text File 6	The original HTML 4.01 (transitional) file and the migrated XHTML 4.01 (transitional) file display the menu at the top of the site differently to the HTML 4.01 and XHTML 1.0 (strict) files in Firefox but not IE.
Structured Text File 7	None
Structured Text File 8	None
Structured Text File 9	None

Table 1. Visually discernible differences in conversions.

5 Conclusions

The HTML Metadata which can be recorded by JHOVE is:

Primary Language
Other Languages
Metatags
Frames
Images
Citations
Defined Terms
Abbreviations
Entities
Unicode Entity Blocks

Of the 37 significant properties specified by the project team date, creator, rights and keywords were included in the Metatags section of JHOVE where relevant. In addition, the languages, frames, images, citations, defined terms (DFN element) and abbreviations metadata were also regarded as significant if recorded. However, information under abbreviations, citations, defined terms or frames was not recorded for any of the sample test images. This highlights two drawbacks with the experiments carried out:

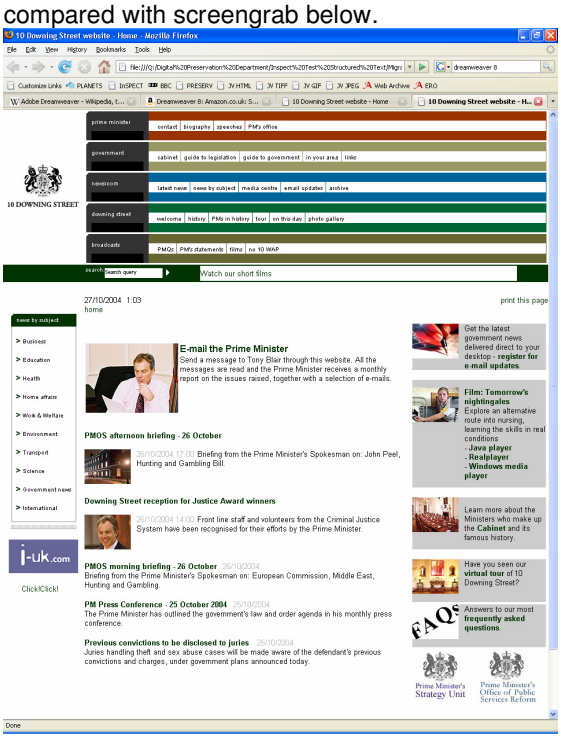
- A major drawback of the JHOVE tool is that of the 37 suggested significant properties, only 10 are potentially recorded by JHOVE. As one of the major characterisation tools available, it would be helpful for more of these to be identifiable, with values, within JHOVE.

- Four of the significant properties that can be identified by JHOVE were not represented in the test samples assembled. However this could be a fault of JHOVE in that they were possibly just not identified.

In all cases, where significant properties information was recorded, it remained the same across migrations. In all cases, the size of the file changes between migrations. File size is not a significant property but it can be seen that even where this and the format changes, it does not mean that significant property information will not be migrated correctly.

Of all the visual inspections, only one website showed any noticeable differences in how it rendered and this was only using one of the browsers, Mozilla Firefox. This difference was shown between the transitional versions of both formats when compared to the strict versions of both formats (see screengrabs 5 and 6 below). It is unclear why this happened and as no significant property information was recorded by JHOVE for this site it isn't possible to draw any conclusions about the relationship between significant properties and the migration process here.

Screengrab 5. Rendering of website in HTML 4.01 (transitional) – note rendering of colour at top compared with screengrab below.



Screengrab 6. Rendering of website in HTML 4.01 (strict) – note rendering of colour at top compared with screengrab above.

5.1 Other Issues

- Working within a government organisation produces its own additional challenges when doing this type of research work as all work has to be conducted within the standard operating procedures concerning internet and software usage. Many websites are blocked which hinders research as judging whether the site would be useful is impossible without going through procedures to get it unblocked which is time consuming and often results in it being obvious, immediately that a site is unblocked, that it isn't a useful resource. In addition, it is not possible to easily download tools to trial to see if they are suitable for a particular project. Again IT procedures need to be complied with which can make it prohibitively time-consuming when trying to analyse and compare suitable tools. In future, these additional constraints would need to be factored into such a project.

5.2 Recommendations

- Recommend that further experimentation work is done with other migration and characterisation tools to compare results and develop tools further as necessary.

- Recommended that further experimentation with other, structured text formats be done in order to see how well the other significant properties are migrated.

- Recommend that a larger sample set of test files be built up which have values for all of the significant properties (and other properties) allowable by the format for use in future tests. Web archiving is a complex and resource-intensive process[23] and it is recommended that further work with web crawlers such as HTTrack and Heritirx, and the advice of experts in the field, would be valuable in order to build up such a larger set of test files.

  Some work is currently being carried out at the University of Cologne to assemble a set of test files of various digital objects as part of the Testbed workpackage in the EU-funded Planets project[24]. It is not yet known if this resource will be more widely available in the future.

[23] See Brown, A (2006) Archiving Websites for a detailed practical analysis
[24] http://www.planets-project.eu/

Appendix 1: Software Tools

The project examined a number of software tools capable of analysing representation formats used for the storage of structured text objects. To document the process it adopted the format adopted by the CAIRO project for its tool survey[25].

Photoshop CS

Tool Name	Dreamweaver
Source URL	http://www.adobe.com/support/documentation/en/dreamweaver/documentation.html http://www.amazon.co.uk/gp/product/product-description/B000ALM5Y8/ref=dp_proddesc_0?ie=UTF8&n=300435&s=software
Formats supported	htm, html, hta, htc, xhtml, shtm, .shtml, stm, .ssi, .inc, js, xml, dtd, xsd, xsl, xslt, rss, rdt, lbi, dwt, css, asp, asa, aspx, ascx, asmx, cs, sfm, sfml, sfs, as, asc, asr, txt, php, php3, php4, tpl, lasso, jsp, jst, jsf, tld, java, .wml, edml, vbs, vtm, btml.
Technology Base	C++
Operating system	Cross-platform
Dependencies	
Licence	Proprietary
Category	Integrated web development environment
Description	Dreamweaver is a software package for the design, development and maintenance of standards-based web sites. Versions 1.0 - 8.0 were developed by Macromedia but the two most recent versions, CS3 and CS4 have been developed by Adobe.
Output methods	
Notes	

JHOVE

Tool Name	JHOVE (JSTOR/Harvard Object Validation Environment)
Source URL	http://sourceforge.net/projects/jhove/
Formats supported	Arbitrary byte streams, ASCII, UTF-8, GIF, JPEG2000, JPEG, TIFF, AIFF WAVE, PDF, HTML, and XML
Technology Base	Command line and GUI. Written to conform to Java 2 Platform, Standard Edition (J2SE) 1.4
Operating system	Any Unix, Windows, or OS X platform with the appropriate J2SE installation.
Dependencies	J2SE 1.4-compliant Java Runtime Environment (JRE)
License	GNU Library or Lesser General Public License (LGPL)
Category	Identification, validation, characterisation
Description	JHOVE (JSTOR/Harvard Object Validation Environment) is an identification, validation and characterisation tool developed by JSTOR and Harvard University Library. These actions involve being able to identify files of particular specified formats, state whether particular object examples of these formats are well formed and valid, and determine the specific properties of a particular object in a supported format. It has modules to support these actions for arbitrary byte streams; ASCII and UTF-8 encoded text; GIF, JPEG2000, JPEG and TIFF images; AIFF and WAVE audio; PDF, HTML, and XML. Output from these modules is available in text and XML formats. It includes both a command line and GUI version, with the latter being used in this project.
Output methods	Text, XML
Notes	

[25] Further details of the format can be found on p11 of the Cairo Tools Survey, located at http://cairo.paradigm.ac.uk/projectdocs/index.html