Journal of Universal Computer Science, vol. 7, no. 12 (2001), 1125-1136
submitted: 15/10/01, accepted: 7/12/01, appeared: 28/12/01

Transclusions in the 21st Century

Harald Krottmaier
(Graz University of Technology, Austria
hkrott@iicm.edu)

Hermann Maurer
(Graz University of Technology, Austria
hmaurer@iicm.edu)

Abstract: When quoting some part of a document authors usually cut and paste the relevant content into the new document. Thereby the connection between this selected part and the original document is lost. Transclusions – first mentioned in 1960 by Ted Nelson – address this problem of 'lost context'. With transclusions it is possible to store information about the original document and the exact position of the quote in the newly created document and provide the reader with additional navigational features. Document formats and information systems matured over the last 40 years. This paper gives an overview of some document formats available today in the WWW environment and points to some requirements for server systems providing transclusions. Thereafter we present some ideas on how to implement transclusions based on a Hyperwave Information Server (HIS).
Key Words: Transclusion, Hyperwave Information Server
Category: H.1, H.3

1 Introduction

More than 40 years ago, Ted Nelson dreamed about a universal information system called Xanadu. One key feature of that system was the idea of transclusions ([Nelson, 1995]):

> The central idea has always been what I now call transclusion, or reuse with original contexts available, through embedded shared instances (rather than duplicate bytes).

The following aspects must be considerd when talking about transclusions: the original piece of text to be used as quote, the newly created document using a quote, the system where the documents are stored and the environment of the reader. In the late 1960's, when defacto no information systems where available, Nelson based his ideas on one large and homogeneous system. As we all know there are now many different systems available. This makes it much more difficult to provide a solution for transclusions.

Nowadays HTML (HyperText Markup Language) is the most common format in distributing text documents on the web. Hence we discuss it in section 2. Usually not a whole document is quoted, but parts of it. The selected part must therefore be defined in some way.

When using some data to be 'included' in a document, this data must be accessible to the author via some protocol. That means in the context of today's available IT infrastructure that the reused data must be stored on some server system on the Internet. The systems at issue are mostly accessible via HTTP (HyperText Transfer Protocol, [Berners-Lee et al., 1996], [Fielding et al., 1999]) and have a restricted command set. See section 3 for details about this aspect.

To enable reuse with the original context available, both of the involved serversystems or the client program must provide navigational features to the reader. It must be visible to the reader which part of the document is written by the author and which part is quoted. The original context of the quote is also important to the reader, especially in scientific publications.

Using a special information server (Hyperwave Information Server, HIS) with sophisticated features as publishing server system, makes it easier to solve some of the problems. We introduce a concept on how to implement simple data inclusions with documents stored on a HIS in section 4.

Before we start with the more technical parts, let us summarize the need of inclusions of text data and point out some problems.

1.1 Intellectual Property

As noted above, using some text via cut and paste removes not just the context of the quote, but also other information (meta data) of it like the original author, date of publication etc. With systems supporting the transclusion idea this problem would simply disappear. Every access to the quote would be redirected to the original server. Therefore even charging for pieces of data could be implemented. Nelson dreamed about automatically paid royalties to the owner of each document. However, we are miles away from a solution like this. Micropayment (the payment of very small amounts of money called 'microcents') is still under development. The W3C E-Commerce/Micropayment Working Group ([W3C, 2001b]) specified a common markup for Micropayment but currently this specification is just available to members of the working group. Most of the content providers finance their activities via advertisements (like banners etc.). These advertisments can be easily removed by 'web washer' programs (i.e. programs which block specific URLs or content, matching some pattern). Implementations of transclusions may extract just the interesting part of some document. Hence the quoted content can be displayed without the advertisement. This is not what an author wants, therefore we propose that the original author must permit the reuse of the whole document or parts of it.

1.2 Disk space

If the same material is used in numerous documents the necessary space to store the material is reduced when using transclusions. In 1960 storage-space was an issue because of the high costs. We are now in the year 2001 and it is well known that disk space of several gigabytes is not a problem neither in terms of costs nor in terms of availability. Hence one might believe that this issue is not a problem any more. This is not true. Think of a very large multi media document where you want to delete or add a single character. If someone wants to preserve previous versions of the document separate files of the document have to be stored. After several changes in the document, disk space is again an issue. Therefore transclusions or version control systems (like cvs, [CVS-HOME, 2001]) should be used. As this paper is not about version control systems, we just include a short discourse on how version control with transclusions may work in section 5.

The availability of large local disk space brings a new aspect of transclusions. Locally stored data should also be available for transclusions! If someone e.g. bought and installed the new Brockhaus multimedia DVD ([BMM3, 2000], one of the largest available multimedia collections), why is it not possible for an author to refer to a special part of this – locally installed – multimedia collection? There must be a possibility to address parts of documents stored on the local file-system in an arbitrary format. As another example imagine a database query. Why can't a system integrate the *query* rather than the *result* of the query?

1.3 Update

The publishing system designed by Nelson (Xanadu) was based on a 'write once / read many'-storage system. Therefore a once published document could not be changed and stayed stable forever. Nowadays information systems are more dynamic. This results in several problems especially for authors using quotes. For published documents which are changeable by the author ('privately published'), the advantage of updating documents automatically arise ([Nelson, 1987]):

> No copying operations are required among the documents throughout the system and thus we solve the problems of updating documents. We solve this problem simply by windowing to a changing document. Thus the problem of distributed update, . . . , disappears.

Replication mechanisms for certain systems are available, but these mechanisms mostly work in one homogeneous environment. The web consists of very inhomogeneous systems. It is difficult to keep the copies up to date. Some systems on the web, e.g. news services, implement distributed updates of material using a common protocol. Nevertheless, it takes time, bandwidth and also disk space to synchronize content.

In some circumstances automatic update of parts of a document is not what an author wants! Therefore some attributes concerning the update behaviour of the quote window must be introduced to prevent automatic updating. For a detailed discussion about different kinds of quote windows and how a system should handle quotes see section 3.

1.4 Two way reading

Having the possibility to read a quote in the original context is 'added value' to the reader and of interest to the author of the original part of information. I.e. the question: 'Who uses (parts of) my article?' can easily be answered if the part is included via transclusions. The answer to that question is invaluable! Bidirectional links are the obvious requirement to enable this feature. Although WWW links have now been arround for over ten years, only few systems implement the paradigm of two way links.

As the list above shows there are enormous advantages of using transclusions rather than simply cut and paste parts of a document. It is obvious that some additional tools are required when preparing documents. In this paper we introduce some ideas on how to implement the idea of transclusions using a Hyperwave Information Server. A transclusion is specified by the author with some tool and is then converted to some special object in the database. Thereafter the system takes control of the compound document and provides author and reader with additional navigational features to the original document of the quote. Let us now take a look at different document formats on the web and compare some of the features relevant to the transclusion concept.

2 Document Formats

Since we propose to use transclusions in an internet environment, we take a closer look at the document formats available there. Documents are usually described using HyperText Markup Language (HTML, e.g. [W3C, 2001a]) or the Extensible Markup Language (XML, [W3C, 2000]).

2.1 HyperText Markup Language

Documents are usually sequences of characters. In HTML special sequences of characters ('tags') indicate some markup. This is an indication for the rendering program to display parts of the material in a special manner to the user. All content displayed to the user – except images, inline frames and objects – is stored inside the HTML file. Hyperlinks are also stored inside the file and are unidirectional. This makes it very difficult to manage the integrity of links and include sophisticated navigational features. Without the ability to include any type of content stored outside the HTML file in the rendered file and bidirectional links, HTML is not suitable for transclusions. Let us take a look at a very simple tag in HTML: the tag to include images in the document (IMG-tag).

Addressing documents in the world wide web is done via uniform resource identifiers (URI, [Berners-Lee et al., 1998]) and uniform resource locators (URL, [Berners-Lee et al., 1994]). To include an image which is accessible via some URI in HTML formated documents, simply add the following tag:

```
<IMG SRC=[URI] {WIDTH=[X] HEIGHT=[Y]}>
```

The browser program now 'knows' to include some image addressed by the specified URI in the document. The user reading that document on the screen does not know the exact storage location of the image (without reading the source code of the document)! The physical location of the image does not have to be the same as the physical location of the document. The rendering routines of browsers do not distinguish the rendering of local and remote images.

Inline images are – as links (A-tag) in HTML documents – unidirectional, i.e. the author of the image does not know in which documents the image is used. Therefore changing the content of the image without changing the URI can be misleading. See below for a discussion about links in HTML and see section 3 for requirements of a system supporting transclusions (bidirectional links etc.). But there is another restriction with the image tag: It is not possible to only include a part (e.g. 'the upper left corner') of an image. Therefore an image must be used as it is – even in a different size – or a copy of that image must be prepared.

To summarize: Existing images can be reused exactly as they are in different contexts without physically copying them. Including images in HTML is therefore a very restricted version of transclusion.

[Pam, 1996] proposed simple tags for text inclusions similar to the IMG-tag. However, these tags do not provide any backlinks from the quote to the document using that quote.

The HTML version 4-Recommendation ([Raggett et al., 1999]) in late 1999 defines some possibilities to include other kinds of documents via the OBJECT and IFRAME-tag. The OBJECT-tag may include single objects of any type and IFRAME (inline frame) can be used to include a whole HTML-document. However, the same problems as with inline images remain: addressing parts of a document is not possible and backlinks to the original context are not supported by these tags. Please note, that links are stored *in* the HTML document, i.e. try to link from one HTML document (document A) to some other document (document B) implies a modification of document A! Creating a link from document B back to document A results in a modification of document B. Therefore a document management system must aid the user when performing such modifications. It is possible to implement bidirectional links based on HTML files if the links are parsed by the publishing system and are stored in some link-database. As noted above bidirectional links are an absolute requirement for transclusions. Please see section 3 and [Maurer, 1996] for a detailed discussion of bidirectional links.

2.2 Extensible Markup Language

The Extensible Markup Language (XML) is a very general format for structured documents and data on the Web. XML is derived from Standard Generalized Markup Language (SGML) but is much easier to handle for the user. Several working groups are building recommendations and standards. The most interesting parts in the XML community for our discussion of transclusions are the parts from the XML Path Language (XPath, [Clark and DeRose, 1999]) and XML Linking Language (XLink, [DeRose et al., 2001]).

With XPath it is possible to address 'nodes' of documents, e.g. chapters, paragraphs and words. In the transclusion context additional features like 'identify the selection made by the user with the mouse' must be available, therefore an extension to XPath, the XML Pointing Language (XPointer, [DeRosea et al., 2001]) was specified by the W3C. XPointer is at the time of writing a 'Candidate Recommendation' and some implementations already exist. Hence XPointer can be used to identify parts of an XML document and can therefore be used for defining the quote to be inserted in another document. XML media types (text/xml, application/xml, text/xml-external-parsed-entity and application/xml-external-parsed-entity) are supported at the outset, therefore using HTML-documents as source of the quote requires additional format conversion or implementation of extension facilities. Internal structures of XML documents are addressed by element types, attribut values, character content, and relative positions.

Linking in XML (XLink) is much more powerful than in HTML. As mentioned above, in HTML links are stored inside the document. In XML links may be stored outside of the document in some link database and therefore also read only documents (like documents stored on a CD-ROM or remote WWW server) may be interlinked by the user. Not only unidirectional links, but also more complex link structures are provided by XLink. It is possible to interlink more than two resources and to associate meta data with the link element. The destination of a link is defined via XPointer described above.

XML Inclusions (XInclude, [Marsh and Orchard, 2001]) is another specification where a processing model and syntax for general purpose inclusion is discussed. It differs from XLink links with attribute `show="embed"` and provides a media type specific (XML into XML) transformation.

This was a very short overview of some related document formats and specifications in our discussion about transclusions. In the last months some W3C workingdrafts of XML specifications became recommendations. We expect some full implementations of XPointer and XLink very soon. The best technology will not survive if the browsers available to users are not supporting it. At the time of writing webbrowsers supports XML rudimentarily and information providers have to deal primarily with HTML. Therefore either XML to HTML processing must be performed at the server side or sophisticated server systems must combine HTML-documents and XML-linking in some way. Let us now take a look at some aspects of a server system supporting the transclusion idea.

3 Server Systems

Usually at least two server systems are involved when talking about transclusions: The system where the original quote is physically stored, and the system where the *compound document* (document using that quote) is stored. Please note that also a 'cascading composition' of transclusions is allowed, i.e. parts of a document might be included which itself uses quotes from some other document. We will call the representation of the quote in the compound document a *quote window*.

In this section we take a closer look at some requirements for the different server systems involved when processing documents. General requirements like availability, fault tolerance etc. are not discussed here. If the server which holds the quote is not available because of a server break down or any other reason, some message will appear to inform the reader of the *broken information*.

When looking at the compound document, additional navigational tools must be displayed to the readers allowing them to jump to the original context of the quote ('two way reading'). The whole quote must be clearly rendered as quote to the reader.

There are different kinds of quotes: static and dynamic ones. An author of a compound document must clearly define which kind of quote is to be used in the document. Static quotes should change neither the content nor the layout over time. However, if changes occur, the author of the compound document must be informed and it must be clearly indicated to the reader that the displayed content of the quote is not the quote the author originally used. The quote may also completely disappear, depending on the attribute of the quote set by the author. If bidirectional links are supported by the involved systems, the system providing the quote can inform the author of the compound document about a modification. If they are not supported, the server holding the compound document must do this task. In HTTP a simple 'GET' request can check for any changes of the content of a document ('if-modified-since' parameter). Therefore the server system where the compound document is stored must check all quotes on a regular basis and must inform the author accordingly. The author then must react via some webform or via email.

If the quote is dynamic – e.g. 'the joke of the day' or some weather forecasts – the structure of the content should not change and the quote must be identified in a uniform manner e.g. by identifier tags. However, if the system can't find the quote, the author must be informed.

It might happen that the context of the quote changes, but not the quote itself. The discussed procedure to detect changes in documents will therefore notify the author of the compound document needlessly. Implementing a simple checksum of the quote and storing that checksum as attribute of the quote window will prevent such notifications.

Changes in the document where the quote is stored are very dangerous for the author of compound documents, therefore the server systems involved should provide version control facilities to authors of WWW documents. If such facilities are available, users might see a complete timeline of a quote.

While checking the status of the quote it may happen that the document where the quote is stored can't be accessed. The system must then determine if the document was removed permanently or the service is temporarily unavailable. If the document was removed, the author again must be informed and the quote window will be automatically deactivated by the system. Handling *broken information* is a key feature of a transclusion system. Therefore an easy to use and understandable interface has to be implemented.

As noted above, bidirectional links are of great help for both reader and author of documents. They add additional information to the documents and can answer quite difficult questions like 'Who uses parts of this document?' or 'What is the original context of that quote'. However, a very popular document quoted by many other authors may provide the reader with too much unwanted information. Therefore it must be possible to filter incoming links depending on various attributes, e.g. 'show just incoming links from specific server systems'.

The separation of document content and linking information is an absolute requirement for both systems. Unfortunately not many systems available provide this paradigm. XML and XLink, respectively, may solve this problem in the future.

4 Implementing Transclusions

In this section we discuss some ideas of how to implement transclusions using an existing, very sophisticated information system named Hyperwave Information Server (HIS, e.g. [Maurer, 1996]). Many necessary features for transclusions are already built into the system and therefore the effort of implementation will be minimal.

Since users' browsers are not predictable, we assume standard WWW browsers, capable of rendering HTML documents. Therefore all of the including and parsing processes must be performed on the server side. In an initial scenario we consider a HIS as server system for compound documents and some HTTP server as server serving the quote to be included.

As the discussion above showed, separation of content and linking information is an absolute requirement when building a system which supports transclusions. This implies the use of a link database which is already built into HIS. If the document format itself does not support this separation – like HTML – the systems have to handle this task. To give an example: When a document in HTML-format is uploaded to the system all the links stored *in* the document are parsed and *link objects* are created by the system. These link objects – holding some attributes, like exact position of the link and linktarget etc. – are then controlled by the link database. This means that after operations on document objects – like (re-)moving, renaming etc. – the links are still valid and the common HTTP-error '404 – Requested URL not found' is avoided. This link database enables also arbitrary attributes to be stored with the link object. It is obvious that links to objects which are not controlled by the system (i.e. links to some different server system) may become inconsistent. To have full control over link visualization we suggest to include *remote objects* for *every* 'external' link. *Remote objects* are representations of non-local (i.e. remote) stored data in HIS. Several service programs running regularly may then check for existence or modification of the target of the link.

The same mechanism can be used for implementing transclusions on a HIS. The remote object is a representation of the 'quote window' in the system. Special attributes enables a distiction between 'ordinary' remote objects and special transclusion objects. The displaying process have to evaluate the attributes stored with the object and perform the appropriate actions.

Much work has been done to define exactly some part of the HTML-page (see e.g. [Poon and Kontogiannis, 2001] or [Huck et al., 1998]). Graphical tools like the WysiWyg Web Wrapper Factory (W4F, [Sahuguet and Azavant, 1999]) make it quite easy for the user to specify the textregion to quote. Depending on the tool we build into the system, the quote window must store the definiton accordingly.

The remote object is responsible for retrieving the content selected. For HTML documents, the exact definiton of the quote must be in a syntax similar to XPointer. Parts in plain text documents must be addressed via a simple string range (like the 'string-range' definition of XPointer). Please note that not only text based document formats, but also multimedia documents like images, sound, video etc. can be included with this approach. The resulting HTML page is assembled on the server side, therefore simple HTML-links to the original context of the quote can be inserted automatically to enable 'two way reading'.

The obvious drawback with this generic solution is the missing backlink from the quote to the document. Some systems are trying to solve this problem by executing special queries on search engines trying to get all of the documents pointing to the document containing the quote. However, this approach is inappropriate because only documents indexed by that search engine are returned. Furthermore it is not possible to jump immediately to the exact position where the quote was used. If the quote is also stored on some Hyperwave Information Server it is possible to make use of the integrated link database and display a backlink to the compound document. A timeline via different versions of the quote is also possible when the quote itself is stored on some Hyperwave Information Server. Therefore 'real' two way reading can be implemented for the benefit of the reader.

For the first implementation we assume as source format of the compound- and inlined document HTML. Even with this simple file format problems may arise when quoting parts of a HTML document. The text might either be quoted without the HTML-tags or the system rejects a transclusion if corrupt syntax is found in the quote. However, certain tags (like `</HTML>` etc.) must not appear in the quote.

In a universal transclusion system it must be possible to quote non-HTML documents. A user may want to transclude parts of a PDF-file or any other text format. As long as the transcluded document format can be translated in some ways into a common format, transclusions are possible.

5 Applications for Transclusions

Especially in scientific publications it is necessary to access the sources of a quote. Therefore our aim is to provide this transclusion feature in the Journal of Universal Computer Science ([J.UCS, 2001]) to make it possible for the authors to use transclusions in the HTML format of the article.

Another application of transclusions is version control. Imagine a document where a word is replaced. The new version of the document will then only contain everything before the word as transclusion, the new replaced word, and the rest of the document as transclusion. Also shifting the position of two paragraphs can be implemented without wasting disk space.

An author of documents with inclusions may also use the standard rights and user management tools available with a Hyperwave Information Server. Using these features enables different representations of one and the same document for different users or user groups. Combined with user preferences, this is a very flexible way for providing content.

6 Conclusion and Future Work

This paper shows that the concept of transclusion is still as important today as it was when proposed 40 years ago. Amazingly few implementations exist. The described structure on a Hyperwave Information Server enables inclusions of some text-sources available on the web. In a first implementation we consider intraserver documents to reduce the known problems of server breakdowns and communications failures. Transclusions of locally available sources of text, images and other data is not available but there is a need for it. Therefore, future work will address development of tools to enable inclusions independent of location and format of data.

References

[Berners-Lee et al., 1996] Berners-Lee, T., Fielding, R., and Frystyk, H. (1996). Hypertext Transfer Protocol – HTTP/1.0.

[Berners-Lee et al., 1998] Berners-Lee, T., Fielding, R., Irvine, U., and Masinter, L. (1998). Uniform Resource Identifiers (URI). RFC 2396.

[Berners-Lee et al., 1994] Berners-Lee, T., Masinter, L., and McCahill, M. (1994). Uniform Resource Locators. available online `http://www.w3.org/Addressing/rfc1738.txt` .

[BMM3, 2000] BMM3 (2000). Brockhaus Multimedia. available online `http://www.brockhaus-multimedia.de` .

[Clark and DeRose, 1999] Clark, J. and DeRose, S. (1999). XML Path Language (XPath) Version 1.0. available online `http://www.w3.org/TR/1999/REC-xpath-19991116` .

[CVS-HOME, 2001] CVS-HOME (2001). Concurrent versions system homepage. `http://www.cvshome.org` .

[DeRose et al., 2001] DeRose, S., Maler, E., and Orchard, D. (2001). XML Linking Language (xlink) Version 1.0. available online `http://www.w3.org/TR/xlink` .

[DeRosea et al., 2001] DeRosea, S., Maler, E., and Jr., R. D. (2001). XML Pointer Language (XPointer) Version 1.0. available online `http://www.w3.org/TR/2001/CR-xptr-20010911` .

[Fielding et al., 1999] Fielding, R., Gettys, J., and Mogul, J. (1999). Hypertext Transfer Protocol – HTTP/1.1.

[Huck et al., 1998] Huck, G., Fankhauser, P., Aberer, K., and Neuhold, E. (1998). JEDI: Extracting and Synthesizing Information from the Web. In *Cooperative Information Systems (COOPIS)*.

[J.UCS, 2001] J.UCS (2001). Journal of Universal Computer Science. available online `http://www.jucs.org` .

[Marsh and Orchard, 2001] Marsh, J. and Orchard, D. (2001). XML Inclusions (XInclude) Version 1.0. available online `http://www.w3.org/TR/2001/WD-xinclude-20010516` .

[Maurer, 1996] Maurer, H. (1996). now Hyperwave — The Next Generation Web Solution.

[Nelson, 1987] Nelson, T. (1987). *Literary Maschines*. Ted Nelson.

[Nelson, 1995] Nelson, T. (1995). The Heart of Connection: Hypermedia Unified Transclusion. *Communications of the ACM*, 38:31–33.

[Pam, 1996] Pam, A. (1996). Methods for implementing transclusion of text into HTML pages. Technical report, Xanadu Australia.

[Poon and Kontogiannis, 2001] Poon, F. and Kontogiannis, K. (2001). i-Cube: A Tool-Set for the Dynamic Extraction and Integration of Web Data. *LNCS 2040*, pages 98–115.

[Raggett et al., 1999] Raggett, D., Hors, A. L., and Jacobs, I. (1999). HTML 4.01 Specification, W3C Recommendation. available online `http://www.w3.org/TR/html4` .

[Sahuguet and Azavant, 1999] Sahuguet, A. and Azavant, F. (1999). Wysiwyg web wrapper factory. In *WWW Conference 1999*.

[W3C, 2000] W3C (2000). Extensible Markup Language (XML) 1.0 (Second Edition). available online `http://www.w3.org/TR/2000/REC-xml-20001006` .

[W3C, 2001a] W3C (2001a). HyperText Markup Language. available online `http://www.w3.org/MarkUp` .

[W3C, 2001b] W3C (2001b). Micropayments Markup Working Group. available online `http://www.w3.org/ECommerce/Micropayments` .