Portable Hypermedia: a New Format for WWW Documents

László Kovács, András Micsik

MTA SZTAKI
Computer and Automation Research Institute
of the Hungarian Academy of Sciences
Department of Distributed Systems

MTA SZTAKI, H-1111 Budapest XI. Lágymányosi u. 11. Hungary
{laszlo.kovacs, micsik}@sztaki.hu

Abstract

Several problems are revealed into lights in the case of handling of compound hypermedia WWW documents (document-like objects) of the current Internet. The requirements for a new container architecture are discussed and as a solution, the Portable HyperMedia (PHM) format is developed. A set of metadata is attached to the compound hypermedia document which is compatible with the architecture of the Warwick Framework. Structure and operations on PHM are discussed and basic tool set is introduced for handling as well.

Keywords:

hypermedia documents, document-like objects (DLO), WWW, portability, Warwick Framework, Dublin Core, Internet

1.0 Introduction

As the World Wide Web [WWW] spread the world from the beginning of this decade, it incorporated more and more powerful tools and formats, and the information served via WWW became more and more complex. The content and layout of WWW pages become competitive with printed material, and in other aspects WWW pages have far more potential than printed documents. The meaning of document in case of the WWW is changing. WWW documents are sometimes more similar to a piece of software than to printed material. They may contain animations or has an annotation facility, and what is most important they are linked together.

The Dublin Metadata Workshop [WGMD95] investigated this new kind of information source, and tried to set the floor for descriptive techniques for WWW documents which were termed Document-like Objects (DLOs) [Ferber96]. Books and other printed material have traditional descriptions and cataloging technique. Something similar is desired for DLOs for classification, search and retrieval purposes.

DLOs can also be called as hypermedia documents. A type of hypermedia document can be created and played with popular applications like ToolBook or HyperCard, and consists of a single file. These are easily manageable, viewable and transferable. A large part of the ToolBook functionality is achievable by using a Java enabled WWW browser, yet it is a much harder task to migrate a complex WWW document to another server. One cause for this is that hypermedia documents on the Internet may consist of hundreds of files. Anyway they are still desired to use as single documents, to send, to store, to manipulate easily. But this is not the case, since a DLO has a complex nature:

- it may contain files in lots of different formats: text, graphics, animation, video, audio and 3D (VRML)
- it may even contain executable parts (JAVA and CGI)
- it may contain objects which respond to object requests
- its files and data are interconnected with links.

In this paper a way for evolution of Internet hypermedia documents is shown, by first examining common problems in current usage of these documents (Section 2), then setting the requirements for current needs (Section 3), and proposing a new container format, the Portable Hypermedia (PHM) for the Internet society. The solutions offered for the problems listed in Section 2 are readable in Section 4.

2.0 Known problems with current WWW usage

2.1 Naming and addressing

Currently WWW documents are addressed with URLs [RFC1738], which are machine- and hostname-dependent. An HTTP URL consists of the scheme descriptor (http), a descriptor for an Internet host (the server), and a descriptor for the document inside the host (path). Furthermore the descriptor of the document is based on its location in the filesystem. An URL is therefore prone to changes because of multiple reasons, since documents can be moved inside the filesystem, and the DNS name and port number of the server may also change.

The usual solution is that WWW server names and filepaths for main parts of a WWW server are chosen very carefully and kept unchanged as long as it is inevitable. There are efforts [PURL, CNRI-h] to decouple physical location from the address of documents. These introduce names for documents which can be mapped to several URLs from where the document can be retrieved. This is done under the global specification of Uniform Resource Identifiers [UR*].

However the use of document names or URNs in the Internet offers only the tool for solving problem, and it does not tell us how to name millions of WWW pages existing today on the Internet.

2.2 Web server maintenance

Maintaining link integrity

Pages of a WWW server contain lots of links pointing at other local and remote WWW pages. If some page has to be withdrawn or moved to another place, it has serious effect on the integrity of links both locally and outside the server. First local links has to be updated reflecting the changes in the location of the moved document. Secondly there has to be a mechanism to notify those following unchanged links from remote servers that the location of the page has changed. Optionally an effort may be taken to notify the administrators of all remote servers from where there is link a to the moved document.

Links can spoil mainly for two reasons: links to remote pages expire because the server address has changed or the page has been moved inside he server, while links to local pages can loose correctness if local pages are moved.

Inserting new documents into the server

Whenever a more complex document (containing several HTML pages) is to be inserted into a server, the administrator has to make references to the new document, linking important parts of the document with the rest of the server. This sometimes requires the exploration of the document, to identify the important parts. Afterwards the correct operation of the inserted document has to be verified. This includes the check of links pointing outside from the document, the check of images, CGI scripts, and that all MIME types are handled correctly. This process is time-consuming and cumbersome, and may have very occasional problems, for example when the document index is called “Welcome.htm” instead of “index.html”.

Mirroring

The document insertion process has to be automated in case of mirroring, when changes in a master document are frequently implemented in the replicas as well.

2.3 Handling of compound hypermedia documents

A WWW document may consist of practically unlimited number of files. These files represent different multimedia types (text, audio, images, etc.) and therefore are in different formats. Files are stored in usual filesystems, i.e. in a directory hierarchy in most cases. This is called the storage structure of the document and can be drawn as a tree.

Hyperlinks are pointing from locations in one presentation piece to other presentation pieces (internal links) or to places outside of the document (external links). This is called the link structure of the document, and can be drawn as a directed graph. The fact that links are described with the help of the storage structure explains that changes in the storage structure may require simultaneous changes in the link structure.

Applying executables in the document may introduce further dependencies from the link and storage structure, for example when a CGI script has to read a separate parameter file and output a page containing internal links. These dependencies are usually hidden in the code of the executable.

Basic operations on a WWW document is viewing/executing a node and following a link to another node. Different node formats require different pieces of software to view or execute. Proper presentation/operation of a complex WWW document, for example a searchable image collection is a hard configuration and installation task on both the server and the client side.

There can also be operational differences viewing a document locally or viewing from a server (e.g. CGI scripts cannot be run locally). The detection of what parts are working and what parts are not can only be done by manually testing every link and every file. The most common manifestation of these problems when users would like to save a WWW document locally and use it from their disk. A connected problem is the lack of tools/protocols for replicating a multi-file document locally.

2.4 Metadata and advanced document retrieval

WWW document metadata may have several intents of use. The two most important purposes are the following:

- General description of document for cataloging and indexing (such as Dublin Core metadata)
- Information that is needed for correct presentation and management of the document

These two classes of metadata will be called here cataloging and technical metadata. Technical metadata is to expose file and software dependencies which are necessary for integrating the WWW document into a WWW server properly, but are otherwise very hard to detect.

With the help of metadata more elaborate document retrieval tools would be possible to construct, and more manageable documents would be possible to create. Metadata are applicable presently for WWW pages, yet there are only rear occasions of their use. The correct definition of syntax and semantics of metadata currently evolving can speed up the spread of metadata usage. However another speeding factor can be the definition of what documents should have metadata and where should it be placed. It is cumbersome to equip hundreds of

HTML files with metadata which are furthermore very similar. Currently there is no method or practice to store/ find metadata for WWW documents as a whole.

3.0 Requirements for hypermedia documents on Internet

Given the above problems we can distill a set of requirements for the everyday use of hypermedia documents or DLOs on the Internet. The following vaguely defined requirements are supposed for DLOs:

- URIs are used as existing syntax and semantics for links,
- most presentation formats widely used on the Internet are integrated,
- movable locally, transferable between servers and also viewable locally from the disk if possible,
- have metadata for cataloging and management purposes,
- have widely available tools for management
- extensible

There exist partial solutions for these requirements. One example is DigitalPaper [DP]. It is a single file document format with enhanced portability options. Its drawback is that it applies a totally new document format, which needs appropriate software for viewing and creation. Another direction taken by authors of WWW documents is to create WWW documents in a certain way so that some of the problems mentioned above can be eliminated. This approach uses existing formats and existing software therefore is usable more widely.

Clearly there is a need in the Internet community to handle units bigger than one WWW page. In our suggestion a container format is introduced for WWW pages, with the twofold aim of keeping present day formats and tools usable, while enhancing the representation of semantical relations within WWW documents. This container format is called Portable HyperMedia (PHM) as it enables the handling and porting of hypermedia documents or DLOs.

4.0 Portable HyperMedia (PHM) format

The container format is built upon the storage structure of WWW documents. A PHM document is stored in a directory hierarchy with a single root directory. Files with hyperlinks inside the PHM has to meet some rules:

- links pointing at files inside the PHM document are relative,
- links pointing outside of the PHM document are absolute. (In case of embedded PHM containers, only the outermost container obeys these rules.)

The metadata contained in the files of the PHM document has to be prepared with consideration that the PHM document itself has a central metadata. In most cases a pointer to the central metadata, or to the root of the PHM document would suffice. This metadata, concerning the whole PHM document is stored in a fixed metadata directory under the root directory. The metadata is arranged in packages according to the Warwick Framework [LLD96].

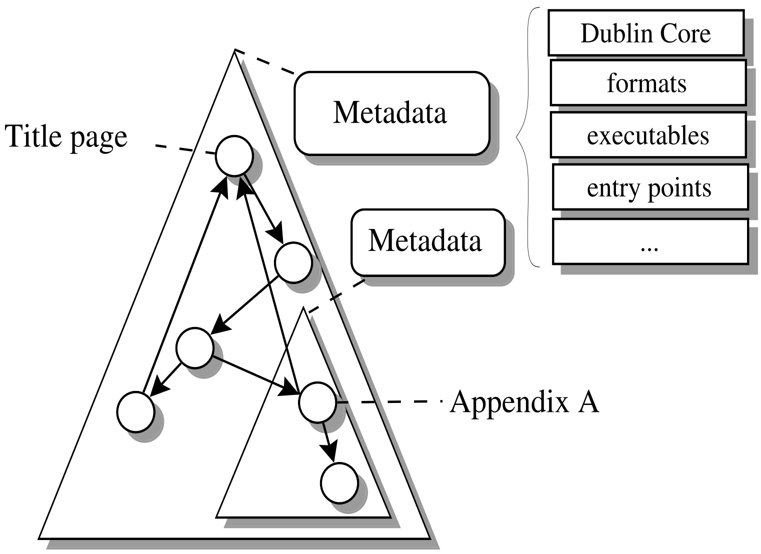

FIGURE 1. The PHM container format

The Warwick Framework is an architecture for aggregating multiple sets of metadata. It was the result of the 2nd Metadata Workshop in Warwick U.K. in April 1996. The framework allows to define various metadata sets for various purposes with different syntax. These metadata sets are called packages. Packages may be grouped in containers in an arbitrary depth and packages may be referenced with a link as well. This architecture ensures that a wide variety of metadata packages can be developed and combined in order to describe Internet documents more and more efficiently.

The Dublin Core metadata set which was defined at the 1st Metadata Workshop Dublin, Ohio in March 1995 acts as a package in the Warwick Framework. The Dublin Core is a minimal set of common descriptive elements for Internet documents, yet it is able to stimulate the spread of metadata-based tools over the Internet.

This container format can be very flexible and makes it possible to view a PHM document even without any knowledge of PHM format, and also to extend non-PHM documents step-by-step into PHM with little work. The use of the PHM format needs a new practice of creating documents rather than the use of new software.Thus the final syntax and semantics of the format should be formed as a result of a consensus and everyday practice.

4.1 Metadata in PHM

Metadata for PHM is divided into two parts. The first part is for cataloging purposes where the use of Dublin Core is recommended. Copyright can be added as a separate package as well. The second part is the management/technical description, for which an own PHM metadata package is used. This is a container and contains several packages for the description of:

Information on formats used in the PHM document
: This includes the mapping of file extensions to MIME types, and additional format-dependent informations (format subtypes, character sets, etc.). As a further improvement, links to WWW pages describing the format can be here as well.

List of scripts, applets and other executable objects
: Scripts and applets may have parameters that are to be set according to the current location of the document. Furthermore they may call other software or require additional classes or modules. Their types and required environment for execution are described here.

Links as main entry points for the PHM document
: WWW documents has distinguished pages which are essential for navigation or simply very often used. It is very likely that these are the pages where links are pointing from other places in the Internet, therefore these are called document entry points. Some examples for typical entry points: the homepage or title page, table of contents, index, or search page.

Other characteristics of files
: Other system and environment dependent options are defined here, mainly to incorporate those into the configuration of the WWW server. For example the name of directory indices (index.html), the extensions showing the language of the document, security restrictions, etc.

Relations to other PHM documents
: Possible relationships are surveyed in the next subchapter.

4.2 Relations between PHM documents

Part/Container (or parent/child) relationship
: PHM containers can be embedded. In this case the PHM document claims a Part/Container relation to its immediate container. Metadata of the container document can be inherited, overridden or repeated in the part document. By default if a particular metadata is not defined in the part document, it is taken from its containers.
: Container embedding makes it impossible that the two link rules achieve the goal of documents being freely movable. Current URLs does not allow relative addressing with respect to documents, therefore it cannot be solved that both the embedded PHM document be movable inside its PHM container and the PHM container be movable over the Internet. As a compromise the movability of the outermost PHM container is considered more important and therefore link rules apply only to the outermost PHM container, considering links pointing into embedded PHM parts as internal links. Moving of PHM parts inside the container is supported by PHM operations.

Equivalence relationship
: PHM documents are equivalent if they appear/work identically for each user (in a semantical way), though they may have different storage structures or different formats. For example if all GIF images in one document are replaced with equivalent PNG images in the other (and there are no other differences), then those documents are equivalent. Because of the semantical meaning, equivalence relationship cannot be detected automatically, rather it is declared by the author.

Master/Replica relationship
: Mirrored documents are connected to their master equivalent with this kind of relation.

Alternatives relationship
: A document may have alternatives that contain same information, but it is presented differently in language, in formats or in link structure. For example an English language HTML document may have a German alternative or an alternative in PDF or an alternative which keeps sections in one HTML file, not in separate HTML files.
: This relationship only lists the alternatives for a document, the nature of the alternatives can be discovered from the metadata of the related PHM document.

4.3 Operations and tools for PHM

Management of PHM documents is done by tools. Basic operations are system independent and form a basic set of methods that can be generally supported for PHMs. More complex management processes and operations that are dependent of the WWW server or the software environment are executed by tools. Tools can decompose their task into a sequence of PHM operations and system dependent subtasks.

4.3.1 Operations

Create/Delete
: Creates/removes the PHM container for a WWW document or a chosen directory containing HTML files. It leaves the document in place, just rewrites links and metadata in files, creates the central metadata store and fills it with automatically detectable metadata. The operations can also be used to create/delete embedded PHM subdocuments, but in that case link rewriting is omitted. The parent PHM container can be automatically detected. After the removal of a PHM container the remained directory tree can be erased optionally as well.

Move/Copy
: Moves or copies an embedded PHM document to a new location inside a PHM container. Since moving/ copying PHM documents inside a WWW server or from one server to another is environment specific, not operations but tools accomplish these tasks.

Insert/Extract
: The insert operation embeds a stand-alone PHM document into an existing PHM container at the given position. This includes the rewriting of links inside the inserted part, and the merging of metadata.
The extract operation removes an embedded PHM document from its container and makes a stand-alone document of it.
These operations make links automatically follow their target, but the insertion of new links to connect the link structure of the copied PHM documents with the world-wide link structure cannot be done automatically, neither the removal of links pointing to the old place of an extracted/removed document. In these cases warnings are issued.

Compare
: This operation finds the differences between two PHM documents, not in a semantical way, but with file-by-file comparison. So it cannot be used to detect equivalence relationship.

Archive/Unarchive
: Make/Unpack an archive from/to a PHM document. The resulting archive is easily transferable between hosts.

Filter/Merge/Split
: The filter operation takes out files with given characteristics from the document (for example French pages or JPEG images). The merge operation merges new files into the PHM container. The mapping of the files to be inserted to insert points has to be defined. The split operation creates two PHM documents from one according to the given criteria (for example the splitting of a bilingual document into two monolingual documents). In all cases the correction of the link structure is a semi-automatized task.

Verify
: Checks the document, finds spoilt links and other errors. The integrity of the metadata is investigated, too. The verification of the link structure can be restricted to internal links, but it can also detect outdated external links. The enhancement of the verify operation is the verifier tool that can find more system specific errors.

4.3.2 Tools

Basic operations are supported by PHM toolkits. An overview of more powerful tools which can extend a basic PHM toolkit follows.

An *installer* tool not only places a PHM document into the document space of the WWW server, but also reconfigures the WWW server for serving the PHM document correctly. Another variation of the installer tool would install the document for local browsing without WWW server. This tool may try to install automatically all viewers necessary for browsing the document.

Moving and copying tools are very necessary for changing the location of the PHM document. These tools apply operations (archive,unarchive), other tools (installer) and a network transfer mechanism.

A special type of copying tools are *mirroring* tools that can be used to automate mirroring tasks with the help of master/replica PHM relations. Several scenarios for exploitation of such a tool is described in [KM95]. A prototype of a simple mirroring tool based on PHM can be found in [KMS96]. In fact the idea of PHM format came from the need to handle various mirroring tasks, for example creating temporal copies of WWW structures or maintaining continuous WWW mirrors.

A *verifier* tool can check not only the link structure and the metadata, but also authentication problems, the operation of CGI scripts etc. There are many possibilities for more specialized PHM tools. For example a tool could prepare thumbnail views for large images in a document or to cut large HTML files into sections and generate the table of contents.

5.0 Using PHM in present Internet

To support the use of PHM, a format description and usage guidelines are to be created along with the implementation of a PHM toolkit that provides the basic PHM operations.

The use of PHM can offer solutions for most of the problems described in Section 2. PHM documents can be moved easily as archives from one host to another. Then the installer tool can be used to integrate the document into a WWW server. The document space of the WWW server becomes modularly manageable with the use of embedded PHM containers. PHM containers can be easily reorganized in the document space. The correct operation of PHM documents can be checked regularly and automatically.

WWW mirroring could be better organized and automated since PHM documents would know about their masters or replicas.

PHMs are a natural target for persistent naming. It is enough to maintain one URN for a PHM document to make it easily movable. It is also possible to create hierarchical naming structures. If a name of a document entry point is appended to the URN and this name is used to retrieve that entry point, then internal details of the document may be hidden. Entry point names are mapped to internal locations inside PHM.

In case of URL addressing PHM can be used for intelligent correction of spoilt links. If an URL points to a non-existing page, the PHM container for that region of the WWW server can be found and the homepage for the PHM can be offered to the user.

Authors can choose PHM as the distribution format of their documents, and use PHM tools to manage and improve their documents. As there are PHMs that are equivalently viewable from a local PC and from a WWW server, PHM could become a new alternative for nowadays used proprietary transfer formats.

WWW search engines that utilize PHM metadata could give more reliable search results than full text indexers, and could offer wider range of search criteria.

6.0 Summary

A new container format (PHM, Portable HyperMedia) was proposed to reduce the problems with handling multifile WWW documents. The PHM container does not obsolete current WWW usage and file formats, but enhances the manageability of hypermedia documents on the Web. The main advantage of this approach is the seamless integration of this new format into the current Internet without requesting the rewriting of the current tools. Portability problem of compound WWW documents is one of the most serious questions in the current Internet practice. PHM solves this problem using associated metainformation in the form that is compatible with the Warwick Framework.

7.0 Bibliography

[CGI]
Rob McCool: The CGI Specification,
URL: http://hoohoo.ncsa.uiuc.edu/cgi/interface.html

[CNRI-h]
The CNRI Handle System
URL: http://www.handle.net/

[DP]
Common Ground Digital Paper
URL: http://www.hummingbird.com/cg/

[Ferber96]
Reginald Ferber: Hypermedia and Metadata 2nd DELOS Workshop, October 1996, Bad Honneff, Germany, URL: http://www.darmstadt.gmd.de/~ferber/delos/ws2/frame/frame.html

[HTML]
Hypertext Markup Language,
URL: http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html

[HTTP]
Hypertext Transfer Protocol,
URL: http://www.w3.org/hypertext/WWW/Protocols/Overview.html

[KM95]
L. Kovács, A. Micsik: Replication within Distributed Digital Document Libraries. Proceedings of the 8th ERCIM Database Research Group Workshop on Database Issues and Infrastructure in Cooperative Information Systems, Trondheim, Norway, 1995

[KMS96]
L. Kovács, A. Micsik, G. Schermann: An Environment for Mirroring Hypermedia Documents JENC 7, Budapest, May 13-16 1996.

[LLD96]
Carl Lagoze, Clifford A. Lynch, Ron Daniel Jr.: The Warwick Framework - A Container Architecture for Aggregating Sets of Metadata, Cornell Computer Science Technical Report TR95-1558

[RFC1738]
Berners-Lee, T., Masinter, L., and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738

[PURL]
Persistent URLs
URL: http://purl.oclc.org/

[WGMD95]
Stuart Weibel, Jean Godby, Eric Miller and Ron Daniel: OCLC/NCSA Metadata Workshop Report
URL: http://www.oclc.org:5046/oclc/research/conferences/metadata/dublin_core_report.html

[WWW]
About the World Wide Web
URL: http://www.w3.org/pub/WWW/WWW/

[UR*]
WWW Names and Addresses, URIs, URLs, URNs,
URL: http://www.w3.org/hypertext/WWW/Addressing/Addressing.html

[VRML]
The Virtual Reality Modeling Language Specification, Version 2.0
URL: http://vrml.sgi.com/moving-worlds/index.html