Media Browser: An Example of Metadata-Based Browsing

Alison Lennon, Daniel Lloyd-Jones, Ernest Wan, Ken Yap, Michael Anderson, Belinda Yee
Canon Information Systems Research Australia (CISRA)
3 Thomas Holt Drive, North Ryde, NSW, 2113, Australia.
Telephone: 61-2-9805 2970
alison@research.canon.com.au

ABSTRACT

Current methods for finding relevant content, especially in media-rich web environments, suggest that metadata is critical for accurate and efficient information retrieval. We describe a Media Browser tool, which enables users to access content by visually browsing and searching metadata that is stored in a distributed fashion over the web. The basic constraint imposed by the Media Browser tool is that the distributed metadata can be represented in XML which uses XLink semantics (and preferably defined with XML Schema). The Media Browser tool enables users to access content from a single interface from which they can maintain their own information landscapes, which comprise links to metadata of interest to the user.

Keywords

Metadata, Information Retrieval, Multimedia, Information Landscape, XML Schema.

1. INTRODUCTION

Currently we use web browsers to explore and view text-based content in which links have been manually authored, and search engines to locate and explore content that has not been explicitly linked. This information retrieval process has evolved based on text-based content and is not necessarily well-suited to media rich web environments where the only means for finding non-textual content is using metadata and specific tools for accessing that metadata. We describe a general-purpose multimedia browsing and searching tool, called Media Browser. This tool integrates the current concepts of browsing and searching in a single rich browsing and searching visual interface in which the entity being browsed is metadata. Only when users have established from the metadata that they wish to view/play the content is the content accessed. Playing/viewing of the content is then enabled using media-specific plug-ins. The metadata used by Media Browser can be stored in disparate repositories on the web and must contain navigable links to the relevant content. It is preferable, though not essential, that the structure and semantics of the metadata are defined using XML Schema. Media Browser does not require prior knowledge of the schemas that define the metadata.

Media Browser visually represents metadata to users for the purposes of browsing and searching. It achieves this by differentiating between those components of the metadata (descriptors) which represent structural information about the content (e.g., a component which describes a clip of a video) from those descriptors that represent properties or index information (e.g., the date an image was captured). The structural descriptors typically contain links to either further structural descriptors or to content and are used to construct a browsable Table of Contents (TOC). The index descriptors are used to enable searching for content. The motivation behind this separation into a TOC and an index is that people are generally familiar with this concept. Also metadata creators, either consciously or unconsciously, often use this concept in their design of metadata schemas. Although the general concept of metadata-based browsing is not new, we believe our method of generating a visual representation of available metadata for the purpose of browsing and searching using a TOC and index is novel. Also the ability to enable users to browse metadata which has been defined using schemas of which Media Browser has no prior knowledge is a valuable capability.

Media Browser attempts to visually represent any XML metadata that uses W3C recommended XLink [1] semantics. However, it can provide a higher level of interpretation and thus functionality for metadata defined using XML Schema [2] for the following reasons. First, Media Browser can use the type extension/restriction and element substitution information of XML Schema to infer relationships between descriptors and thus provide more useful search results. Second, descriptor definitions in schemas can be used to provide lists of possible index descriptors for TOC descriptors. These lists can be used to construct structured query expressions and to enable metadata editing. Finally the datatype functionality of XML Schema can be used to constrain metadata editing.

Each XML metadata object fetched by Media Browser is firstly transformed into a native Media Browser description with its descriptor components being defined as either TOC or index descriptors. The transformation can be achieved by using an XSLT [3] stylesheet for commonly used metadata standards (e.g., Dublin Core [4], MPEG-7 [5] and DIG [6]). Alternatively, the transformation can be performed dynamically using a set of interpretation rules that we have defined. These rules use linking semantics (e.g., the *xlink:href* attribute) to infer whether a descriptor should be treated as part of the TOC or index for a description. The transformation step enables Media Browser to visualize metadata using a wide range of vocabularies and thus act as a general-purpose metadata-based browser.

Media Browser allows users to access legacy metadata repositories in a standards-compliant manner. The mechanism requires the content provider, or owner of the legacy database, to provide a module called a metadata server. This server receives requests for metadata using a request syntax that is based on XPath [7]. It then responds by effectively translating the stored metadata which is relevant to the request into dynamically-generated XML metadata (defined using XML schema). This means that Media Browser users can include items in their TOCs that contain links to a metadata server, the links identifying particular metadata item(s) of interest in a legacy database.

Some general-purpose media browsers exist, perhaps the best known of which is Windows Media Player 7 [8]. However typically with such browsers, the entity being browsed is local workstation content, the metadata available is limited because it is often extracted directly from the content, and the accessible media types are usually limited because of the tight coupling between content and metadata. Media Browser removes these limitations by allowing the user to browse metadata directly and by using media-specific plug-ins for the viewing/playing of content. Media Browser also provides some advantages over existing search engines by providing a general-purpose integrated browsing and searching interface and by allowing users to access remote and disparate metadata repositories.

2. Media Browser Implementation

Media Browser is implemented as a client-server application using a standard web server (Apache) and browser. The client is implemented using Flash. The server is implemented using JServ, servlets, and Swift Generator [9], with the Swift Generator tool being used to dynamically generate Flash objects from templates. The (media) content-dependent tasks of Media Browser are achieved using media tools plug-ins. Standard web browser plug-ins can be used or specific tools can be developed.

Media Browser can operate as either an intranet or internet service. In both scenarios users are required to subscribe to the service to be able to access their personal links to metadata. The Media Browser server stores a TOC and a set of stacks (collections of links to descriptors) for each subscribed user. A user's TOC consists of a tree of TOC descriptors. Each TOC descriptor is a node of an XML document that either contains child nodes or a link to a further XML document node or metadata server. The user can add or remove descriptors to their TOC and thus it comes to represent their personal information landscape, which selectively identifies internet or intranet metadata which refer to content of interest to the user. Stacks are used to store collections of links to descriptors of interest. Stacks are also used to contain the results of searches.

The predominant feature of the user interface is a browse window, which contains a grid of visual identifiers for the child descriptors of the currently-selected TOC descriptor. The visual identifiers are indicative of the linked content and may represent thumbnails or key frames, or can be generated automatically using a textual identifier (which may have been inferred as part of the metadata transformation process). The user interface provides two methods for users to navigate through their TOC. First, they can use the standard method of simply double clicking on visual identifiers in the browse window to display any child descriptors of the selected descriptor. Second, we provide a "breadcrumb", which is composed of a hierarchical sequence of TOC descriptors. Within each level of the breadcrumb users can select to pull down a menu showing the child TOC descriptors of the selected parent at that level of the breadcrumb. The main advantage of the latter navigation approach over the standard tree-based method is that it is simple to exit from one sub-tree and directly enter another without having to navigate up and down the tree.

Users can search the metadata contained in one or more descriptors from their TOC. Simple text-based and structured queries are possible. For structured queries, users are presented with a list of all the possible index descriptors that have been defined for either the selected TOC descriptor(s) or their TOC children. Currently, this requires the presence of XML Schema definitions and therefore this functionality is only available to that metadata that has been defined using XML Schema. On presentation of the list of index descriptors, the user can then specify required constraints for particular descriptors.

Index descriptors that apply to a selected TOC descriptor (and thus refer to an item or a collection of items of content) can also be viewed and edited. Editing requires that a schema definition for the descriptor exists so that the datatype of that descriptor can be appropriately constrained and that the descriptors for the item are defined to editable. Lists of relevant descriptors for the item can be obtained using the method described above for structured queries. Creation or editing of metadata is particularly useful in cases where personal media content is being accessed (e.g., personal digital images and video).

2.1 Conclusions

Any method of browsing that is suited to media-rich web environments must use metadata. It is difficult to believe that a single metadata vocabulary is sufficient to express all the concepts one would like to convey with multiple media types. However, it is possible that much metadata may eventually be represented using, or translated into, a well-accepted standard that is both easy to use and sufficiently powerful to express the desired concepts. The Media Browser tool that we describe is just one such method of browsing. By being technically based on the powerful evolving markup language, XML Schema, and by attempting to enable access to existing distributed metadata and metadata repositories, we believe that Media Browser is a step in the right direction.

3. REFERENCES

[1] S. DeRose, E. Maler & D. Orchard, XML Linking Language (XLink) Version 1.0, W3C Proposed Recommendation, 20 December 2000, http://www.w3.org/TR/2000/PR-xlink-20001220/

[2] D. C. Fallside, XML Schema Part 0: Primer, W3C Candidate Recommendation 24 October 2000, http://www.w3.org/TR/xmlschema-0/

[3] J. Clark, XSL Transformations (XSLT) Version 1.0, W3C Recommendation 16 November 1999, http://www.w3.org/TR/xslt

[4] Dublin Core Home Page, http://purl.org/DC/

[5] MPEG-7, The Multimedia Content Description Interface, http://www.cselt.it/mpeg/

[6] Digital Imaging Group Home Page, http://www.digitalimaging.org/

[7] J. Clark and S. DeRose, XML Path Language (XPath), Version 1.0, W3C Recommendation 16 November 1999, http://www.w3.org/TR/1999/REC-xpath-19991116

[8] Windows Media Player 7, http://www.microsoft.com/windows/windowsmedia/en/software/Playerv7.asp

[9] Swift Generator, http://www.swift-tools.com