Metadata and library mining: Analyzing the usage of a distributed electronic library

Matthias Will (will@informatik.uni-freiburg.de)
Institut für Informatik, University of Freiburg

April 14, 2000

1 Introduction

In the last few years, the World Wide Web has developed from an academic network into a universally used tool, serving many diverse purposes, ranging from research to electronic commerce. Institutions and companies who are offering services over a Web server need to know about the usage behavior of potential customers in order to be able to optimize their Web presentation and services. While generic Web mining tools serve the purpose to analyze the user's surfing behavior, it is often desirable to gain more specific information about the types of visited documents consulted by specific users and their interactions on a specific document repository. For these purposes, it is not sufficient to analyze the data stored in the server logfiles, but more knowledge about a server's content is needed. Furthermore, if the user's interactions with the system which go beyond requesting and retrieving a specific document are to be traced, suitable action types have to be implemented into the system which result in additional logging data when selected by a user.

A domain where these needs become apparent are digital libraries. Many publishing houses are reluctant in offering electronic publications on-line because conceiving and implementing interactive multimedia products is a costly process, and making them available to end-users is a service that publishers cannot offer for free. While discussing charging and billing mechanisms, it is also necessary to be able to determine whether an electronic product is successful or should be discontinued. While this can be determined by looking at the sales in the traditional publishing business, in a digital library, it is possible to determine the way specific modules are accessed if suitable metadata is available. Thus, a successful analysis consists of a combination of Web mining and metadata retrieval.

In this paper, we first propose a suitable architecture to incorporate metadata into a digital library. In reviewing the logging behavior of traditional Web servers, we will discuss the requirements for library mining, which cannot be met by available web mining tools. The main section will discuss the design and implementation of a suitable analyzing tool for digital libraries. We conclude by pointing out various applications for library mining.

2 Incorporation of metadata into a document repository

As the recall and precision numbers dramatically decrease while the number of Web servers still increases, the necessity to provide additional searchable information about electronic documents to help potential users to find relevant information becomes apparent. For electronic publications, providing a set of universally accepted bibliographic attributes is a suitable approach; however, as they may also consist of

multimedia modules which may require certain hardware and software prerequisites on the part of the user, the metadata will certainly be more complex than for a printed publication. In this field, the Dublin Core Metadata Element Set (see Appendix A) [Thi98] seems to be an evolving standard in which is widely recognized among librarians, publishers and researchers alike. On the other hand, some of its weaknesses have been pointed out elsewhere ([Ru98], [Mar97], [Cla97]), such as its original focus on text-only publications.

While the user access of a networked digital library is usually via a Web browser accessing a Web server, the documents themselves may also be stored in database systems. Since it is not possible to embed the metadata into an electronic publication in the general case, it has to be maintained separately. In order to store the metadata in an exchangeable format, the Resource Description Framework (RDF) [Mil98], which has also been adopted by Dublin Core, has been proposed. The metadata of a digital publication, whose primary function is to enhance its indexing and searchability, will later be needed for the library mining process. Thus, we define an electronic publication as a document tree[1], consisting of

atomic nodes: An atomic node is associated with a monolithic document, e. g. a single file, which may have been automatically generated upon a database query. Thus, atomic nodes do not have a successor.

composite nodes Several monolithic documents may be grouped together to form a composite node by grouping their associated atomic nodes. An example for this is a list of documents returned upon issuing a database query. Composite nodes may be contained within other composite nodes as well. However, a composite node may not be self-contained, i. e. circular references are not allowed.

Each node, whether atomic or composite, is associated with a set of metadata attribute sets. The Dublin Core is assumed as the lowest common denominator and compulsory. In other words, the metadata contains at least one set of attributes. However, specific applications and modules may require specialized attributes, e. g. domain-specific parameters which need descriptors beyond the Dublin Core. Another example would be rights management issues: although the Dublin Core provides a Rights attribute, it is essentially undefined. However, with sophisticated charging models, differing among publishers, a more sophisticated data model for electronic commerce is mandatory. Also, in case of a module which is derived from a copyright work, it may be necessary to provide for cascades of rights[2]. As some of the metadata sets used may not be compatible, the semantics of the metadata attribute fields may be overlapping, leading to some redundancy. This can be desirable in those cases where semantic interoperability among metadata formats should be achieved [GM96]. In order to distinguish between different metadata records, a list of available records for a specific module is kept up to date, where each line points to the corresponding record.

Upon inserting an electronic publication into a document repository, the core metadata record is stored jointly with each module in a format that is suitable for later retrieval and/or processing. In order to be reusable, a reverse mapping mechanism has to be implemented, which converts the metadata as they are stored in the document repository into its original exchangeable format. The source data from which to generate the core bibliographic record (e. g. keywords) should be provided by the author or by the creator of an electronic publication. Other descriptive metadata may be attached by any other entities involved in the publication chain, e. g. librarians who may wish to add a MARC [Marc] record, or publishers adding their rights description.

[1] If multiple views on modules of a publication are to be allowed, which may be desirable in order to offer the potential user different access modes, a single module may have multiple parents, thus generalizing a publication's structure to form a directed acyclic graph (DAG)

[2] An example would be a digital version of a reproduction of a painting, where several parties (the painter, the reproducer, and the institution who produced the digital version) may be copyright owners

3 Requirements for library mining

A record of the accesses of a Web-based digital library can be obtained by looking at the log files, which contain a list of all accesses to the server, including database queries. As the interaction between WWW servers and clients follows the HTTP protocol, which is stateless and thus records each access separately. Consequently, it does not have any notion of sessions, i. e. consecutive server accesses from the same host[3]. Even though the information recorded for each access is the same, the format in which the data is stored may vary. Here, we assume that the protocol of a HTTP query is stored in Extended Common Log File Format (see Appendix B)[4]. Unlike with conventional logfile analyzers, which count most favorite pages, most favorite accessing domain and server load, just to mention a few, the main points of interest in library mining are

- recording the access frequency of publications and publication modules
- tracking the use of publications in individual sessions
- recording the use of the built-in functions of the library system

In order to achieve these goals, the following tasks seem crucial:

- distinguishing between publications and publication modules
- sorting out automated requests effectuated by robots, spiders or worms
- grouping accessed pages and documents per sessions

The first task cannot be achieved by looking at the logfiles alone, since the URLs stored with each access request do not reveal the type of accessed document. For these purposes, it is necessary to execute a database query for metadata of the requested document, as will be explained in the next section. In order to achieve this, the structure of the metadata set has to be known as well as the query syntax to retrieve it.

4 Design and implementation of a library usage analyzer

In this section, we first introduce a number of definitions which are necessary to understand the design of the library usage analyzer that was implemented as part of a digital library project [LIB98]. Before the actual analysis, the logfiles are pre-processed in order to cover similar time periods as well as to discard the access to internal icons. The initial task during library mining is to discard irrelevant entries (see 4.2). We then describe our approach to distinguish ordinary documents from publications and their modules. Furthermore, the modeling of specific functions built into the prototype digital library and the interpretation of the user interactions are explained. We conclude this section by briefly sketching some of the results obtained.

[3] In order to maintain information about user modes, such as login and password, some servers send cookies to the accessing clients, which allow the client to re-send data to the server. In this way, it is possible for a user to access a restricted server area by providing their identification only once, even though a separate connection is opened and closed for each interaction between client and server. The version 1.1 of the HTTP protocol provides mechanisms to keep certain connections open over multiple accesses. Here, we assume that each access results in a single connection, and that a client accesses the server directly. The latter, however, not always the case if several clients access a server through an intermediate proxy server.

[4] For a detailed explanation of different log file formats, see e. g. Summary's Web Site at `http://summary.net/summary/LogFormats.html`

4.1 Definitions

- **Session**: A session is a sequence of consecutive hits, where the time span between accessing two objects does not exceed a maximum value. This implies that at least two consecutive hits are required to form a session[5].
- **Recorded session**: For each session exceeding a time span, a separate file is written. It is assumed that a meaningful session is goal-directed, which involves browsing and searching until finding a specific document or publication. Thus, only the sessions whose time span is superior to this limit are fully reported
- **User Browsing Behavior Model**: We assume that the accessed documents during a session are used for either *navigational* purposes or for *content* purposes [CMS97]. The former may also be classified as *transitory* documents as they server to guide to the modules they wish to work with. As the relevance of a document is user-dependent, a suitable classification is to measure the time span between two consecutive hits and classify a hit as a viewed document if a fixed threshold is exceeded. All other hits are classified as transitory, assuming that the user's viewing intentionality is rather low and concluding that those pages were used for either navigational purposes or that they were accessed to immediately be discarded because of insufficient content relevance.
- **Accessing time**: For all entries of a session except for the very last entry, the access time of a document can be computed by subtracting the time stamp of the current entry from the time stamp of the next entry. This means that the document which is considered last in a session has an access time which is undefined. Therefore, as the interactions with the library server are assumed as being goal-directed, the last document's viewing time is computed as the median of all previous viewing times for documents of the current session. Since the last server hit for a session is assumed to have been executed because of content relevance, its minimum access time is set to a document's minimum viewing time if there were no other documents classified as viewed within that session.
- **Publication**: A publication is either an atomic document or a document tree. Publications are either derived from printed versions, or can be declared as such by the use of bibliographic attributes. In our case, we used the Dublin Core Metadata Element Set. In the system architecture used here, it was possible to attach an arbitrary set of attributes to any kind of document on the server. Thus, only those documents were considered to be publications which had be indexed with appropriate attributes before running the analysis.

4.2 Discarding logfile entries

In order to discard robot accesses, we maintain a list of known robot types, as well as a list of known browsers. The latter list is much more reduced than the list of known robots, although both lists are likely to be incomplete, as a mechanism for an automatic update is not known. Discarding robot accesses only works if a logfile format containing information about the accessing client type, such as the extended CLF, is available.

If a server was accessed from a host where the accessing client is known to be a robot from looking at the robots resource file, it is immediately discarded and counted as a robot access. If the host's client is detected as a robot in a line read from the logfile and not matching the list of known browsers, it is added to the list of robots as detected by the system. If the client is not in the list of known browsers, it is counted among

[5] If this value is set too high, it may happen that multiple sessions are merged into one. In order to partially circumvent this, an idea was to force the start of a new session when a change of the type of accessing client was detected. However, this would have only valid under the assumption that, within a period of time declared to be a session, there would only be one accessing client for each detected host.

unknown accesses; a list of unknown clients is written out by the program. All other hits are assumed to be browser accesses and processed for further analysis.

A list of document types which should be ignored (e. g. image documents) can also be supplied, as well as a list of server object names. If, following this process, several URLs are found in the logfile entry referencing the same document, only the first occurrence is considered. The skipped documents are then counted as *duplicates and non-documents*. Also, certain document classes can be discarded by looking at a document's MIME type and matching it against the list of unwanted document types.

4.3 Ordinary documents, publication modules and their use

In analyzing the use of the electronic library, we want to focus on the use of its publications. As the URLs of the retrieved documents do not reveal their document type, we query each object corresponding to a distinct URL for three of the fifteen Dublin Core attributes. These are

Membership For each module belonging to a publication, this field indicates the name of the entry collection for that publication. The entry point for that publication, i. e. the collection, does not contain this attribute, as it is only a container for individual modules. Ideally, this should be an identifier unique to a specific publication. For atomic publications, i. e. those that only consist of one single document, this field is identical to the **Name** attribute, thus pointing to itself

Identifier Each printed publication usually contains a unique identifier, such as ISBN[6] for books, ISSN[7] for periodicals, or ISMN[8] for sheet music. For digital publications, the Digital Object Identifier may be an option[9]

Publication Title This field is provided to allow for a publication's official record as opposed to the corresponding object's title. Usually, this field contains the authors' name and the publication's title.

For every server object occurring in a logfile which is being analyzed, its `Membership` attribute is checked. In this case, it contains a locally unique server object name, which reveals the document root. The title of the publication is then found by retrieving the value of the field `Publication Title` or, if not available, from the title of the server object, as presented to the user.

The bibliographic information about publication modules is not only used to generate a detailed list of accessed publications and its modules for each session, including access duration, but also to associate specific clients and their use of publications, as well as statistics about the sessions that accessed specific publications. Furthermore, a ranking of most popular publication modules is generated. In order to analyze the use of publications over a longer time period, we incrementally update a record of statistical parameters such as number of accessing sessions, viewed and accessed modules and viewing times associated to a fixed time span. This record is then used to produce a diagram showing the usage behavior related to specific publications.

4.4 User interactions

The digital library whose usage was analyzed contains many special built-in features, such as navigational buttons, full-text search, editing of user-specific document collections, online help and the selection of user-dependent preferences and others. For each feature invoked by a user, a suffix, containing a string

[6] International Standard Book Number
[7] International Standard Serial Number
[8] International Standard Music Number
[9] As most of the publications in our prototype digital library were derived from printed versions, we used the ISBN or ISSN number. For original electronic publications who lacked this number, we used a locally unique identifier.

describing the type of interaction as well as corresponding attributes, is appended to the address of a server object and stored in the corresponding HTTP request. Thus, a classification of the frequency of built-in commands usage is possible. In order to transform the suffixes of the request, a mapping mechanism which associates suffixes with human-readable command descriptions is used. Thus, the suitability of the user interface in terms of functionality can be judged, e. g. if multiple calls to help topics occur, the usability is likely not to be straightforward. Also, the examination of fulltext search parameters reveal the type or field of publications that library users are searching for, thus giving a hint on how to possibly improve its popularity. Additionally, by looking at the documents consulted after a search, it is possible to judge whether the search results were satisfactory.

4.5 Results

The rankings that were produced from the analysis included the viewing time of individual documents, the duration of individual sessions as well as the number of transitory and consulted documents per session. We found that the overall percentage of retrieved publications was rather low, which may be explained by the lack of content which goes beyond an equivalent of printed publications. Among the consulted publications, we found that reference works and journal or conference proceedings were consulted most. As for textbook-like publications, we found that duration of use was rather low, but that many modules were accessed. This backs up our assumption that electronic publications are mostly consulted to look up specific matters or to get a first impression of a book that is also available in printed form. We also assume that journal articles are mostly accessed to print them out, although we do not know for sure as the printing function does not classify as an interaction between server and client and thus cannot be analyzed. We found that the most commonly used library functionality was the search function, whose importance increases with the volume of publications in an electronic library.

5 Conclusion

Numerous projects in the field of digital libraries have shown that while it is possible to offer sophisticated frameworks to tackle some of the problems that arise when offering copyright publications in distributed networks, such as charging modes or unifying heterogeneous storage modes, it is difficult to provide a substantial body of multimedia publications. Additionally, material which is relatively new may be offered in digital format, while the effort to process material only available in printed form is generally far higher than the obtained value. Therefore, digital libraries will mostly supplement traditional libraries in the following two aspects:

- providing multimedia publications which involve dynamic and interactive components, such as audio, video and simulations, as supplements to paperbound material
- providing out-of-print publications in electronic form

As the storage of publications may vary (local vs. global access, databases or file systems), in addition to their accessibility (access modes may differ according to type of publication and/or publishers), a a digital library has to hide the diversification of the underlying repositories in a consistent user interface. This also applies to the retrieval of information about non-digital publications. In this case, the corresponding electronic document would be an automatically generated record of bibliographic information generated from the meta-data associated with the publication. A possible benefit would be that by detecting repeated access of metadata from a publication which is only available in printed form, publishers may consider producing an electronic edition in order to profit from high demand.

As it was shown here, it is essential to be able to retrieve both structural and organizational information of an electronic library, in order to obtain detailed results about its usage. In a prototype implementation, we were able to demonstrate that by combining web statistics approaches with metadata retrieval, it is indeed possible to obtain a detailed result about the usage of electronic publications in a distributed digital library. We proposed a general, extensible model for metadata storage at a conceptual level. As the document repositories may vary, appropriate mapping techniques to produce queries resulting in the retrieval of the appropriate metadata need to be developed. Two related issues, namely rights management and models for electronic commerce have only briefly been mentioned here, but these need to be considered as well if the digital library sector is to grow into the commercial publishing market. A detailed usage analysis, as outlined here, is crucial to answer some of the questions raised concerning the acceptance of digital libraries in the general public.

References

[Arm96] W. Arms et al.: *Uniform Resource Names. A Progress Report*. D-Lib magazine, February 1996. Available at `http://www.dlib.org/dlib/february96/02arms.html`

[Ber98] P. Berinstein: *DOI: A New Identifier for Digital Content*. Searcher: The Magazine for Database Professionals, Vol. 6, No. 1, January 1998. Available at `http://www.infotoday.com/searcher/jan/story4.htm`

[Bid98] M. Bide: *In Search of the Unicorn. The Digital Object Identifier from a User Perspective*. BNBRF Report 88, February 1988. Available at `http://www.bic.org.uk/bic/unicorn2.pdf`

[Cat98] W. Cathro: *Metadata: An Overview*. Standards Australia Seminar, August 1997. Available at `http://www.nla.gov.au/nla/staffpaper/cathro3.html`

[Cla97] R. Clarke: *Beyond the Dublin Core: Rich Meta-Data and Convenience-of-Use Are Compatible After All*. July 1997. Available at `http://www.anu.edu.au/people/Roger.Clarke/II/DublinCore.html`

[CMS97] R. Cooley et al.: *Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns*. Proceedings KDEX '97, November 1977 Available at `http://maya.cs.depaul.edu/ mobasher/papers/webminer-kdex97.ps`

[GM96] J. Godby, E. Miller: *A Metalanguage for Describing Internet Resources Using the Dublin Core Element Set*. Proceedings INET '96. Available at `http://www.crim.ca/inet96/papers/a2/a2_2.htm`.

[LIB98] R. Stubenrauch et al.: LIBERATION - *A Value-Added Digital Library*. Proceedings ECDL '98, Heraklion, Crete, Greece, September 1998, Springer LNCS 1513, pp. 647-648. Available at `http://www.iicm.edu/stubenrauch/ECDL98`.

[Mar97] D. Martin: *Beyond Dublin Core. The Need for High-Quality Product Information*. BIC/BL Seminar on Trading Electronic Content, March 1997. Available at `http://www.bic.org.uk/beyonddc.rtf`

[Marc] Library of Congress Network Development and MARC Standards Office: MARC *standards*. Web site at `http://lcweb.loc.gov/marc/marc.html`

[Mil98] E. Miller: *An Introduction to the Resource Description Framework*. D-Lib Magazine, May 1998.
Available at `http://www.dlib.org/dlib/may98/miller/05miller.html`

[Ru98] G. Rust: *Metadata: The Right Approach. An Integrated Model for Descriptive and Rights Metadata in E-commerce*. D-Lib Magazine, July/August 1998.
Available at `http://www.dlib.org/dlib/july98/rust/07rust.html`.

[Sha96] K. Shafer, S. Weibel, E. Jul, J. Fausey: *Introduction to Persistent Uniform Resource Locators*. Proceedings INET '96.
Available at `http://purl.oclc.org/OCLC/PURL/INET96`

[Thi98] H. Thiele: *The Dublin Core and Warwick Framework: A review of the literature. March 1995 - September 1997*. D-Lib Magazine, January 1998.
Available at `http://www.dlib.org/dlib/january98/01thiele.html`

Appendix A: The Dublin Core Metadata Set

In Dublin Core, there are fifteen fields which form a set of metadata for publications. Each of these fields may contain subdivisions for a more precise specification. In particular, each field may contain a *Canberra Qualifier*, which serves as a hint how to interpret its value. In this way, implementations of the Dublin Core may vary even though the semantics of the basic attribute fields are defined in the standard. Depending on the application, only a subset of the attributes may be used. The full set of elements is as follows (see also [Cat98]):

Title: The title of the publication, as given by the author (see field **Creator**) or publisher (see field **Publisher**).

Creator: The name of the person being primarily responsible for the intellectual content of the publication.

Subject: A list of keywords describing the subject or content of the publication.

Description: A short description or the abstract of the publication.

Publisher: The name of the entity responsible for making the publication available. This is usually the publishing house, but as the as the roles of authors, content providers and publishers are redefined, other options are possible (e. g. multimedia courseware where an academic institution is responsible for its availability)

Contributors: Other persons having contributed to making the publication available besides the person specified in the **Creator** field

Date: The date of making the resource available in its present form or version

Type: Resource type, such as technical report, preprint, dictionary or journal

Format: Document format of the publication module; may be seen as a rough equivalent to Internet Mime Types

Identifier: A globally unique identifier of the resource, independent of its storage. Examples are digital object identifiers (DOI) [Ber98], uniform resource names (URN) [Arm96] and persistent uniform resource locators (PURL) [Sha96]

Source: Name of the publication from which its electronic form was derived (if applicable)

Language: The Z39.59 three character code of the language of the resource

Relation: If the resource has some relation to another resource, e. g. being a chapter of a publication, a diagram of an article, this field indicates its primary resource. Otherwise, in case of a publication, this field's value is equivalent to the **Name** attribute.

Coverage: Spatial locations and temporal durations characteristic of the resource (*experimental*)

Rights: Copyright notice or some other means of allowing providers to associate terms and conditions with a specific resource

Appendix B: The Common Log File Format

In the Common Log File Format (CLF), each logfile entry consists of the following fields:

- IP address or host name of the remote computer
- a dash
- the login name of the user
- the date and time of the corresponding server hit
- the request to the server, including the document name and the version of the HTTP protocol used for transmission
- the error code returned by the server
- the number of bytes sent to the client

The extended common logfile format additionally contains

- the URL of the referring page, i. e. the document which was seen previously to the currently requested entry
- the name of the accessing software (usually a WWW browser or a robot)