The XML Log Standard for Digital Libraries: Analysis, Evolution, and Deployment

Marcos André Gonçalves, Ganesh Panchanathan, Unnikrishnan Ravindranathan, Aaron Krowne, Edward A. Fox
Virginia Polytechnic and State University
Blacksburg, VA, 24061, USA
{mgoncalv, fox}@vt.edu

Filip Jagodzinski, Lillian Cassel
Villanova University
Villanova, PA 19085-1699
+1-610-519-7341
{filip.jagodzinski, lillian.cassel}@villanova.edu

ABSTRACT
We describe current efforts and developments building on our proposal for an XML log standard format for digital library (DL) logging analysis and companion tools. Focus is given to the evolution of formats and tools, based on analysis of the deployment in several DL systems and testbeds. Recent development of analysis tools also is discussed.

Categories and Subject Descriptors
H.3.7 [**Information Systems**]: Information Storage and Retrieval – *Digital Libraries*. H.2.7 [**Information Systems**]: Database Administration – *Logging and recovery*.

General Terms
Measurement, Design, Human Factors, Standardization.

1. INTRODUCTION
In 2002 we proposed an XML log standard for digital libraries, (DLs) and companion tools for storage and analysis [1]. The goal was to minimize problems and limitations of web servers, search engines, and DL systems log formats (e.g., incompatibility, incompleteness, ambiguity). Accordingly, our new format and tools allow capturing a rich, detailed set of system and user behaviors supported by current DL systems. In this paper, we report advances based on analysis of experimentation and deployment in several DL systems and testbeds. We hope that discussion of this work will move the community toward agreement on some DL log standard, which is urgently needed to support scientific advance.

2. EVOLUTION OF THE LOG TOOL
The evolution of the log tool is illustrated in Figure 1. The first version had a monolithic architecture, which was strongly coupled within the target system. Whenever DL events needed to be logged, the client invoked the corresponding methods of the log tool, since specific calls had been inserted within the target system. The first tests were performed with the MARIAN DL system [2]. This implementation revealed two major drawbacks: 1) small changes in the format required complex changes of the DL logger code and complete recompilation of the tool and target system, therefore preventing extensibility; and 2) the Java-based implementation and close coupling required a deep understanding of the target tool architecture and caused problems in connecting the tool with DLs implemented in other languages (e.g., Perl), therefore preventing wide-spread adoption.

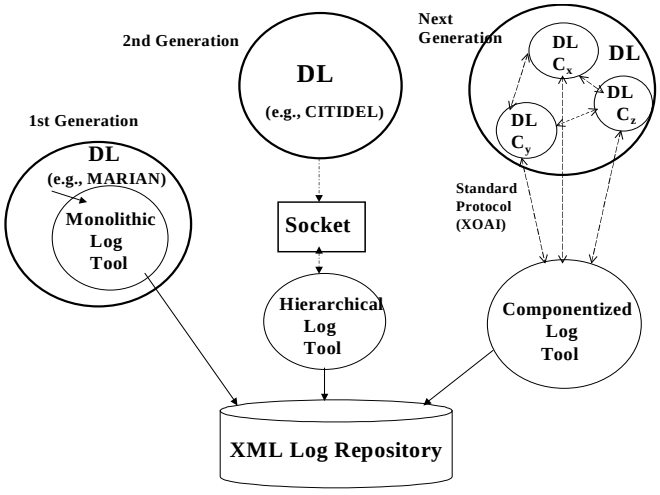

Figure 1. Evolution of DL Log Tool (Key: C_i= component).

Our second generation implementation solved those problems by 1) re-implementing the tool with an OO hierarchical, bottom-up design that mimics the organization of the XML schema of the log format, therefore making internal communications clearer and isolating points of communication and modification; and 2) de-taching the tool from the target DL system by using connectionless sockets. For socket communication, we devised a simple, ad-hoc datagram packet format.

Our next generation DL logger will enhance this communication by allowing direct, peer-to-peer communication between DL components and the (componentized) log tool. Following the philosophy of the Open Archives Initiative [3], we intend to use standard (or slightly extended) lightweight protocols, to allow this direct communication, therefore promoting interoperability and reuse. In particular, the extended OAI (XOAI) set of protocols defined by the ODL approach [4], provides specialized OAI protocols for several DL services and can serve as a foundation for such communications.

3. EVOLUTION OF THE LOG FORMAT FOR DL SERVICES

To be useful, the DL log format has to be reflective of how a generic DL system behaves. Accordingly, we have designed and organized the log structure in accordance with our 5S theory for digital libraries [5]. In 5S, services are composed of scenarios, which describe service behavior through sequences of specific user and system events. Since we are mostly interested in understanding user interactions and the perceived value of responses, we have chosen to record only the initial user input and final service output events along with corresponding parameters (modeled as XML sub-elements of events), and ignore most of the internal system communications (except administrative information).

The Computing and Information Technology Interactive Digital Educational Library (CITIDEL) [6] constituted the first large-scale application of our second-generation tool. From the beginning, it was clear that the log format, which currently models only storage, searching, and browsing services, as well as administrative information, was not able to capture all the CITIDEL interactions, given the rich set of requirements and services offered by CITIDEL. Therefore, extensions of the format were required. According to the 5S philosophy, extensions regarding new service events are to be modeled by analyzing user inputs and system outputs. Table 1 shows the current and in-development services and input events supported by the log format. The table also connects log events with Open Digital Library (ODL) [4] components which: 1) currently or in the near future will implement services in CITIDEL; 2) provide the necessary underlying protocols for communication between the DL services and the next generation XML DL logger.

Table 1. DL service, log event, ODL component

Service	XML log event and sub-elements	ODL component
Searching	Search (Collection, SearchBy(Field), QueryString)	IRDB
Browsing	Browse(DocInfo(PathName,DocID, Collection), Category, SortOrder)	ODL-Browse
Storing	Update(AddInfo(DocInfo))	Box
Annotat-ing	Annotate (AnnotateInfo (AnnotationID, DocInfo))	ODL-Annotate
Recom-mending	Current being modeled	ODL-Recommend
Rating	Current being modeled	ODL-Rating
Reviewing	Current being modeled	ODL-Review
Filtering	Filter (criteria (expression), UserId)	Filter

4. LOG ANALYSIS TOOLS

Standardization of the logs will ideally lead to a standardization of their processing/analysis. Accordingly, we are developing several analysis modules and tools, designed for easy expansion. The primary component of the log analysis tools is the log line parser that sends the content of each log line to an appropriate module. A module increments appropriate variables, populates files that are intermediate aggregate statistics of key log features, and performs a host of other required actions. The modules that are already developed track browse and search requests for each resource, maintain a record of the number of accesses from each domain; keep statistics on the words used in all the search queries; record the number of hits and logons per day, month, year, etc.; and keep track of the number of times that various tools and CITIDEL provided resources have been utilized.

The design of the log analysis tools is highly object oriented, with little or no coupling between modules. The design makes modification and creation of new modules very easy. In the case where a novel statistic is required or in the case that a new XML format feature is added, a new module can be built and connected to the already existing set of modules.

The modular design of the log analysis tools also will allow for more advanced analysis capabilities to be integrated into future versions. The current document search and browse output statistics provide information about the total number of hits for each document as well as a breakdown of hits based on aspects of the server domain. We are extending these output statistics so that we can see the clickstream path of users through the website, which will allow us to identify bottleneck pages and features. We also are developing an analysis query system that will address the large combinatorial possibilities that result from the union of multiple log variables. The query system will mine the intermediate statistics files to get a specific result.

5. CONCLUSIONS

As expected with any newly proposed standard, evolution to cope with results of the early stages of experimentation is expected. Accordingly, our formats and tools have evolved to deal with the results of such experiments. With the interest demonstrated by many DLs and institutions (e.g., CiteSeer, MyLibrary, Daffodil) in adopting the format and tools, we expect soon to release stable versions of both. Once this phase is achieved, other research issues will become the focus of future efforts, such as richer analysis and evaluation, and efficient use of distributed storage.

6. ACKNOWLEDGMENTS

Thanks are given for the support of NSF through its grants: IIS-9986089, IIS-0002935, IIS-0080748, IIS-0086227, DUE-0121679, DUE0121741, and DUE-0136690. The first author is supported by CAPES, 1702-980.

7. REFERENCES

[1] M. A. Gonçalves, M. Luo, R. Shen, M. F. Ali, E. A. Fox: An XML Log Standard and Tool for Digital Library Logging Analysis, ECDL 2002, 129-143, Rome, Italy.

[2] M. A. Gonçalves, P. Mather, J. Wang, Y. Zhou, M. Luo, R. Richardson, R. Shen, L. Xu, E. A. Fox: Java MARIAN: From an OPAC to a Modern Digital Library System. SPIRE 2002, 194-209, Lisbon, Portugal.

[3] Open Archives. http://www.openarchives.org

[4] H. Suleman. Open Digital Libraries. Ph.D. dissertation. Virginia Tech, Department of Computer Science. Nov. 2002

[5] M. A. Gonçalves, E. A. Fox, L. T. Watson, N. A. Kipp. Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries. VT Tech. Rep:TR-03-04

[6] CITIDEL. http://www.citidel.org.