A Deposit for Digital Collections

Norman Noronha[1], João P. Campos[1], Daniel Gomes[1], Mário J. Silva[1], and José Borbinha[2]

[1] Faculdade de Ciências, Universidade de Lisboa, Portugal
{normann, jcampos, dcg, mjs}@di.fc.ul.pt
[2] Biblioteca Nacional, Portugal
jose.borbinha@bn.pt

Abstract. We present the architecture and requirements for a novel system for managing the deposit of specific genres of digital publications in a deposit library. The system adopts a simple model for online publications and supports both harvesting and delivery models of deposit. This paper describes that system, and presents an evaluation after a trial period with the harvesting functions.

1 Introduction

Publications are changing from the traditional formats, like paper magazines, to digital media, such as online news feeds. In addition, everyone with a connected computer is now a potential publisher. This is increasing the number of new publications, making the management of their deposit more complex.

The deposit and preservation of publications has a significant role in preserving the historical past. Publications in traditional media have been archived since ancient times. However, archiving the Internet aiming for long term preservation is a non trivial task [3]. The tools used for building digital publications have not been designed with preservation in mind and so do not meet most of the preservation requirements. In addition, publishers on the Web do not have a tradition of sending copies of their digital items to library deposits for archival. However, the Internet makes it possible for librarians to harvest copies of publications on the Internet. There are efforts to archive the entire web [17]. These are of marginal relevance to librarians, as most of the collected information has no historical interest. Libraries need the tools to selectively choose the electronic publications of great interest to collect and preserve [4]. Our research goal is to evaluate how these publications could be collected for preservation.

The Networked European Deposit Library (NEDLIB) [5] is an intiative to develop a common architecture and basic tools for building deposit systems for electronic publications. Practical experiences, technical infrastructures and organisational approaches taken by individual NEDLIB partners are gathered and compiled in such a way that these experiences can be of use to other libraries [6]. The National Library of Portugal is a one of the partners in NEDLIB.

We have developed an initial framework to select relevant publications from the Internet, retrieve their contents over time and provide easy methods for accessing the collection to the general public. Access methods include a service to

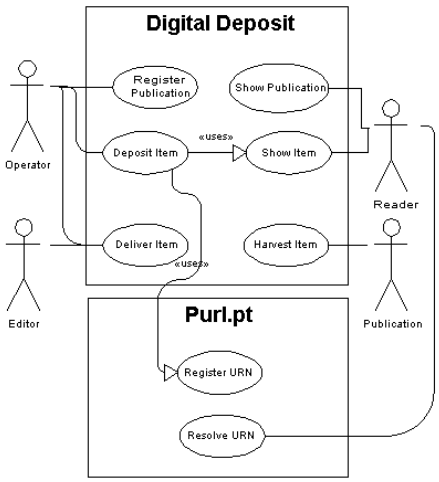

Fig. 1. Our framework for digital deposit and access to deposited copies of electronic publications is composed of two subsystems, Digital Deposit (DD) and PURL.PT. An *operator* registers *publications* with the DD system. Each of the *items* of a *publication* is either *harvested* by or *delivered* to the system. Collected items are then verified by an *operator* that either discards or accepts them in the *deposit* use case. A Universal Resource Name (URN) [7] is then assigned to each deposited item. Any user that later wants to access that item gives its URN to PURL.PT, which *resolves* it into an URL that references the storage location of the collected publicaton

resolve Universal Resource Names (URNs) [10, 7] into the individual collection items. This development is a part of DROP - Deposit of Online Digital Publications, a project for building a digital repository of all Portuguese Internet publications of historical interest jointly developed by the National Library of Portugal and the University of Lisbon. This paper presents the system we designed for retrieving copies of publications from the web, storing and accessing them from our repository.

Unlike Web Search engines, we do not intend to collect everything published on the net, but only those sites that librarians classify as historically relevant. The goal of DROP is to provide means for identifying, selecting and retrieving bounded and well defined publications.

Figure 1 illustrates the use cases of the system. Our framework consists of two subsystems:

Digital Deposit (DD) – registers publications, harvests, accepts the delivery, verifies and deposits items.

PURL.PT – registers and resolves URNs [7, 15] into URLs that represent the address of the collected publication. This is similar to the temporary solution taken by the PANDORA Project to resolve a permanent name [9, 8].

We support two methods for retrieving publication contents: delivery and harvesting. In the delivery case, someone, either a system operator or a publication editor, submits the contents of a publication directly to the system. In the harvesting case, a previously registered publication is copied from its home site on the Internet into the DD (possibly periodically).

The deposit of received items in our system starts after the harvesting or delivery of their elements. Before integrating these contents in the DD, an operator inspects the collected items; inspection may involve viewing the data or using some high level verification procedures. After evaluating the contents the operator may decide that it is not valuable, and discard it, or that it should be preserved, and accept it

Once an item is deposited, it may be accessed by the external readers of our collection. When a reader wants to access some publication referenced by a URN within the namespace controlled by the system, PURL.PT resolves the URN either into a reference to a publication within the deposit, or into a reference to the site on the Internet. If the reference points to the deposit, the reader will engage in the show publication use case.

We have an initial implementation of the DD and PURL.PT systems. This paper presents their design options, implementation approach and some of the statistics for the collection of Portuguese periodic publications that have been harvested with our system. It is organized as follows: section 2 explains the requirements for the various functions of the system; section 3 presents the architecture and documents the implementation options; section 4 presents some of the statistics for the collection of the periodic publications that have been harvested with our system. Finally, in section 5, we present our conclusions and directions for future work.

2 Requirements

A deposit system for a digital collection must support multiple, sometimes conflicting requirements. We discuss the main requirements that we have identified for the digital deposit.

Persistency of the Entry Points - The system has to provide persistence of the entry points. As URNs [7] provide this kind of persistency, we assign URNs for resources following the National Library namespace draft proposal [1]. Each item stored within the DD has an assigned URN.

We support this as a general service that translates URNs in our namespace to their associated URLs on the Internet.

Publications Registry - Registering a publication needs to be a simple process, where a human operator, after identifying a publication to include in the collection creates a record indicating what URLs have to be collected and the retrieval method to use: harvesting or delivery.

DROP is part of a larger system within the National Library of Portugal. Publication records are not meant to overlap any existing data within bib-

liographic cataloging systems in use [11], but rather to specify meta data to be used when retrieving and processing items to be collected.

The publication title and the algorithm to be used to generate URNs for each of the items are attributes that must be specified in every record. If items are added to the collection through harvesting, additional information is necessary, such as where to start harvesting and the specification of the data formats to be transferred. Periodic publications have more specific attributes, including their periodicity and schedule for fetching new items.

DD operators also must be able to edit these records to make the necessary updates as the properties stored for harvested publications change.

Harvesting - The publications included in our collection are documents organized as sets of URLs that are available on the Internet. Some of these publications are published as Postscript or PDF files, some others as HTML documents linked to GIF or JPEG images.

The main focus in harvesting these publications is respecting restrictions imposed by the collection administrator on the contents to gather. Our harvester enables restrictions by domain name, depth of harvesting (number of links to follow), MIME types of the documents to retrieve and maximum size of the documents.

The harvester also can be programmed to collect every item of periodic publications and to retrieve their contents once published, before they are removed from the Web where they were originally published.

URN Assignments - The system generates an associated URN for every publication and item it manages. The URNs are generated by an algorithm that is specific of each publication.

The URN generation algorithm in DROP follows the policies defined by the National Library of Portugal [1].

Items of a periodic publication collected from the same space (same URL) but at different points in time will result in separate items with different URNs. A general URN can be used as a reference point to the last collected item of a periodic publication.

Publications Delivery - Delivery is another supported mode for importing contents into the system. In this mode, publishers or other agents push information into our repository. An operator, the publication editor or an author, manually collects all its files/URLs and submits them to the system. The delivery process associates the received data with the meta-data available for the publication (from the publication record).

Delivery can be partially automated by distributing software to editors for submitting new items. We have established initial contacts with some of the most prominent publishers in our culture for joint development of processes for automating the deposit of their digitally published materials. However, we fear that editors will show small interest in installing delivery software, as the only advantage they will get from doing it is the persistent depositing, which is of low business value in the short term.

Deposit - In our system, data collected from publications web sites is not stored permanently in the repository. Before deposit, data stored through

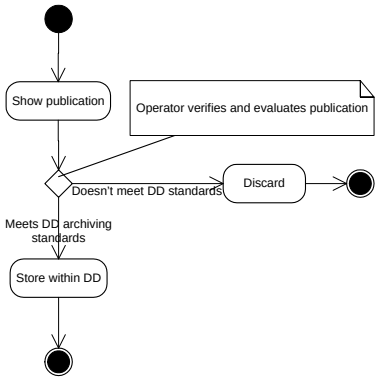

Fig. 2. Activity diagram for the deposit operation: deposit starts by showing the publication contents to the operator for verification. If it meets the depositing standards, it is then stored in the DD

harvesting or delivery is just candidate data. During deposit, this data can be either discarded (because it does not meet the requirements to be part of the collection), or stored permanently (figure 2).

Resolving a URN - The resolve operation is accomplished by the PURL.PT subsystem. When a reader calls the resolve operation, he is redirected either to our deposited copy of the item or to the original publishing site on the Internet. The decision of where to redirect the reader is based on the following criteria:

- permission to republish the resource
- location of the reader: local or remote

Redirection is supported directly by the HTTP protocol, and is used in our system so that readers can have their browsers point to the item intended when giving a URN to the PURL.PT server without intervention.

If the reader is redirected to a collected copy of the publication, it will ask the DD for the item requested. The copy displayed should look as similar as possible to the original. In particular, the reader should be able to browse collected items in the same way as he does with the original.

Storage - Persistent storage of digital contents is a complex problem with multiple perspectives and many pitfalls [12]. Our current work is on the building process of repositories for digital collections and providing easy access methods. Our assumption is that, in a production environment, our framework will operate upon a storage system that provides information preservation guarantees.

3 Architecture

DROP has been modeled as two main subsystems, to reflect the two different main functions:

Digital Deposit is the system that registers, collects and shows publications; **PURL.PT** is the system that registers and resolves URNs.

In this section, we present class and component models of our architecture.

3.1 Class Model

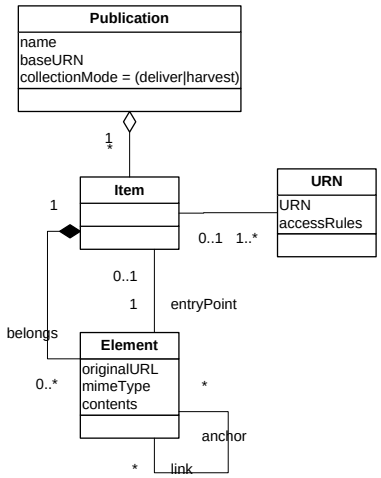

Fig. 3. Class diagram for the system. Publication objects represent publication records. Items represent collected editions of publications. Items are made of elements: elements are the objects that compose an item. Items are assigned URNs when deposited

The class model of our collection repository is represented in figure 3. A *publication* describes online publications in the digital deposit. Each publication has a *name*, a *baseURN*, specifying how to generate URNs for new items of the publication and the collection mode used to import it into the system.

Item represents the saved copies of monographs or issues of periodic publications in the collection. Each *publication* may have multiple associated items. The system assigns a URN to each item at deposit time. System operators may assign additional URNs to the same item.

Each *item* is composed of one or more *elements*. An *element* represents the contents of an URL downloaded from the Web or delivered to the system. *Original URLs* of elements are saved to enable the reconstruction of relative URLs linked from them. *MIME types* will indicate to the final user what interpreter to use when decoding the data. Links and anchors between documents are already embedded in the *contents* and need not be saved as meta-data.

URNs reference resources. Each deposited item must be assigned a URN. URNs may reference several copies of the same resource to direct the user to the appropriate resource provider, according to some defined *access rules*.

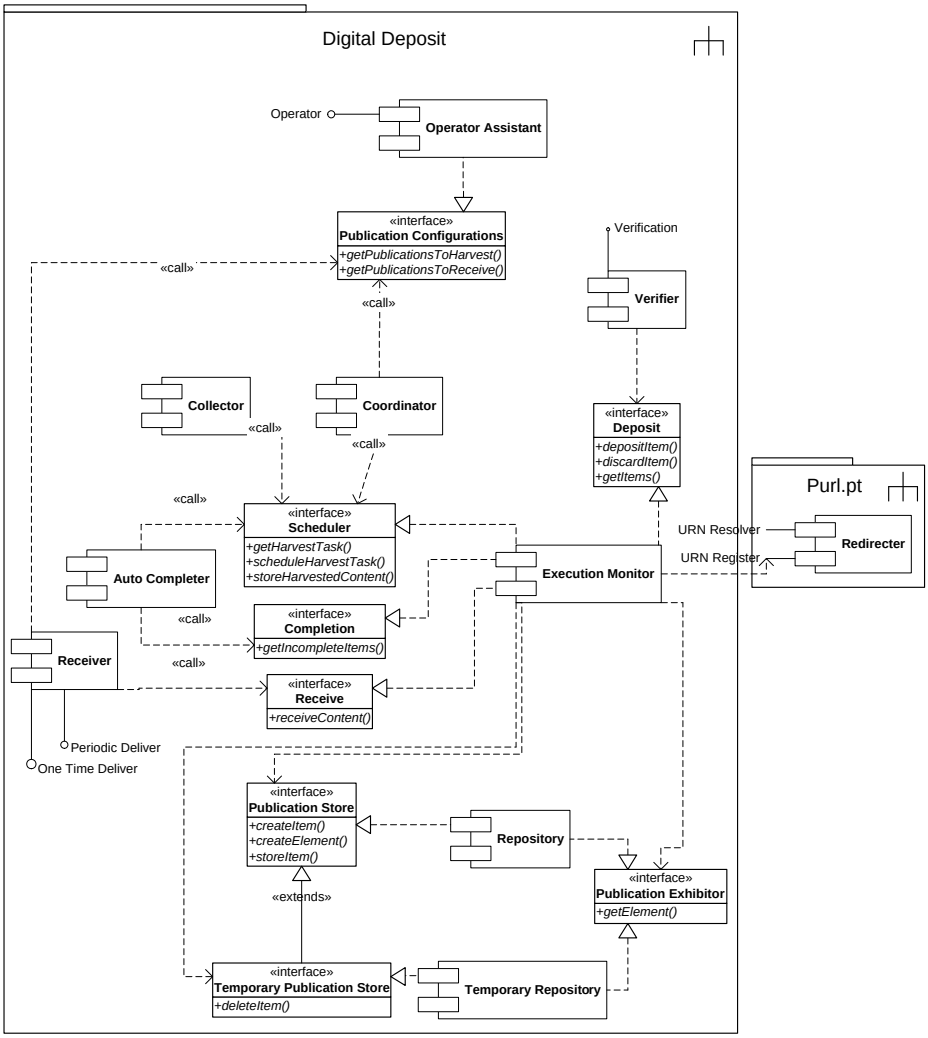

Fig. 4. Component diagram representing DD and PURL.PT subsystems. For the sake of simplicity only main operations are depicted. Component attributes are not represented

3.2 Component Model

Our system architecture has the following main components (Figure 4 represents its component diagram):

Operator Assistant – saves publication records and provides an operator interface to register publications.

Coordinator – interprets publication records and schedules harvesting tasks.

Collector – gets harvesting work units from the execution monitor and retrieves the specified contents from the Web.

Auto Completer – is notified of possible incomplete items, and schedules a task to retrieve contents to complete the items.

Receiver – receives items from users through one of the two interfaces provided and inserts them in the system.

Verifier – provides a user interface to verify, evaluate and deposit or discard publications.

Repository – saves deposited items. Offers an interface to save items and an interface to retrieve them.

Temporary Repository – works like the repository but keeps temporary items, already delivered or harvested, but not yet deposited. The interface allows deletion of items.

Execution Monitor – orchestrates the concurrent execution of the other components. Provides interfaces for scheduling harvesting tasks, retrieve incomplete items, save delivered items, and deposit accepted items. Saves contents in the temporary repository and moves them to the repository (or deletes them). Registers URNs for deposited items.

Redirecter – saves URN resolving data. Offers interfaces for registering and resolving URNs.

The system components work together as follows:

- External operators register publications to collect with the *Operator Assistant*. Periodically, the *Coordinator* retrieves publication records to harvest from the *Operator Assistant* and schedules the tasks needed to gather them with the *Execution Monitor*.
- When the *Collector* is free to gather more data, it asks the *Execution Monitor* for work. It then gets the harvesting tasks, retrieves data from the Web and requests the *Execution Monitor* to store it. The *Collector* is designed for quick execution and does not try to recover from failures, it just logs them.
- The *Auto Completer* checks if all the retrieval tasks were successful. If not, and if items can be completed by retrieving the remainder of the contents from the Web, the *Auto Completer* re-schedules the appropriate tasks.
- The *Receiver* waits for a publisher to deposit contents through one of the provided interfaces or if this is a delivery of an item of a periodic publication, the receiver gets the publication meta-data from the *Operator assistant*. If a publication is delivered for the first time, the publisher must insert the data required to register the publication.
- The *Verifier* enables the evaluation of the items and provides an interface for depositing or discarding them. It calls the necessary interfaces on the *Execution Monitor* to complete these tasks.
- The *Execution Monitor* controls the contents flow within the system. It receives the contents from the *Collector* or from the *Receiver*, stores them in the *Temporary Repository*, guides the user to its evaluation and either moves it to the *Repository* or deletes it.

- The *Repository* saves items and allows access to them through the publication exhibitor interface.
- The *Redirecter* keeps all the data needed to reference users to the resources designated with the URNs presented for resolution. It collects this data when it is entered through the register URN interface, and uses it to resolve calls on the URN resolver interface.

The components are hosted in two separate subsystems so that one does not depend on the other: one might want to keep the PURL.PT and discontinue DD (if a better solution is offered on the market).

3.3 Implementation

In our implementation, we store items and elements in the file system of the server that hosts the DD: an item is a directory and all the elements that compose it are files within the directory. This structure may not reproduce the original file structure on the publications web site, as one element may be part of multiple items in the original site. However, it makes it easy to handle items as collections of files that can be detached from our file system and later accessed from a Web browser.

Additionally, publications harvested into the file system can be accessed from a *publication exhibitor* interface.

All the meta-data needed by the execution of our system is maintained in a PostgreSQL database[13].

The redirecter is implemented using the HTTP redirection mechanisms, on an Apache Web server[16] with the *mod_rewrite* and *mod_redirect* software.

4 Collection Statistics

We now present some statistical results from an initial crawl of a set of selected publications identified by our librarians. The selected publications contents varied in topic (general and specific) as well as in periodicity (daily, weekly, monthly, annual) and distribution (national and regional).

We limited our *collector* to retrieve only documents residing on the base server. The base server is the server that contains the entry point document for the publication. We also restricted our scan to collect URLs within a maximum depth of six from each entry point of a publication.

The *collector* made 60523 HTTP requests which retrieved successfully 72% of all documents of the list of publications (see Table 1). About 19% of the documents not collected resided on servers which were rejected because we chose to harvest only those documents available from the base server. We were surprised to find that less than 5% of all our HTTP requests were returned as non successful HTTP codes. HTTP response and *collector* error codes generated during the crawl of the Portuguese periodic publications collection are displayed in Table 1. Results from generic crawlers usually indicate higher percentages of broken links.

Table 1. HTTP response and *collector* error codes generated during the crawl of the Portuguese periodic publications collection

Code	Meaning	Number	Percent
200	OK	43322	72%
-5	Invalid server	11468	19%
-11	Exception	2743	5%
404	Not Found	2381	4%
-10	Exceeded document retrieval interval	103	0%
-2	Unnatural error	181	0%
-4	Exceeded size	129	0%

Another interesting statistic is that only one publication excluded our crawler from harvesting it by the Robots Exclusion Protocol(REP)[14].

We also noticed that more than 83% of all documents varied in size in between 2KB and 32 KB (Figure 5).

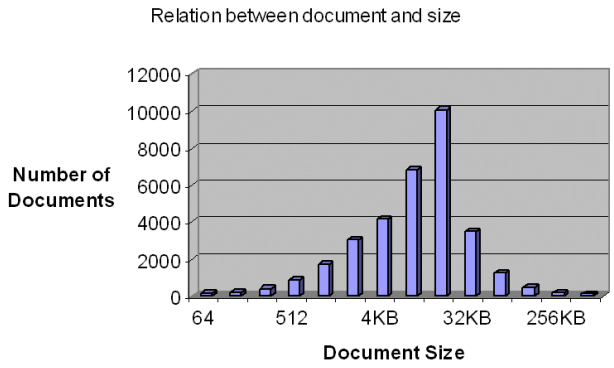

Fig. 5. Relation between documents retrieved and their size

Around 90% of all documents collected were either HTML documents, GIF images or JPEG images (see Table 2). Even though our *collector* was unable to determine the type of 9% of all documents, a more thorough study of their file extensions imply that most of them are HTML documents, GIF images and JPEG images.

This means that 99% of all documents retrieved are small and easy to interpret. This shows a crawl restricted to those publications of historic interest could easily retrieve almost all their contents and that it can be easily reviewed with widely available tools.

Table 2. Distribution of MIME types

MIME type	Number	Percent
text/html	34424	70%
image/gif	5448	11%
unknown	4478	9%
image/jpeg	4400	9%
text/plain	238	0%
application/zip	64	0%
application/msword	57	0%
application/pdf	45	0%
	49256	100%

The fact that the web publications that we harvested are much more easily handled (they only use the most widely adopted formats, have very few broken links, etc) may suggest that publishers in general strive to make their publications as easily accessible as possible to maximize their readership. This in turn makes our goal in harvesting for preservation easier to achieve. We believe that this trend will increase as the Internet matures as a publishing media.

5 Conclusions and Future Work

We have developed an information system for the deposit of digital items. Our approach separates the mechanisms for harvesting and accessing collected items from the policies that estabilish how they are performed. Hence, operators need to configure DROP and PURL.PT to behave as intended. However, policies are hard to estabilish, as we still face several open issues, such as respecting copyright policies and handling the merging and splitting of publications. Initial usage of the first prototype shows that it can handle those Internet publications of interest to our end-users.

We still face the problem of later being able to access harvested items. Many publications today use languages and programs such as JavaScript, Java Applets or ShockWave Flash to provide dynamic contents; our collector does not analyze these objects and may loose the harvesting of some of the URLs of a publication. As a result, collected publications may not be properly harvested or may be hard to interpret.

These difficulties are just the unraveling of a bigger issue: it is very difficult to build a system made for preservation in a Web world with proprietary, incomplete, undefined, unclear, non followed, complex standards. Future work in this direction may provide the collector with a parser for interpreting scripts and collect all the referred documents. Additional future work in the harvesting process is also necessary, to automate ways of detecting faults and recovering from them. The study of techniques to optimize harvesting is currently a major work area [2].

References

1. Jose Borbinha. A URN namespace for resources maintained by the National Library of Portugal – Internet Draft (submission in progress).
2. Junghoo Cho and Hector Garcia-Molina. The evolution of the web and implications for an incremental crawler. In Amr El Abbadi, Michael L. Brodie, Sharma Chakravarthy, Umeshwar Dayal, Nabil Kamel, Gunter Schlageter, and Kyu-Young Whang, editors, *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt*, pages 200–209, 2000.
3. Working Group of the Conference of Directors of National Libraries. The legal deposit of electronic publications. Available at `http://www.unesco.org/webworld/memory/legaldep.htm`, December 1996.
4. Library of Congress Must Improve Handling Of Digital Information. LC21: A Digital Strategy for the Library of Congress. Available at `http://www4.nationalacademies.org/news.nsf/isbn/0309071445?OpenDocument` Accessed on June 2001.
5. Networked European Deposit Library Available at `http://www.kb.nl/nedlib/`. Accessed on June 2001.
6. Long-term Preservation of Electronic Publications: The NEDLIB project Available at `http://www.dlib.org/dlib/september99/vanderwerf/09vanderwerf.html`. Accessed on June 2001.
7. Naming and Addressing: URIs, URLs, ... Web Naming and Addressing Overview. Available at `http://www.w3.org/Addressing/`. Accessed on June 2001.
8. Universal Resource identifiers in WWW Available at `http://www.w3.org/Addressing/URL/uri-spec.html`. Accessed on June 2001.
9. The PANDORA Project: a summary of progress PANDORA Archive - Key Documents Available at `http://pandora.nla.gov.au/documents.html`. Accessed on June 2001.
10. R. Moats. RFC 2141: URN syntax, 1997.
11. National bibliographic database - Porbase. Available at `http://portico.bl.uk/gabriel/en/countries/portugal-union-en.html`, Porbase available at `http://porbase.bn.pt/`.
12. Andrew Waugh, Ross Wilkinson, Brendan Hills, and Jon Dell'Oro. Preserving digital information forever. In *Proceedings of the Fifth ACM Conference on Digital Libraries, June 2-7, 2000, San Antonio, TX, USA*, pages 175–184. ACM, 2000.
13. PostgreSQL. PostgreSQL - a sophisticated Object-Relational DBMS. Available at `http://www.postgresql.org`,
14. Martijn Koster. A Standard for Robot Exclusion. Available at `http://info.webcrawler.com/mak/projects/robots/norobots.html`, The Robots pages at WebCrawler available at `http://info.webcrawler.com/mak/projects/robots/robots.html`.
15. OCLC PURL Service. Persistent URL at `http://purl.oclc.org/`
16. The Apache Software Foundation. Available at `http://www.apache.org`
17. Brewster Kahle Archiving the Internet Scientific American, March 1997.