Preserving the Pyramid of STI Using Buckets

Michael L. Nelson and Kurt Maly

ABSTRACT

The product of research projects is information. Through the life cycle of a project, information comes from many sources and takes many forms. Traditionally, this body of information is summarized in a formal publication, typically a journal article. While formal publications enjoy the benefits of peer review and technical editing, they are also often compromises in media format and length. As such, we consider a formal publication to represent an abstract to a larger body of work: a pyramid of scientific and technical information (STI). While this abstract may be sufficient for some applications, an in-depth use or analysis is likely to require the supporting layers from the pyramid.

We have developed buckets to preserve this pyramid of STI. Buckets provide an archive- and protocol-independent container construct in which all related information objects can be logically grouped together, archived, and manipulated as a single object. Furthermore, buckets are active archival objects and can communicate with each other, people, or arbitrary network services. Buckets are an implementation of the Smart Object, Dumb Archive (SODA) DL model. In SODA, data objects are more important than the archives that hold them. Much of the functionality traditionally associated with archives is pushed down into the objects, such as enforcing terms and conditions, negotiating display, and content maintenance. In this paper, we discuss the motivation, design, and implication of bucket use in DLs with respect to grey literature.

INTRODUCTION

Research projects produce information in a variety of formats. A traditional formal publication, such as a journal article, is generally supported by a large quantity of software, datasets, images, video, informal notes, presentations and other documents. Collectively, we call this set of scientific and technical information (STI) the "Pyramid of STI".

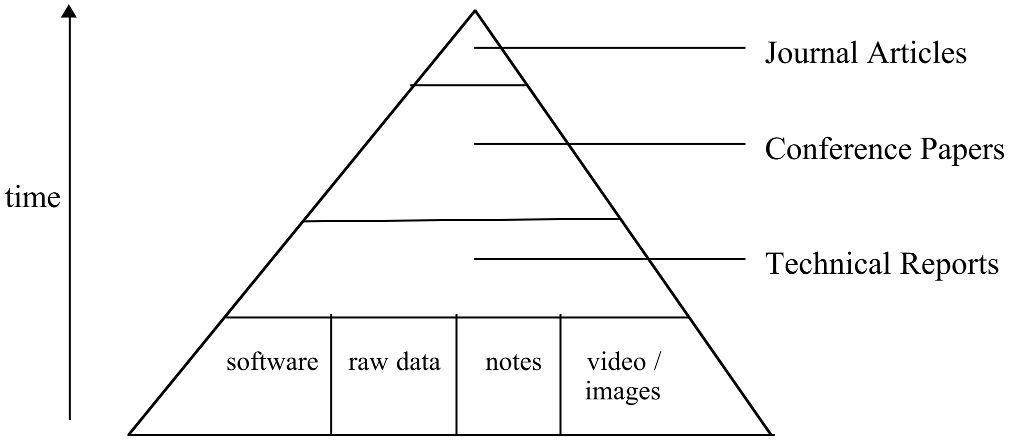

Figure 1: Formal Publications Rest on a Pyramid of STI

Although the information in this pyramid was created together and subtle relationships between its components can exist, different semantic instantiations are generally segregated along currently obsolete media boundaries. Reports are placed in report archives, software might go into a software archive, but most of the data, supporting materials and other grey information are likely to be kept in informal personal archives or discarded altogether. Our experience with NASA digital libraries (DLs) is users wish to have access to the supporting materials, data and software used in the preparation of formal literature (Sobieszczanski-Sobieski, 1994) -- even though there currently is no well established publication vector for much of this information.

We feel many DL projects focus on simply automating the formal publication process, while providing little attention to the lower tiers of the pyramid of STI. Similarly, we feel that creating “separate but equal” DLs implemented congruently with obsolete media boundaries pushes the burden of STI (re)integration to the user.

To address these concerns, we have created buckets: aggregative, intelligent agents tailored for publishing in digital libraries. Buckets can be thought of DL-specific Digital Objects as described in (Kahn and Wilensky, 1995). Buckets provide a container mechanism to capture and preserve the pyramid of STI. To enhance their long-term survivability, buckets are completely self-contained. All the logic to manage, and protect their content is embedded in the bucket itself, not in a separate server. Buckets can also be thought of as “archivelets”. Communication with the bucket occurs through messages utilizing the bucket application programming interface (API). By promoting buckets to first class network citizens, we are stressing that content is more important than the search engine or DL protocol used to access it. By imbuing the content with additional functionality, we increase the long term usefulness and survivability of the STI.

Grey Literature at NASA

NASA communicates its research findings through the traditional open literature process as well as its own multi-tiered, self-published report series (Pinelli, 1990). The NASA report series offers a number of advantages to authors: no page restrictions, potential for restricting dissemination, possibility of color graphics, and occasionally the inclusion of a CD-ROM of data, images or software. However, the latter two are rarer than most authors would like because they are expensive to create, and their distribution is more expensive still. The NASA reports are often ingested in systems that can handle only paper hard copy or possibly just microfiche -- leaving few options for propagation of additional media formats such as CD-ROMs.

An even more compelling case for capturing grey literature at NASA is that the formal publications (NASA's report series or open literature) represent a decreasing percentage of the total amount of STI created and used by NASA and its customers. Due to the increasingly proprietary nature of NASA's work, as well as increasing time constraints on fewer staff members, many research projects are no longer resulting in a formal publication. Instead, the projects remain as a collection of briefings, data, and other forms of grey literature -- often with proprietary access restrictions. While neglecting the formal publications achieves the short term goal of increased project turn around time, the inability to capture and preserve any of the resultant STI creates a gap in the corporate memory. There is no well-defined, large scale publishing outlet for a majority of the STI created at NASA.

Shortcomings of Current Digital Libraries

We recently surveyed a number of digital libraries from multiple disciplines and found that most focused only on formal publications (Esler & Nelson, 1998). Even DLs that focus on grey literature (i.e. technical reports, pre-prints, working papers) still only focus on the hard-copy representation. Software, images, video and other material, if they are preserved at all, are often sent to DLs that serve only those respective media formats. Such examples include NIX, an image DL for NASA (von Ofenheim, et al., 1998); Netlib, a software DL for the high performance computing community (Browne, et al., 1995); and Alexandria, a DL for geospatial data (Smith, 1996). While these and other like DLs can provide custom interfaces for interacting with non-textual STI, they have as a side effect an artificial segregation along media formats.

We are aware of no other current DL that allows for archiving and serving a complex set of STI. Consider a research project that produced raw data, software to reduce the data, reduced data, shift notes, image or video representation of data, presentations, informal reports, and a journal article that summarized the project's findings. Currently, the journal article has the best chance of appearing in a DL. Possibly the software will make it to a software DL, and depending on the discipline the data might make it to a data archive, but neither is guaranteed. The "base" of the pyramid of STI is at worst, effectively thrown away, and at best is splintered and sent to separate DLs. If researchers read the journal article and wish to extend the data analysis differently than presented, they can either search in other DLs for the software and datasets, or they can contact the authors in the hopes of obtaining the software and datasets through collegial distribution. The former assumes the various non-report STI has been placed in DLs somewhere, while the latter assumes the authors can still be reached.

BUCKETS

Buckets are object-oriented container constructs in which logically grouped items can be collected, stored, and transported as a single unit. Buckets are completely self-contained: they have their content physically resident, the intelligence to serve, manage and enforce terms and conditions for accessing the content. Their self-contained nature allows them to function independently of, and in cooperation with, any DL system or protocol. It also allows them to be completely mobile - a bucket can "move around" and still retain its functionality. Buckets also maintain their own logs of actions performed on them, so they can retain a history that is independent of the servers used to access them.

As far as what goes into a bucket, buckets provide mechanism -- not policy. There are no pre-defined concepts of what information types should go into a bucket, and buckets make no assumptions about their content. The authors and publishing organizations control what constitutes a bucket; if there is reason why a lower strata in the pyramid should not be preserved, there is no requirement that it be included. Similarly, buckets are not unsuitable for white literature; they simply offer the capability (should the authors and publishers choose) to extend beyond the white into the grey.

To the casual observer, a bucket appears as a regular web page. However, messages are sent to the bucket using the hypertext transfer protocol (http). If no message is sent, by default the bucket builds an HTML presentation page of its contents:

```
http://dlib.cs.odu.edu/test-bucket3/
```

However, other messages are possible. If a web robot wanted to gather the structured metadata, it could issue this bucket message:

http://dlib.cs.odu.edu/test-bucket3/?method=metadata

To learn what methods this bucket supports, it could issue:

http://dlib.cs.odu.edu/test-bucket3/?method=list_methods

Many other methods are defined, including methods listing the bucket source code, updating the bucket source code, changing who are privileged principals to the bucket, and other various functions. See (Nelson, et al., 1999) for a further discussion of bucket methods and implementation.

SODA: SMART OBJECTS, DUMB ARCHIVES

An observation from our experiences with current NASA DLs is that a surprising number of people do not find the publications via the respective DLs. Since the full contents of the NASA DLs are browsable, both the abstract lists and the reports are indexed by web crawlers, spiders and the like. Users are formulating complex queries to services such as Yahoo, Altavista, Lycos, Infoseek, etc. to find NASA STI. We presume this is indicative of the resource discovery problem: people start at these portals because they do not know all the various DLs themselves; and the meta-searching problem: they are trusting these services to search many sources, not just the holdings of a single DL.

Although we believe we have built attractive and useful interfaces for the NASA DLs, our main concern is that people have access to NASA’s holdings and not that they use a given DL interface. It is desirable that NASA publications are indexed by many services. Since there can be any number of paths to the information object, the information object must be a first class network citizen, handling presentation, terms and conditions, and not depending on archive functionality.

In the SODA model, we have separated DLs into three separate layers: 1) the Digital Library Services (DLS) layer that provides user interfaces such as searching and browsing; 2) the archive layer that manages collections of objects; and 3) the objects themselves. Separating the functionality of the archive from that of the DLS allows for greater interoperability and federation of DLs. The archive's purpose is to provide DLs the location of buckets (the DLs can poll the buckets themselves for their metadata), and the DLs build their own indexes. And if a bucket does not “want” to share its metadata (or contents) with certain DLs or users, its terms and conditions will prevent this from occurring. For example, we expect the NASA digital publishing model to begin with technical publications, after passing through their respective internal quality control, to be placed in a NASA archive. The NASA DL (which is the set of the NASA buckets, the NASA archive(s), the NASA DLS, and the user communities at each level) would poll this archive to learn the location of buckets published within the last week. The NASA DL could then contact those buckets, requesting their metadata. Other DLs could index NASA holdings in a similar way: polling the NASA archive and contacting the appropriate buckets. The buckets would still be stored at NASA, but they could be indexed by any number of DLs, each with the possibility for novel and unique methods for searching or browsing. Or perhaps the DL collects all the metadata, then performs additional filtering to determine applicability for inclusion into their DL. In addition to an archive's holdings being represented in many DLs, a DL could contain the holdings of many archives. If we view all digitally available publications as a universal corpus, then this corpus could be represented in N archives and M DLs, with each DL customized in function and holdings to the needs of its user base. The SODA model for DLs is discussed in detail in (Maly, et al., 1999a).

FUTURE WORK

There are several areas we continue to explore and develop, the first of which is tools for buckets. The long-term success of buckets will depend on the quality of the tools to create and manage buckets. We have Publishing Tool, that allows users to transparently create and populate buckets. We have a Management Tool, which is a simple workflow mechanism to review buckets submitted for publication, approve or reject them, and move them into designated areas when approved. Finally, we have an Administration Tool that allows for long term maintenance of buckets, performing such operations as large scale source code and principal updates. These tools are undergoing frequent revisions as we gain user feedback.

Secondly, we are developing the concept of discipline specific buckets that can exploit knowledge about their contents. These bucket templates could have specialized display capabilities, specialized methods for data interaction, custom storage options, etc. These would be in contrast to the ontology neutral buckets presented here. The first of these, a bucket template for undergraduate education, is discussed in (Maly, et al., 1999b).

The third area we are actively developing is making buckets intelligent agents. The buckets are already carrying the intelligence to manage their content -- why not make them even more intelligent? We are designing a Bucket Communication Space (BCS) that will allow buckets to communicate with other buckets, people, and arbitrary third party network resources. The BCS will allow for functions such as messaging, format conversion, and matching. The latter is especially interesting: we will allow buckets to find similar buckets off-line. Buckets will register their profiles with the BCS, and the BCS will inform buckets of their potential matches based on a similarity index. Buckets can then either automatically establish linkages, or contact their owners for verification.

Additional functions afforded by intelligent buckets includes a redefinition of software reuse. If general software resides in a bucket in a DL, users can either download and integrate it with their applications, or use the software while it is resident in DL. Many DL projects have agent technology designed to assist the user in searching, browsing or other functionality. However, we are unaware of other projects that attempts to make the archived object itself intelligent. The full implications of making archival objects intelligent agents has yet to be explored.

CONCLUSIONS

Past media format limitations have defined our view of an archival unit of STI. We claim the formal publications that are the focus of traditional libraries and many DL projects are simply an abstract to a larger body of information. This pyramid of STI is supported by informal documents, software, data, images, videos and other multi-format material. Our DL experiences have lead us to believe that there is high user interest in obtaining access to the lower tiers of the pyramid of STI -- many of which are not archived at all.

To this end, we have developed buckets: object-oriented, intelligent agents for publishing in DLs. Buckets are DL system and protocol neutral; they depend on no particular system and are designed to have minimal impact on any such system. Messages are communicated to buckets via an API that uses http as a transport. To increase their long term survivability, buckets are completely self-contained, carrying all of their contents internally as well as the logic to manage, serve and protect those contents. In addition to the aggregative properties of buckets, we are also working to make them intelligent agents. We are implementing the Bucket Communication Space to allow buckets to communicate with each other and perform tasks such as finding and linking to similar buckets. By imbuing STI objects with aggregation and intelligence, we can now preserve the entire pyramid of STI, as well as introduce new functional capabilities for archived objects.

REFERENCES

Browne, S.; Dongarra, J.; Grosse, E. and Rowan, T. (1995), "The Netlib Mathematical Software Repository", D-Lib Magazine, September 1995. Available:
http://www.dlib.org/dlib/september95/netlib/09browne.html

Esler, S. L. and Nelson, M. L. (1998), "Evolution of Scientific and Technical Information Distribution", *Journal of the American Society of Information Science*, Vol 49 No 1, pp. 82-91.

Kahn, R. and Wilensky, R. (1995), "A Framework for Distributed Digital Object Services", cnri.dlib/tn95-01. Available:
http://www.cnri.reston.va.us/cstr/arch/k-w.html

Maly, K.; Nelson, M. L. and Zubair, M. (1999), "Smart Objects and Dumb Archives: A User-Centric, Layered Digital Library Framework", D-Lib Magazine, Vol 5 No 3. Available: http://www.dlib.org/dlib/march99/maly/03maly.html

Maly, K.; Zubair, M.; Liu, X.; Nelson, M. L., Zeil, S. J. (1999), "Structured Course Objects in a Digital Library", to appear in *Proceedings of the International Symposium on Digital Libraries*, Tsuka Japan.

Nelson, M. L.; Maly, K.; Zubair, M. and Shen, S. N. T. (1999), "Buckets: Aggregative, Intelligent Agents for Publishing", *Webnet Journal*, Vol 1 No 1, pp. 58-66.

Pinelli, T. E. (1990), "Introduction to National Aeronautics and Space Adminsrtration's Scientific and Techincal Information Program", *Government Information Quarterly*, Vol 7 No 2, pp. 123-126.

Smith, T. (1996), "A Brief Update on the Alexandria Digital Library Project", *D-Lib Magazine*, March 1996. Available:
http://www.dlib.org/dlib/march96/briefings/smith/03smith.html

Sobieszczanski-Sobieski, J. (1994), "How to Improve NASA-Developed Computer Programs", NASA CP-10159, pp. 58-61.

von Ofenheim, W. H. C.; Heimrel, N. L.; Binkley, R. L.; Curry, M. A.; Slater, R. T.; Nolan, G. J.; Griswold, T. B.; Kovach, R. D.; Corbin, B. H. and Hewitt, R. W. (1998), "NASA Image eXchange (NIX)", NASA/TM-1998-206295.

Michael L. Nelson
NASA Langley Research Center
MS 158
Hampton, VA USA 23681
+1 757 864 8511
+1 757 864 8342 (fax)
m.l.nelson@larc.nasa.gov
http://home.larc.nasa.gov/~mln/

Michael is an electronics engineer at NASA Langley Research Center and a Ph.D. candidate in computer science at Old Dominion University.

Kurt Maly
Department of Computer Science
Old Dominion University
Norfolk, VA USA 23592
+1 757 683 4817
+1 757 683 4900 (fax)
maly@cs.odu.edu
http://www.cs.odu.edu/~maly/

Kurt is the Kaufman Professor and Chair of the Computer Science Department of Old Dominion University.