

Unsupervised Creation of Small World Networks for the Preservation of Digital Objects

Charles L. Cartledge
Old Dominion University
Department of Computer Science
Norfolk, VA 23529 USA
ccartled@cs.odu.edu

Michael L. Nelson
Old Dominion University
Department of Computer Science
Norfolk, VA 23529 USA
mln@cs.odu.edu

ABSTRACT

The prevailing model for digital preservation is that archives should be similar to a “fortress”: a large, protective infrastructure built to defend a relatively small collection of data from attack by external forces. Such projects are a luxury, suitable only for limited collections of known importance and requiring significant institutional commitment for sustainability. In previous research, we have shown the web infrastructure (i.e., search engine caches, web archives) refreshes and migrates web content in bulk as side-effects of their user-services, and these results can be mined as a useful, but passive preservation service. Our current research involves a number of questions resulting from removing the implicit assumption that web-based data objects must passively await curatorial services: What if data objects were not tethered to repositories? What are the implications if the content were actively seeking out and injecting itself into the web infrastructure (i.e., search engine caches, web archives)? All of this leads to our primary research question: *Can we create objects that preserve themselves more effectively than repositories or web infrastructure can?*

Categories and Subject Descriptors

H.3.7 [Digital Libraries]:

General Terms

Algorithms, Design, Experimentation

1. INTRODUCTION

In previous research, we have investigated alternative models for preservation, including how the web infrastructure (search engine caches, web archives, and other projects that refresh and migrate web resources) can be used to reconstruct “lost” web sites [20] as well as discover new and similar web pages [15], and how web servers could play a role

in preparing “preservation-ready” resources [30]. In the attempt to explore additional models, we are currently researching the technologies necessary to create web-based digital objects that can be imbued with the capability to outlast their individual or organizational stewards. While conventional repository and preservation techniques will continue to be appropriate for collections of objects of known value, we are interested in techniques that can be applied for the large mass of web content that may become valuable in the future, or even content that a third party can deem valuable and worthy of preservation.

This work builds on our earlier research of “buckets” [22], which are based on the digital objects (DOs) first introduced in the Kahn Wilensky Framework [14], but with the internal capability to perform functions normally reserved for repositories. While the original implementations were based on server-side web programs, we envision future implementations taking advantage of client-side technologies such as JavaScript or Adobe Flash. In addition, the OAI-ORE project [18] has introduced *Resource Maps* which are suitable for defining the boundaries of DOs. The combination of client-side scripting and Resource Maps has been explored we will build on some of the concepts introduced in client-side preservation research [21].

Our interest in self-preserving DOs is spurred by the emergence of these technologies as well as the proliferation of free storage in the web infrastructure. This paper introduces the motivation for and scenarios describing how self-preserving DOs could be deployed as well as some preliminary discussions about how the DOs can arrange themselves, without archivist intervention, into small-world graphs in the web.

2. RELATED WORK

2.1 Repositories

There are a number of repositories that range from theoretical to ready-to-download. Some are frameworks or architectural proposals such as aDORE [32] and SAV [6]. Some, like Fedora [25], are middleware systems, ready to be the core repository technology in a local deployment. Some are complete systems, ready to deploy. These include DSpace [31], sponsored by MIT and HP Laboratories and LOCKSS [19], sponsored by the Stanford University Libraries. All are widely implemented and enjoy a large user community. DSpace is an institutional repository, intended to archive the intellectual output of a university’s faculty and students. LOCKSS allows libraries to create “dark archives” of publish-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

JCDL’09, June 15–19, 2009, Austin, Texas, USA.

Copyright 2009 ACM 978-1-60558-322-8/09/06 ...\$5.00.

ers' websites. As long as the publishers' websites are available, all web traffic goes to those sites. But if the publishers' contents are lost, the dark archives are activated and the content is available again. Risk is mitigated through many sites archiving content of their own choosing. Depending on an institution's requirements, the systems described above can be quite attractive. But there is an implicit assumption on any repository system: that there is a person, community or institution that exists to tend to the repository. What happens when the responsible organization no longer exists? There are repository trading and synchronization provisions (e.g., [7]), but most are specific to a particular repository architecture.

Moving away from institution-centric repositories, several generic network storage systems and APIs have also been proposed, including Cooperative File Systems (CFS) [8], Internet Backplane Protocol (IBP) [3], Storage Resource Broker (SRB) [26] and OceanStore [29]. Systems such as CFS and OceanStore rely on distributed hash tables and an overlay network to locate content in the Internet. Such additional levels of shared infrastructure have not been widely deployed, and are not in use in any institutional repository. IBP and SRB are more traditional in their repository design and have enjoyed greater deployment. SRB (and its follow-on, iRODS) in particular has a user community similar in size to LOCKSS and Fedora.

2.2 Graph construction

Our approach for the construction of networks of DOs contrasts with others [24] where connections to other nodes are proportional to the destination's degree count. We can use preferential attachment only to select the first node when starting to consider connecting to any nodes as compared with algorithms that starts with a graph (or lattice) and then grows a "small world" by the addition of new links [10, 12, 9, 16]. Or, by connecting a node to a fixed number of vertices based on their degree [4], or even creating a small world graph from a random one [11], whereas we construct a small world one from the beginning. Some approaches grow a graph based on preferential attachment (or "fitness") [17, 2]. A survey of small world graph construction techniques and analysis is given [23], but none discuss the creation of small world graphs based on locally gleaned knowledge in a manner that we demonstrate.

3. SELF-PRESERVING DIGITAL OBJECTS

3.1 Flocking for Preservation

We are motivated by Reynolds's seminal paper on "boids" [28], in which he demonstrated that three simple rules were sufficient to simulate the complex behaviors of schools of fish, flocks of birds, herds of animals and the like. The rules themselves are simple, but the behaviors that emerge from the rules are complex and realistic. The salient feature is that these rules are scale-free: only the neighbors are accounted for in the computation; knowing the entire size of the network is not required. We believe these simple rules can be adapted to create self-preserving digital objects with similar complex emergent behaviors. Table 1 lists the rules that Reynolds proposed for boids (his term for bird-like objects) and our interpretation for DOs.

In the same way that flocks self-navigate to new locations that have the resources they need, we envision collections of archival data objects self-preserving, using a loose confederation of cooperating archives with varying levels of resources and availability. Making copies of oneself in new repositories is performed in an opportunistic model, within the guidelines imbued in the DOs at creation time. Also, the entire collection (or parts of it) may be steered from time to time by an archivist, but for the most part the data objects replicate themselves.

Collision avoidance is perhaps the easiest rule to visualize the transcription from boids to DOs. DOs flocking to a new repository cannot overwrite each other (collide in physical storage), nor collide in namespaces (have the same URI). This is orthogonal to the naming mechanism used: URN implementations such as handles or DOIs, globally unique identifiers (GUIDS) or content addressable naming schemes [27].

With boids, the concept of *velocity matching* is to travel the same speed as your neighbors. This is perhaps the most difficult rule transformation. However, interpreting velocity as resource consumption (i.e., storage space) makes this rule more intuitive. Specifically, a DO should try to consume as much, and only as much, storage as everyone else. In resource-rich environments, making as many copies of yourself as you would like is easy. When storage becomes scarce, this becomes more difficult. So there must be a provision for DOs to delete copies of themselves from different archives to make room for late arriving DOs in low-storage situations. DOs will never delete the last copy of themselves to make room for new DOs, but they will delete copies of themselves to come down from a soft threshold (e.g., 10 copies) down to a hard threshold (e.g., 3). When resources become plentiful again, new copies can be made.

For boids, *flock centering* means staying near (but not colliding with) other flockmates. We interpret this similarly, with DOs attempting to stay near other DOs as they make copies of themselves at new repositories. In essence, when a DOs learns of a new repository and makes a copy of itself there, it should tell the other DOs it knows so they will have the opportunity to make copies of themselves at the new location if they wish. Announcing the location of a new repository will thus cause DOs at other repositories that have not reached their upper limit on creating copies to flow to the new repository.

3.2 Unsupervised Small-World Graph Creation

We introduce some terminology to discuss how DOs can self-arrange. *Friends* are DOs are within a graph. The nodes are directly connected to particular DO are considered its friends. A *Family* is all DOs that are replicas of each other. A *Parent* is the family member that was first inserted into the graph and is responsible for ensuring that enough family members are created to meet its preservation goals.

Based on a review of graph structures, their characteristics and attributes, small world graphs appear to be the most practical choice for minimizing a graph's size, communication costs and construction effort. Small world graphs

Rules	Flocking Boids	Flocking DOs
Collision Avoidance	avoid collisions with nearby flockmates	not overwriting one's own copies nor the copies of other DOs (i.e., namespace collision avoidance)
Velocity Matching	attempt to match velocity with nearby flockmates	deleting copies of oneself to provide space for late arrivals in a storage location
Flock Centering	attempt to stay close to nearby flockmates	following others to available storage locations

Table 1: Original Flocking Rules and DOs Flocking Rules.

also emulate natural processes and occur often in nature and human endeavors, where regular and random graphs are relatively infrequent.

A “wandering” node is introduced to an existing node in the web. The wanderer gets a list of friends from the node it is introduced to. If the wandering node does not form a friendship link with the initial node, it will select another candidate node that it has learned of from all its conversations with other nodes it has encountered. When the wandering node finally makes a friendship link, it will then look back at all the nodes that it did not connect with as well as some that it was intending to communicate with and make friendship links with some of them.

Friendship links are separate from HTML navigation links (i.e., `link` instead of `a HTML elements`). These links serve as a way for DOs to send messages from one to another, such as when new storage locations are available or message concerning the scope and migration of file formats (cf. the semi-automated alert system described in Panic [13]). Friendship links are used to support the replication process. If a DO needs to replicate itself and it has a friend that lives on a different host and if there is room on that host for an additional DO then the DO can replicate itself onto the new host. Replication is how a family grows from a single copy (a parent) to multiple copies (a family). The friendship links, family links and how family members are spread across various hosts is shown in Figure 1.

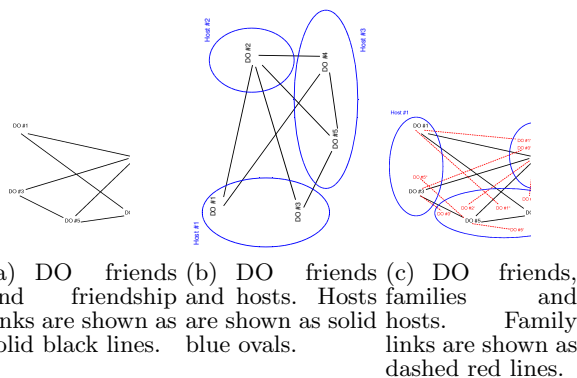


Figure 1: A layering of DO friends, hosts and families. A DO has many friends and lives on a host. Each family member lives on a different host.

4. FUTURE WORK AND CONCLUSIONS

Our investigations to date have focused on developing the concept of constructing a small-world graph for preservation

purposes based on locally available data. One of the next areas that we will focus on is the error and attack tolerance (cf. [1]) of the USW. An error is an unintended failure of a network component, where an attack is an intended failure:

Loss of a parental DO: A parental DO has special responsibilities with regard to the creating and distributing preservation copies and so is important to the well being of the family. When a parental DO is lost, a replacement must be selected from all remaining family members. There are a number of distributed election algorithms, including bullying and ring could be used to elect a new parent that would assume all parental duties.

Loss of a host: This would probably result in the loss of several nodes and preservation copies from the graph. For those families that lost a parental DO, the general approach from the previous bullet would apply. Loss of preservation copies would be resolved during routine family maintenance.

Disconnection of the graph: This is a major event and could be the result of the failure of a underlying hardware infrastructure or a concerted attack and can be characterized as the loss of many hosts simultaneously. Parental election would occur to replace lost parental DOs, concurrently with the creation of replacement preservation copies to bring the remaining families up to their directed levels.

Reconnection of the graph: Reconnection could bring multiple copies of the same family (one in each of the previously disconnected graphs) back into contact. As these families make contact, they could engage in an election process to select the truest parental node. The newly elected parent would then set about managing its preservation copies.

We have run extensive simulations to determine the parameters needed for DOs to create a “small world” graph based on purely locally derived information and without any global knowledge, guidance or direction using our unsupervised small world (USW) algorithm in a simulation environment (the algorithm and results are detailed in [5]). Next, we will implement the USW using client-side approaches such as JavaScript or Adobe Flash, combined with ORE Resource Maps to define the boundaries of the DO.

5. ACKNOWLEDGEMENTS

This work supported in part by the NSF, Project 370161.

6. REFERENCES

- [1] R. Albert, H. Jeong, and A.-L. Barabasi. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, July 2000.
- [2] A.-L. Barabási, R. Albert, and H. Jeong. Scale-free characteristics of random networks: The topology of the world wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1):69–77, June 2000.

- [3] M. Beck, T. Moore, and J. S. Plank. An end-to-end approach to globally scalable network storage. In *SIGCOMM '02: Proceedings of the 2002 Conference on applications, technologies, architectures, and protocols for computer communications*, pages 339–346, 2002.
- [4] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. The Degree Sequence of a Scale-Free Random Graph Process. *Random Structures and Algorithms*, 18(3):279–290, 2001.
- [5] C. L. Cartledge and M. L. Nelson. Unsupervised creation of small-world graphs of digital objects. In *Submitted for Publication*, 2009.
- [6] B. Cooper, A. Crespo, and H. Garcia-Molina. Implementing a reliable digital object archive. In *ECDL '00: Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries*, pages 128–143, 2000.
- [7] B. F. Cooper and H. Garcia-Molina. Peer-to-peer data trading to preserve information. *ACM Transactions on Information Systems (TOIS)*, 20(2):133–170, 2002.
- [8] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative storage with CFS. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP '01)*, October 2001.
- [9] P. Duchon, N. Hanusse, E. Lebhar, and N. Schabanel. Could any graph be turned into a small-world? *Theoretical Computer Science*, 355(1):96–103, 2006.
- [10] P. Duchon, N. Hanusse, E. Lebhar, and N. Schabanel. Towards small world emergence. In *SPAA '06: Proceedings of the eighteenth annual ACM symposium on Parallelism in algorithms and architectures*, pages 225–232, New York, NY, USA, 2006. ACM.
- [11] B. Gaume and F. Mathieu. From random graph to small world by wandering. Technical Report 6489, Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France), 2008.
- [12] K. I. Goh, B. Kahng, and D. Kim. Universal behavior of load distribution in scale-free networks. *Physical Review Letters*, 87(27):278701+, December 2001.
- [13] J. Hunter and S. Choudhury. A semi-automated digital preservation system based on semantic web services. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 269–278, 2004.
- [14] R. Kahn and R. Wilensky. A framework for distributed digital object services. *International Journal on Digital Libraries*, 6(2):115–123, 2006.
- [15] M. Klein and M. L. Nelson. Revisiting lexical signatures to (re-)discover web pages. In *ECDL '08: Proceedings of the 12th European Conference on Research and Advanced Technology for Digital Libraries*, pages 371 – 382, 2008.
- [16] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, 2000.
- [17] K. Klemm and M. Eguíluz, Víctor. Growing scale-free networks with small-world behavior. *Physical Review E*, 65(5):57102, 2002.
- [18] C. Lagoze, H. Van de Sompel, P. Johnston, M. L. Nelson, R. Sanderson, and S. Warner. Object Re-Use & Exchange: A Resource-Centric Approach. Technical Report arXiv:0804.2273, 2008.
- [19] P. Maniatis, M. Roussopoulos, T. J. Giuli, D. S. H. Rosenthal, and M. Baker. The LOCKSS peer-to-peer digital preservation system. *ACM Transactions on Computer Systems*, 23(1):2–50, 2005.
- [20] F. McCown, N. Diawara, and M. L. Nelson. Factors affecting website reconstruction from the web infrastructure. In *JCDL '07: Proceedings of the 2007 conference on Digital libraries*, pages 39–48, 2007.
- [21] F. McCown, M. L. Nelson, and H. Van de Sompel. Everyone is a curator: Human-assisted preservation for ore aggregations. In *Proceedings of the DigCCurr 2009*, 2009.
- [22] M. L. Nelson and K. Maly. Buckets: smart objects for digital libraries. *Communications of the ACM*, 44(5):60–62, 2001.
- [23] M. E. J. Newman. Models of the small world: A review. *J.STAT.PHYS.*, 101:819, 2000.
- [24] V. Nguyen and C. Martel. Analyzing and characterizing small-world graphs. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 311–320, 2005.
- [25] S. Payette and T. Staples. The Mellon Fedora project. In *ECDL '02: Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries*, pages 406–421, 2002.
- [26] A. Rajasekar, M. Wan, and R. Moore. MySRB & SRB: Components of a data grid. In *HPDC '02: Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing HPDC-11 20002 (HPDC'02)*, pages 301–310, 2002.
- [27] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, 2001.
- [28] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, 1987.
- [29] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz. Maintenance-free global data storage. *IEEE Internet Computing*, 5(5):40–49, 2001.
- [30] J. A. Smith and M. L. Nelson. A quantitative evaluation of dissemination-time preservation. In *ECDL '08: Proceedings of the 12th European Conference on Research and Advanced Technology for Digital Libraries*, pages 346 – 357, 2008.
- [31] M. Smith. DSpace: An institutional repository from the MIT libraries and Hewlett Packard Laboratories. In *ECDL '02: Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries*, pages 543–549, 2002.
- [32] H. Van de Sompel, J. Bekaert, X. Liu, L. Balakireva, and T. Schwander. aDORe: A modular, standards-based digital object repository. *The Computer Journal*, 48(5):514–535.