
EDEN: A Web-Based Model for Electronic Document Exchange

JOHN DURNO

ABSTRACT

The advantages of Web-based document exchange between libraries are just beginning to be systematically explored. This article focuses on general considerations in the development of a Web-based model for electronic document exchange (EDE) in the context of the OpenILL Cooperative's EDEN project. These include an overview of the existing document delivery standard (GEDI) and its relationship to emerging models and a discussion of factors being considered in the development of a Web-based protocol, including document exchange format, application event sequencing, metadata, and security.

INTRODUCTION

The spread of the Web and its associated hypertext transfer protocol (HTTP) have all but eliminated the technical difficulties associated with moving computer files from one place to another. For a variety of reasons, however, library document delivery networks do not currently take full advantage of HTTP, relying instead on the earlier file transfer protocol (FTP) for the interchange of documents between sites. HTTP, when it is used at all, tends to be employed in the final stage of the document delivery process, delivering content to end-users.

The advantages of Web-based document delivery to end-users have been widely discussed and documented (Schnell, 2000; Sayeed, Murray, & Wheeler, 2001). The advantages of Web-based document exchange between libraries are just beginning to be systematically explored. Atlas Systems has announced that its Odyssey document delivery software is being designed around a new open, Web-based protocol. And the OpenILL Cooperative's

EDEN (Electronic Document Exchange Network) project is focused on building an open source implementation of Web-based document exchange to work in conjunction with its open source interlibrary loan (ILL) management system. At the time of writing, neither project has yet published its protocol specification (although both may now be available).

This article focuses on general considerations in the development of a Web-based model for electronic document exchange (EDE). These include an overview of the existing document delivery standard (GEDI) and its relationship to emerging models and a discussion of factors being considered in the development of a testbed for the EDEN project, including document exchange format, application event sequencing, metadata, and security.

The term “document delivery” can be used to cover a wide range of activities. In this context I am using the phrase “document delivery network” to refer to a group of libraries capable of exchanging documents over the Internet and capable of receiving documents from commercial suppliers. It makes sense as well to limit the concept of “document delivery” to documents that are not directly accessible to end-users in print or electronically; typically this includes documents neither owned nor licensed by the user’s library or documents that are unavailable on the public Web.

EXISTING TECHNOLOGIES AND STANDARDS

Library document delivery networks typically rely on the use of specialized software created specifically for the purpose of streamlining the digitization and Internet transmission of print documents. Infotrieve’s Ariel software (formerly developed by the Research Libraries Group [RLG]) is by far the dominant player in this niche, and it is sometimes referred to as the “de facto standard” for document exchange between libraries (Frank-Webb, 2001). Consequently, it is common to define a “document delivery network” as a set of distributed workstations intercommunicating via Ariel or Ariel-type software.

A de facto standard is of course not a formal standard. Part of the reason that libraries have been slow to embrace a Web-based model for electronic document exchange has been that the de facto standard, Ariel, is built around a formal standard, GEDI (Generic Electronic Document Interchange, ISO 17933), that was finalized in the very early days of the Web. The first version of the GEDI standard dates from 1991, when the Web consisted of a handful of experimental nodes (Berners-Lee et al., 1994). Consequently, HTTP would have been on nobody’s radar screen when the standard was being worked out. There have been two subsequent versions of GEDI, in 1995 and 2000. The 2000 version permitted an alternate transfer protocol (email) and alternate file formats (PDF and JFIF [JPEG]). HTTP was not mentioned (International Organization for Standardization, 2000).

GEDI specified a standard file transfer protocol (initially FTAM, later FTP), a standard file interchange format (TIFF), and a standard format for metadata (the GEDI document header). Metadata was included as an SGML header prepended to the TIFF document, containing origin and destination information, document interchange format, and document description.

The GEDI standard was created to solve a particular problem: achieving interoperability between document delivery networks. In the late 1980s and early 1990s a number of separate agencies in Europe and North America were developing systems for electronic document exchange. As the number of agencies and networks increased, it was recognized that the development of incompatible systems would create a "Tower of Babel" impeding document exchange between disparate document delivery networks. These disparate networks were conceptualized as "domains" in the original GEDI Recommendation (Braid, 1994). GEDI was never intended to be a universal standard but rather a means of exchanging documents between domains. It was assumed that alternative means of transmitting and encoding documents would still be employed within individual domains; GEDI compliance was only needed to ensure interoperability between them (Braid, 1994).

As an example, the French FOU DRE domain at the time used STUDEL as its file transfer protocol while the British JANET used x.400 (email) for file transfer. The GEDI standard was proposed as a means of enabling document exchange despite these fundamentally different architectures by using the GEDI file transfer protocol as a bridge. A GEDI relay on the British side would receive documents via x.400 and forward them via the GEDI file transfer protocol to a relay on the French side. The French relay would then forward documents via STUDEL to their destination points (Braid, 1994).

As one of the agencies participating in the development of GEDI, RLG incorporated the standard into the design of its Ariel workstations in the early 1990s. Today's Ariel workstations send and receive documents formatted to comply with an updated version of the standard. However, Ariel's implementation of the GEDI standard does not conform to the original purpose of the standard as outlined above. Ariel implements GEDI primarily as an exchange format between proprietary workstations, not as a means of achieving interoperability between disparate systems or networks.

LIMITATIONS OF THE GEDI STANDARD

The persistence of the GEDI standard fifteen years after its initial conception may be interpreted as a testament to the fundamental soundness of its design. In fact, the standard reflects a number of good design decisions, notably its simplicity, its separation of metadata from the document body, and its integration with related International Organization for Standardization (ISO) ILL standards. However, the GEDI standard imposes limitations

on the design of document exchange networks and restricts their ability to make optimal use of current technologies.

The use of FTP for file transfer requires the presence of an FTP server on both the sending and receiving sides of the transaction. While installing an FTP server is not rocket science, adding an FTP server to a network generally requires the involvement of systems staff and may not be practical for smaller libraries with limited IT capabilities. In addition, FTP does not implement modern security protocols. FTP is widely regarded as insecure. Email was added as an alternative transfer protocol in an updated version of the standard; however, email systems often have policies (notably limits on the size of incoming attachments) that make them impractical for receiving large documents. (VanBuskirk & Caouette, 2000, p. 115)

The use of TIFF (and later PDF and JPEG) as the file format reflects the assumption that the documents libraries want to exchange are exclusively static and visual: journal articles and book chapters consisting primarily of text but also containing nontextual elements such as photographs, diagrams, and charts. In the past this was probably a safe assumption, but there is no reason to assume this will continue to be the case. With the proliferation of sound and moving image file formats, and the ever increasing availability of bandwidth and computer memory, it is inevitable that some of the documents libraries will wish to exchange will not fit comfortably into the current paradigm (Baker, 2002).

Another limitation of the GEDI standard is the assumption that a document can be represented as a single file or a collection of discrete files. For newer hybrid media that is likely not to be the case. A new standard should leave open the possibility of documents consisting of multiple, interrelated files.

THE EDEN PROJECT

The goal of the EDEN project is to develop an open protocol for Web-based electronic document exchange (Leggott, 2005). It is an outgrowth of the OpenILL project to develop an open source ISO-compliant ILL system, spearheaded by the University of Winnipeg in partnership with a coalition of academic libraries in western Canada. The testbed EDEN system is being designed to integrate with OpenILL using a plug-in, modular architecture that will enable stand-alone implementations of the software.

The document delivery transaction exists within the larger context of interlibrary lending, which has been formalized according to the ISO standards 10160 and 10161 (ISO ILL). Although the EDEN protocol will be designed to integrate with ISO ILL technologies, the goal is to design a protocol that will complement but not require ISO ILL (OpenILL Cooperative, 2003).

As noted above, EDEN is not the only project seeking to develop a Web-based document exchange protocol. Atlas Systems, developers of Odyssey document delivery software, has also announced the forthcoming publication of an open protocol for Web-based document exchange (the specification had not been released at the time of writing). Concerns that these separate but related projects will lead to the development of incompatible systems are probably premature. While the ultimate goal of any standard is widespread adoption in its application domain, a diversity of approaches in the early stages of development should allow for the emergence of a "best-of-breed" technology as the advantages of each are evaluated. As Tim Bray, co-author of the XML 1.0 specification, has noted, "a good standard is what happens when an industry has basically shaken the bugs out of a technology and then, after the fact, writes it down" (Bray, 2003). In any case, the independent emergence of similar projects indicates widespread interest in moving to a Web-based model and may be considered a strong predictor of further development in this area.

Details of the EDEN protocol will be worked out in the context of developing and implementing the testbed, reflecting Gordon Bell's assertion that "standards should be based on real experience, not on committee designs" (Bell, 2004, p. 73). If a working implementation is a precondition to a good standard, a specification is a precondition to a good implementation. To that end, the EDEN project will develop its specification through a process of broad consultation, soliciting feedback from as many stakeholders as wish to be involved. Development of the specification and the resulting implementation will be iterative, on the principle that deployment, testing, and feedback will undoubtedly necessitate changes to the original design. Successive versions of the testbed implementation will be released under an open source license to encourage wide participation in the project.

Adopting an iterative approach increases the likelihood of arriving at a result that is well fitted for its intended use. It also means that details of the implementation are likely to diverge in development from the model outlined below, which represents an initial pass at identifying the design requirements of the EDEN testbed.

RATIONALE

Before proceeding further, it may be worthwhile to take a step backward and ask why we need a protocol at all. As we have seen, the GEDI protocol was developed primarily as a means of conveying documents across disparate networks. At the time it was developed there were several competing file transfer protocols, and it was by no means obvious which of them, if any, would achieve dominance. In fact, it was by no means clear that TCP/IP, the underlying protocol supporting Internet protocols like FTP and HTTP, would achieve the ubiquity that it has (Hafner & Lyon, 1996).

In the balkanized networking environment of the early 1990s the GEDI model made sense. However, it is by now safe to assume that everyone with an Internet connection has the means to access documents via HTTP. Is a protocol for electronic document exchange still necessary? The answer is yes, but at least partly for reasons other than those for which GEDI was created. The purpose of developing an EDE protocol in the present day is not to enable document transmission across networks but to facilitate the exchange of documents between libraries in a controlled and systematic way. In the context of library interlending, a new EDE protocol must be designed to integrate as seamlessly as possible with library business processes and workflows and the ILL management systems and protocols supporting them. The goal of the protocol is to create system efficiencies for libraries on both the sending and receiving sides of the transaction or, more precisely, to enable developers to build systems to achieve that end.

Even if we agree that an open protocol is required to enable developers of different document delivery systems to intercommunicate, is the library-to-library transaction model still valid? It is possible to imagine a world in which suppliers would deliver documents directly to end-users with no need for the requesting agency to act as the intermediary. This is happening to some degree already. Perhaps this is the future we should be moving toward, rather than staying with the library-to-library model.

The EDEN initiative is predicated on the assumption that a library-to-library model is still required, even if direct delivery is an option. There are several reasons why direct delivery may not always be the optimal approach. These include the following:

- Privacy: users may not wish to have their contact information made available to third-party suppliers
- Convenience: the client library may wish to make all requested documents available through a central service point, whether that is the library circulation desk or its Web portal
- Law: some jurisdictions prohibit direct delivery of digitized content to end-users; the library is required to print it first
- Accountability: the client library may wish to confirm that requested documents have in fact been received
- Responsibility: serving the user is the client library's role, not the supplier's. Some suppliers may not be willing to serve another library's clientele, particularly if it means storing unclaimed documents on their server for extended periods of time

REQUIREMENTS

It is important to distinguish between a protocol and its implementations. Protocols dictate the behavior of systems to a degree, but systems

with widely varying capabilities can be built on top of the same protocol. Successful widespread adoption of a protocol depends in part on its relative simplicity and the degree to which it can be implemented using common and widely available technologies.

Design of the protocol must also reflect a consideration of the diverse contexts in which it is likely to be implemented. The scale of a given library's interlending operations to a large extent determines its business processes. Achieving system efficiencies may mean something very different in the context of a small branch library than it does in the context of a large university ILL department. The latter has a strong incentive to build and maintain complex systems to help staff manage workflow; the former may find the volume of documents to be processed is not large enough to warrant it. The protocol must permit both low-volume and high-volume implementations.

In the context of the ubiquitous Web, using HTTP as the transfer protocol for EDE makes sense. However, in itself moving to HTTP does not require the development of a new protocol. As noted above, the GEDI standard has already been updated twice with the addition of alternative transfer protocols (FTP and email). If the goal is to move to HTTP transport, perhaps the simplest way to achieve this would be to update the existing standard rather than developing a new one. However, the goal is not simply to move to a new transport protocol; the goal is to streamline document exchange between libraries. HTTP is only part of the picture.

As noted by Chari and Seshadri (2004), achieving interoperability between applications involves multiple levels:

- Transport, which handles the movement of data between applications
- Data format, which ensures consistency of data representation between applications
- Process, which coordinates the sequencing of events between applications

The GEDI standard covers two of these layers—transport and data format—which are referred to as “Interchange Mechanism” and “Electronic Document Format” within the standard.

GEDI does not specify “process”—the sequencing of events that must occur between the document supplier and receiver at the time of document transmission. This is key to establishing a truly open protocol. If third-party developers cannot predict sequencing, interoperability may and very likely will require customized event handling for every preexisting implementation. It may even require a formal agreement between the developers of different systems. The new protocol will cover all three levels required to achieve true interoperability between document delivery systems, representing a true and important departure from its predecessor.

WEB SERVICES

The decision to use HTTP as the transport protocol for EDE reflects the prevailing trend in the broader information technology (IT) community to employ Web Services to achieve interoperability between systems. This decision is in part strategic, reflecting the requirement that developers should be able to construct implementations using widely available technologies. The broader IT community is much bigger than the library IT community; it makes sense to adapt existing technologies wherever possible rather than building our systems from scratch.

Web Services support interactions with other “software systems . . . using XML based messages conveyed by Internet protocols” (W3C Web Services Architecture Working Group, 2004, chap. 1.1). XML messaging is an efficient platform- and language-independent way to exchange messages between applications. The technologies required to build Web Services applications are readily and often freely available: Web servers and clients, XML processing libraries, and programming toolkits have been developed for many platforms. Web Services are commonly seen as the foundation of the new generation of B2B (Business-to-Business) software applications; it follows that Web Services will likely be useful in the context of developing L2L (Library-to-Library) applications, of which EDE is one.

TESTBED ARCHITECTURE

Interchange is only one component of the complete document delivery cycle. Other components include discovery, ordering, digitization, printing, and administration (billing). Integrated document delivery applications typically handle several of these components.

A modular architecture is seen as key to developing a successful testbed implementation of Web-based EDE. Existing document delivery software often merges the separate facets of the document delivery transaction into a single application: scanning, applying metadata, document transmission, reception, and processing are all handled by the same program. While this architecture may be an effective design for handling library workflow, it will be more useful in the present instance to disaggregate these functions in order to focus as much as possible on document transmission, the core of the EDE protocol. The testbed application will develop only those functions necessary to prototype Web-based document exchange.

A modular architecture may be useful in a production environment as well. Separating the document transmission and scanning modules would enable them to reside on separate machines, which could have advantages for enhancing both security and efficiency. For example, a document scanning module could be installed on a machine within an organization’s firewall, while the transmission module could reside on the organization’s Web server. In fact, this architecture would permit the transmission module

to be installed on a third-party network, which could be a boon to smaller sites operating within a consortium, multibranch public libraries, and multi-campus schools. One installation could serve multiple libraries.

FILE EXCHANGE FORMAT

GEDI specified a standard file format for document exchange. As noted above, the format consisted of a binary image file (TIFF, PDF, or JPEG) accompanied by metadata in the form of a prepended header. The header and binary image file together constituted a new file type, requiring specialized software to process them. A GEDI-formatted PDF is typically no longer readable by applications designed for the purpose, such as Adobe Acrobat Reader. This constitutes another limitation of the GEDI standard: the transmission format is not compatible with common desktop applications.

In order to simplify the document exchange process, it is desirable that all metadata travel with the document and not be sent as a separate transaction. This was reflected in the GEDI standard. Is there a way to achieve this without creating a new file type? In fact, it is done all the time. Widely available software tools exist to package multiple files. These include archiving utilities, such as tar, and compression utilities, such as gzip. Both tar and gzip are available in open-source implementations and do not employ proprietary algorithms, which would require the payment of royalties. Utilities for expanding gzipped tar archives are freely available for common desktop platforms such as Windows, Mac OS, and Linux. Therefore, EDEN will specify that documents be exchanged as one or more binary files accompanied by a separate text file containing metadata marked up in XML. All files associated with a single document delivery transaction will be in a compressed archive format, initially tar/gzip.

DOCUMENT METADATA

Although the EDEN protocol is intended to either complement or supersede GEDI, it is anticipated that EDEN will benefit directly from the work that went into defining its predecessor. The GEDI standard defined a range of metadata in the document header. These elements, many of them optional, were grouped into five types:

- Type 1: identifying information about the Document Interchange Format itself
- Type 2: naming and time information for the Transfer Mechanism
- Type 3: other information about the particular Electronic Document Delivery Transaction
- Type 4: information specific to the document, including a brief bibliographic description

- Type 5: padding to allow for subsequent changes to the header without changing the header length (optional)

Of the five types identified above, only the last is clearly no longer required by EDEN. The GEDI Header is marked up in SGML, the precursor to XML. It is feasible to replicate the GEDI Header elements in XML should that prove to be desirable. In any case, it is expected that the elements defined in the GEDI Header will form the starting point for identifying elements to be included in EDEN metadata.

PROCESS SEQUENCING

For ease of implementation, the EDEN process governing document exchange transactions is designed to be as simple as possible. In the initial iteration of the testbed application, events will proceed according to the following sequence:

1. When a document is available to be sent, the supplier notifies the client system. The notification consists of a Uniform Resource Identifier (URI) pointing to the location of the document. The URI contains at a minimum a unique transaction ID generated by the supplier. The transaction ID will be returned to the supplier in all messages from the client system. The notification may also contain a checksum to be used by the client system to verify successful transmission of the document.
2. When the client receives a notification of document availability, it may return an optional confirmation that the notification has been received.
3. The client retrieves the document from the URI provided in step 1.
4. The client notifies the supplier that the document has been successfully retrieved. If within a set interval the supplier receives neither a confirmation of receipt of the availability notice, nor a confirmation of successful document retrieval, the supplier may send out additional notifications of availability until such time as the document has been purged from the supplier's system. If the document appears to have been corrupted in transmission, the client system may re-request the document.

Documents are purged from the supplier's system after an interval determined by the supplier based on local conditions, in particular the availability of storage space. The supplier may choose to purge a document any time after the confirmation of successful document retrieval has been sent by the client system.

Note that the above sequence does not cover document preparation, as that is expected to be specific to a given implementation. A document is available to be sent when it has been properly formatted with the required metadata and uploaded to an EDEN-compliant server. Document handling following retrieval is also expected to be implementation specific.

SECURITY

Several security considerations need to be taken into account in designing an EDEN implementation. The transmission process described in the foregoing section is insecure insofar as the document to be transmitted exists briefly on the public Web. When a document becomes available, any Web client, including a standard desktop browser, is capable of accessing it. However, this is mitigated to a degree by the fact that documents will come and go rapidly and the URIs are not published except to the client system.

Additional security may be obtained through randomizing transaction IDs. If transaction IDs were to consist of random strings run through a one-way encryption algorithm such as MD5, it would be effectively impossible for third parties to guess them. Security through obscurity is not generally thought to be the best policy but, in the case of the typical materials exchanged via DocDel, it may be good enough. If the payoff to cracking an EDEN implementation consists of access to random research papers, it is doubtful that anyone will expend much effort on it.

A much higher level of security could be obtained through the use of public key encryption. EDEN documents could be encrypted by the supplier with a public key supplied by the client. This would effectively block document access to anyone not in possession of the client's private key. Even if documents were intercepted in transmission, they could not be read. It is questionable whether this level of security is desirable, but if it proves to be necessary EDEN systems could be built to run in encrypted mode.

Security considerations also exist on the client side of the transaction. Here, the key consideration is whether the supplier is a trusted source. In the process described above, the client has no way to know in advance if the document being supplied is related to an outstanding request or not. If the supplier is not trustworthy, the download might be not a document at all. It could be spam, a virus, or a trojan horse. This is true for GEDI-based document delivery systems as well, although risk is mitigated somewhat by the hurdles of participating in existing GEDI-based document delivery networks. Proprietary software and unusual document formats might not completely prevent abuse, but they probably raise the bar high enough that spammers and crackers will continue to choose easier avenues of attack.

One way to limit abuse in an EDEN system would be to require the client to supply its own transaction ID at the time a document was requested. The client's transaction ID would be returned by the supplier along with the notification of availability. If the transaction ID was not present in the notification, the client could simply choose not to retrieve the document. This would require untrustworthy suppliers to guess the client's outstanding transaction IDs in order to complete a successful file transfer.

The problem with this approach is that the client's document requests occur outside the document transmission process as described above. Depending on how well the supplier's ILL and DocDel systems are integrated, including the

client's transaction ID in the notification of availability could require human intervention. Apart from being additional overhead, manual rekeying could introduce errors that would cause the process to occasionally fail.

A better approach would be for the client to maintain a list of trusted suppliers. Servers not in the client's supplier list would be considered untrustworthy. Documents from unlisted suppliers would either not be retrieved at all or retrieved and flagged until their status could be verified.

Finally, proper document handling by the client system can go a long way toward mitigating the dangers posed by external binaries. Documents will arrive in the form of compressed archives, posing no immediate danger to the client system. The XML metadata included with the file can be parsed without expanding the archive and matched against outstanding requests even before the file is processed. Obviously, incoming files will be stored outside the Webserver's document tree; the testbed implementation will store incoming files as blobs in a relational database, effectively neutralizing any executable code. A production system could also scan incoming files for virus signatures.

CONCLUSION

Work on the GEDI standard was partially funded by the European Commission and developed by representatives from the Online Computer Library Center (OCLC), RLG, the Ministère de l'éducation nationale, de l'enseignement supérieur (MENESR), Questel, Telis, the Universitätsbibliothek/Technische Informationsbibliothek (UB/TIB), Pica, and the British Library Document Supply Centre (BLDSC). Its testbed, EDIL, took two years to implement, at a cost of \$2.5 million. The testbed successfully demonstrated the feasibility of using GEDI for document exchange across dissimilar technical environments: over 1,000 documents were exchanged over a period of several months. Despite the successful implementation, the associated costs and the general shift in the mid-1990s toward electronic publishing discouraged further implementation of GEDI as a cross-domain EDE protocol (Braid, 1995).

In contrast, the EDEN protocol will be developed by an ad-hoc group of interested participants based, at least initially, in western Canada, with development work to be carried out by the University of Winnipeg. The first version of the EDEN testbed was expected to be operational in mid-2005, six months after the project was announced.

The difference in scale and timeline reflects the quantum leap forward taken by networking and related applications since the early 1990s. In part, developing the EDEN protocol will be easier simply because aspects of the GEDI design can be repurposed in the present context. But more importantly, the global spread of the World Wide Web provides a uniform environment that will greatly reduce the amount of work required to achieve interoperability. Finally, the ready availability of the software tools

and applications needed to build a testbed implementation means that development will largely consist of assembling preexisting components. Much of the heavy lifting has already been done.

Whether or not the EDEN protocol becomes widely adopted, the project will be considered a success if it can demonstrate that library-to-library EDE is readily achievable using common tools and technologies. Hopefully, it will help to spur the creation of a new generation of library EDE applications that will move beyond the current proprietary model to attain true interoperability.

REFERENCES

- Baker, D. (2002). Document delivery—Breaking the mould. *Interlending and Document Supply*, 30(4), 171–177.
- Bell, G. (2004). A time and a place for standards. *Queue*, 2(6), 66–74.
- Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H. F., & Secret, A. (1994). The World-Wide Web. *Communications of the ACM*, 37(8), 76–82.
- Braid, A. (1994). From Babel to EDIL: The evolution of a standard for document delivery. *Computer Networks and ISDN Systems*, 27, 367–374.
- Braid, A. (1995). Standardisation in electronic document delivery. In Geh, H. P. & Walckier, M. (Eds.), *Proceedings of the European conference "Library networking in Europe"* (pp. 157–166). London: TFPL.
- Bray, T. (2003). *No innovation, please*. Retrieved January 14, 2005, from <http://www.tbray.org/ongoing/When/200x/2003/05/10/RSS-std>.
- Chari, K., & Seshadri, S. (2004). Demystifying integration. *Communications of the ACM*, 47(7), 58–63.
- Franke-Webb, J. (2001). Using DocMorph in conjunction with Ariel to expand digital document delivery options. *Journal of Interlibrary Loan, Document Delivery and Information Supply*, 12(1), 85–92.
- Hafner, K., & Lyon, M. (1996). *Where wizards stay up late: The origins of the Internet*. New York: Simon and Schuster.
- International Organization for Standardization. (2000). *GEDI—Generic Electronic Document Interchange [ISO 17933:2000(E)]*.
- Leggott, M. (2005). *Freeing ILL systems with EDEN*. Retrieved January 12, 2005, from <http://blog.uwinnipeg.ca/loomware/archives/000564.html>.
- OpenILL Cooperative. (2003). *EDEN—Electronic Document Exchange Network*. Retrieved March 14, 2005, from <http://www.openill.org/software/eden.cfm>.
- Sayeed, E. N., Murray, S. D., & Wheeler, K. P. (2001). The magic of Prospero. *Journal of Interlibrary Loan, Document Delivery and Information Supply*, 12(1), 55–72.
- Schnell, E. H. (2000). Freeing Ariel: The Prospero Electronic Document Delivery Project. *Journal of Interlibrary Loan, Document Delivery, and Information Supply*, 10(2), 89–100.
- VanBuskirk, M., & Caouette, D. H. (2000). Ariel in a high-volume environment: How CISTI has integrated Ariel into its document delivery business. *Journal of Interlibrary Loan, Document Delivery, and Information Supply*, 10(4), 113–119.
- W3C Web Services Architecture Working Group. (2004). *Web services architecture requirements*. Retrieved January 17, 2005, from <http://www.w3.org/TR/2004/NOTE-wsa-reqs-20040211>.

John Durno is a project coordinator of the BC Electronic Library Network, a consortium of postsecondary libraries in British Columbia, Canada. His professional focus is on Web development, systems interoperability, distance education technologies, and licensing. Previous publications include articles on file-sharing technologies and open content licensing.