Born-Again Bits

A Framework for Migrating Electronic Literature

beta v1.13 September 30, 2004

By Alan Liu, David Durand, Nick Montfort, Merrilee Proffitt, Liam R. E. Quin, Jean-Hugues Réty, and Noah Wardrip-Fruin

Alan Liu (UC Santa Barbara)
David Durand (Ingenta and Brown University)
Nick Montfort (University of Pennsylvania)
Merrilee Proffitt (Research Libraries Group)
Liam R. E. Quin (W3C)
Jean-Hugues Réty (Université de Paris 8)
Noah Wardrip-Fruin (Brown University)

Contents

B.4 Other Resources Cited

Sidebars (Glossary Definitions): **Base-64**, **Emulator**, **HyperCard**, **Interpreter/Reader**, **Metadata**, **Open source**, **Platform**, **Porting**, **Source code**, **Storyspace**, **XML**, **XML schemas** (Quotations): **Brian Lavoie and Lorcan Dempsey**, **Catherine C. Marshall and Gene Golovchinsky**

Preface: *Born-Again Bits* and the ELO PAD Project

Acid-Free Bits by Nick Montfort and Noah Wardrip-Fruin (June 2004) was the first publication on digital preservation to emerge from the Electronic Literature Organization's Preservation, Archiving, and Dissemination (PAD) initiative. Addressing primarily the community of electronic literature authors, it concentrated on prescribing standards and best practices that creators can follow to prepare for "keeping e-lit alive."

With the release of *Born-Again Bits*, ELO continues the argument by envisioning a technical framework that can not just keep e-lit alive but allow it to come back to life in new forms adapted to evolving technologies and social needs. The intended audience of *Born-Again Bits* includes besides e-lit authors also the publishers, archivists, academics, programmers, and funding officers who will be necessary partners in an overall, renewable ecology of electronic literature. These other communities are already at work on digital preservation strategies. However, experimental e-lit has special qualities that make it an extreme case of the digital artifact. It is hoped that ELO's PAD initiative will contribute to other digital preservation strategies by ensuring that they accommodate e-lit and so, in the process, become more robust for all digital works.

Born-Again Bits had its origin in the work of the PAD Technology/Software Committee (directed by Alan Liu), which in 2002 and 2003 prepared a report for ELO proposing strategies for the long-term preservation of electronic literature. *Born-Again Bits* distills the conclusions of that report into a two-part plan: the ELO Interpreter and X-Literature Initiatives. The specifics of the plan are imagined less as hard-and-fast commitments than as a way to flesh out what a general approach might look like. Though necessarily technical at some points, the overall goal of *Born-Again Bits* is to allow diverse stakeholders (authors, publishers, archivists, academics, programmers, grant officers, and others) to get just enough of a glimpse of each other's expertise to see how an overall system for maintaining and reviving the life of electronic literature might be possible.

1 Bringing Electronic Literature Back to Life: Preservation by Migration

Though much can be done with existing technologies, standards, and practices to give electronic literature a longer life, there will inevitably come a time when changes in hardware, software, and other factors accumulate to the point that keeping the patient on life support is no longer feasible. E-lit, after all, has only been alive a few decades. How much of its corpus will be alive (in the basic sense of readability) in fifty years, or a hundred?

The stakes are even higher when we consider that keeping works of electronic literature alive in their original form does not serve all present needs, let alone those of the future. There are many conceivable uses of e-lit that would be facilitated if works could migrate as needed into other forms. For example, instructors who wish to teach e-lit are now often faced with intractable difficulties when showing works in the classroom in real-time. (Many works cannot be easily navigated, linked to, or shown in such a way that the instructor can jump quickly to a particular section or play back a particular reading.)

For all these reasons, it is useful to think not just of keeping electronic literature alive, but of giving it new lives—of allowing "born-digital" literature to be reborn. The long-term preservation and dissemination of e-lit requires a strategy of hardware and software *migration*.

Defining an appropriate technical and institutional framework in which preservation-by-migration can reliably occur requires first addressing the following questions.

1.1 What Is the object of migration?

Much of the confusion now surrounding digital preservation stems from uncertainty about what is the proper object of preservation—for example, the "work," a "version" or "state" of a work, a work's constituent files, the original "reading experience," documentation about a work, the original software and/or hardware environment, and so on.

From the point of view of long-term digital preservation, however, the entity of interest is not necessarily any discrete object but the working *relationship* among objects (each of which may mutate) that assures readability. This means that the intact "original work" in its initial instantiation (for example, a work authored for HyperCard, Storyspace, or a particular generation of Web browsers, Javascript, Flash, and so on) loses its iconic status and becomes just one of many possible manifestations of a preserved work.

Complex digital works are a kind of swarm behavior. Individual files, formats, scripts, software environments, and so on, may perish, but suitable replacements may be found that allow the living relationship that is the swarm to continue.

1.2 Who will migrate electronic literature?

The migration of electronic literature must occur in a framework that accommodates not just swarming technical changes but equally complex, swarming social needs. The players in the game, after all, will not just be the original authors and readers but also future users with more diverse, autonomous needs—for example, secondary authors or remixers (who might create, for example, works dynamically quoting or aggregating other works), publishers, editors, distributors, instructors, students, and collective users (as in the setting of a classroom or reading society). Indeed, even the burgeoning league of software agents, Web services, RSS readers, and other instances of what might be called machinic "users" (automated ways of distributing, parsing, and repackaging information) will need to be considered as virtual members of the society of e-lit.

Because the long-term digital preservation of electronic literature is such a complex technical and social equation, it will not be the responsibility of any single stakeholder community. The job will not be done by authors, librarians, publishers, or programmers acting separately.

> "Our understanding of the totality of the challenges associated with maintaining digital materials over the long-term is coming more sharply into focus. New questions are emerging, having less to do with digital preservation as a technical issue *per se*, and more to do with how preserving digital materials fits into the broader theme of *digital stewardship*. These questions surface from the view that digital preservation is not an isolated process, but instead, one component of a broad aggregation of interconnected services, policies, and stakeholders which together constitute a digital information environment." —Brian Lavoie and Lorcan Dempsey, **"Thirteen Ways of Looking at . . . Digital Preservation,"** *D-Lib Magazine* (July/August 2004)

The job can only be done through the collaboration of multiple stakeholders and their institutions (organizations such as ELO, research libraries, universities, software firms and consortiums, and so forth). As in the case of other digital preservation initiatives originating in the library or museum worlds (see **Related Initiatives**), the migration of e-lit will require collaborative institutional relationships and shared technical standards.

The unique mission of electronic literature organizations or programs in such a multi-institutional framework will be to serve as the catalyst for the creation of standards specific to e-lit that no other organization makes a high priority.

1.3 What are the specific challenges of electronic literature to migration?

> "Hypertext fiction occupies a provocative niche in defining requirements and testing solutions for the immense problem of digital archiving. . . . Not only is hypertext fiction a literary effort, it may also represent a software development effort, a sophisticated and often unconventional use of different kinds of digital media, a visual design component, and an exercise in interaction

> design that may even involve special types of platforms and hardware," Catherine C. Marshall and Gene Golovchinsky, **"Saving Private Hypertext: Requirements and Pragmatic Dimensions for Preservation,"** *Proceedings of ACM Hypertext 2004* (August 9-13, 2004)

Many technical solutions are being developed by humanities computing scholars and information-science researchers to ensure that digital media will have a longer "shelf life." However, as the shelf metaphor might indicate, these solutions (for example, the Text Encoding Initiative's TEI schema or the library METS metadata standard) are often currently better suited for print, or print-like, static works that have been digitized than for born-digital artifacts of electronic literature with dynamic, interactive, or networked behaviors and other experimental features—including, but not limited to, works making use of hypertext, reader collaboration, other kinds of interaction, animated text or graphics, generated text, and game structures. (**Note 1**) (See ELO's **Electronic Literature Directory** for representative categories of e-lit.) Not only are there relatively few standards for the archival maintenance of such works, but there often is not even a common descriptive vocabulary for the phenomena they exhibit (what Matthew Kirschenbaum, at the *e(X)Literature* conference in 2003 for the ELO PAD initiative, typified as "that squiggly, jumping thing at the top of the screen").

The migration of e-lit will require adapting existing solutions and inventing new ones suited to e-lit.

1.4 What are the main strategies for migration?

> **Interpreter / Reader**: A computer program that takes as input an original electronic literature work's data file (e.g., a HyperCard stack, a Storyspace file, an interactive fiction Z-machine story file) and runs the work so that it can be experienced in an interactive session as it originally functioned. Existing examples of such readers / interpreters include HyperCard Reader, StorySpace Reader, and the Frotz Z-machine interpreter. **Emulator**: A computer program running on platform B that takes as input the binary files that can be run directly on platform A and runs them as they would have run on platform A. An emulator running on platform B is a software implementation of platform A. Existing examples include AppleWin and Cataking (Apple II emulators for Windows and Mac). (**More on Emulators**) **XML** is a markup language designed to create structured representations of textual data (with much of the logical rigor and extensibility of its predecessor, SGML) within a distributed, networked, automated, and multiple channel or display environment. Complemented by its various schemas (or use-specific vocabularies), XML is now the dominant format for structuring textual information. It has seen extremely widespread adoption in both the non- and for-profit realms, and there are many implementations both open source and proprietary. XML is an unencumbered format that can be

freely and openly implemented. (**More on XML**)

One strategy for migration is to *interpret* or *emulate* electronic literature so that works now difficult or impossible to read can be experienced once more in a form as functionally like the original as possible (see also *Acid-Free Bits*, **§ 3.2**).

The other strategy is to *describe* or *represent* works—for example, in XML—so as to facilitate moving them into alternative formats and software (see also *Acid-Free Bits*, **§ 3.4**). This representational method may not always be able to maintain all the functions of the original work. But even so, it has the advantage of being standardized (for interoperability); and it can supplement or enhance the workings of the original. For instance, XML applications could be designed to provide more eloquent and standard methods of reading, navigating, citing, annotating, saving state, searching, or indexing in such databases as the ELO's **Directory of Electronic Literature**.

To imagine what a framework for the long-term preservation and migration of electronic literature might look like, ELO has sketched out a twofold plan that draws upon both the above strategies. The two branches of the plan are the *Interpreter Initiative* and *X-Literature Initiative*. Each is presented below through an overview, technical analyses of issues, and conclusions with implementation recommendations.

2 Interpreter Initiative

Many early works of electronic literature created in extinct hardware or software systems can best be preserved by programming **interpreters** (and/or **emulators**) that run the works on new computers "as if" they were in their original environment.

It's as if a museum exhibited some strange, early electrical device from before the standardization of electricity in the United States, one that couldn't be plugged directly into today's power grid. Building an entire early power grid for the device would be extremely impractical. But a voltage adapter could be created to allow the old device to run using a modern, standard outlet.

ELO proposes the development of open source interpreters to "run" important or populous categories of e-lit—for example, Hypercard—so as speedily to restore large numbers of older works to readable status. Secondary priorities include the development of additional interpreters (including high-priority but technically challenging ones), assisting open source communities working on relevant emulators, and creating supporting documents and services for software interpreters.

2.1 Technical Analysis of Interpretation/Emulation

There are several ways to approach interpreting or emulating electronic literature. These strategies may be grouped under the rubrics of "per-work" techniques (porting and reimplementing) and "per-category" techniques (interpreting and emulating proper), where the former method targets individual works and the latter classes of works.

2.1.1 "Per-Work" Techniques

Porting works directly

> **Porting**: "In computer science, porting is the adaptation of a piece of software so that it will function in a different computing environment to that for which it was originally written. Porting is usually required because of differences in the central processing unit, operating system interfaces, different hardware, or because of subtle incompatibilities in—or even complete absence of—the programming language used on the target environment" (*Wikipedia*). **Source Code**: Many common programs are written in a high-level programming language, such as C or C++, and then compiled into binary form. The uncompiled form, which programmers directly write, is the source code. If you have access to the source code, you can compile the program yourself, and you can modify the source code if you wish to make the program do something different. (Or, if you are not a programmer, you can hire someone to do this for you.) Some simple programs will compile on both Mac OS X and Linux with no changes, for instance, so in some cases you can take a program written for one system and directly compile it to work on another. If changes are needed, as is usually the case with complex programs (including ones that use a graphical interface) or when very different operating systems are involved, the source code can be modified so that it compiles on the new system. This process is called porting.

Porting involves converting the source code of an electronic literature work. Such conversion, however, is only an option when the source code is available. If all that is available is an executable program, an extensive effort would in most cases have to be made to reverse-engineer and reimplement the program before it could be ported. The effort required in porting software can be great, and porting one particular work would not help to make any other works available. Also, when one port has been completed, this may not make it that much easier to port the work to a different platform, either now or in the future. *Porting will probably be used for preservation only in rare but important cases.*

Reimplementing the work

Reimplementing involves writing a new program that does the same thing as the original program. It can be difficult to ensure that the new program functions identically, but in the case of works that are well documented, and particularly when the authors are available for consultation, this strategy may be feasible. Performing a reimplementation today, when the original work is still available interactively, can be much easier than trying to reimplement the work later on, when no working version is present. If a reimplementation is open source, then it may be easy to port that reimplementation in the future. The source code of such a reimplementation may be much cleaner than the source code of a port of the original. For example, in the case of some hypertext electronic literature, the reimplementation of an older work can be achieved using the **Connection Muse** and open Web technologies. *Reimplementation will probably be used for preservation only in rare but important cases*.

Summary of "per work" techniques

"Per-work" techniques will no doubt continue to be used occasionally by those working in new media preservation, but because they are resource-intensive and only result in the preservation of one work at a time (that is, one work per each particular software development effort) they will likely not be the focus of long-term digital preservation efforts. Instead, such preservation will focus on software development that makes whole categories of work accessible.

2.1.2 "Per-Category" Techniques

Creating an open source interpreter

Many works of electronic literature run on "virtual machines" (that is, software computers), "players," "readers," or other sorts of **interpreters**. For instance, a HyperCard stack is an interpreted program that can be accessed using Apple's HyperCard Player. Storyspace similarly uses Storyspace Reader. These are the most obvious examples in electronic literature, but there are many others. For instance, interactive fiction works today almost all run in interpreters, the Z-Machine and TADS being the most common. The most popular general purpose interpreter system of this sort now in use is the Java VM (virtual machine).

> "**HyperCard** was created by Bill Atkinson and initially released in 1987. . . . HyperCard is one of the first products that made use of and popularized the hypertext concept to a large popular base of users. . . . HyperCard is based on the concept of a 'stack' of virtual 'cards.' Each card includes fields that store data, and the pattern for each card (its layout, as opposed to the data in the layout) is known as the 'background.' Backgrounds could include pictures . . . , picture fields, buttons, text, text fields (editors) and other common GUI elements, which would then be copied onto new cards." —***Wikipedia***
>
> "**Storyspace** is a hypertext writing environment that is especially well suited to large, complex, and challenging hypertexts. . . . Storyspace provides a variety of maps and views to help writers create, organize, and revise." — **Eastgate Systems, Inc.**

One preservation approach that can be very effective is to develop new, open source interpreters for obsolete or near obsolete electronic literature systems. If a HyperCard interpreter is developed that runs on Windows and Linux, for instance, a massive readership will suddenly be given the means to access all HyperCard works. Many HyperCard works are now available for free on the Web (although not accessible even to many Mac users) and these will be readable immediately. Some others (such as *Uncle Buddy's Phantom Funhouse*) are still available commercially and could be ordered by Windows and Linux users, who could use the new interpreter to access them. Of course, if there is no means for people to get access to the HyperCard stacks that constitute the original electronic literature work, the interpreter will not help. But in any other case, a new interpreter will result in a much larger group of users being able to experience classic works of electronic literature.

The approach of developing a free, open source interpreter only applies to those works that do run in an interpreter of some sort. The benefits of this approach fall off as the number of works per interpreter approaches one. In the case where there is only one electronic literature work that runs on a particular interpreter, it may be just as easy to **reimplement** the work—although, even then, there could be factors that make development of a new interpreter a simpler and easier task than other sorts of reimplementation. Robert Pinsky's *Mindwheel* was written in BTZ, an interpreted language that was used to create only four works of interactive fiction. Another interactive fiction work by a notable print author, *The Mist* by Stephen King, was one of only a handful of works written in ASG. Further study is necessary to determine whether it would be worth the investment to develop interpreters for such works.

In the case of HyperCard, the value of a free interpreter is more obvious. A very cursory search turns up electronic literature works by John Cayley, William Dickey, Clark Humphrey, Deena Larsen, John McDaid, Stuart Moulthrop, Michael Murtaugh, David Rokeby, Jim Rosenberg, Matthew W. Schmeer, and Sarah Smith. It seems certain that more than a hundred electronic literature works in HyperCard exist, many by top electronic literature authors. The development of a single interpreter program would thus allow large numbers of today's users to access these authors. Currently, HyperCard works can be accessed on Macintoshes in Classic mode, but it is clearly not a priority for Apple that HyperCard remain functional in future Mac OS releases. Apple has also recently refused permission to academics seeking to redistribute the HyperCard Player. *The development of a HyperCard interpreter would be a highly visible and effective way to make a large body of older electronic literature accessible and would have an immediate effect in the classroom, where substantially more works would be made available for study.*

Creating an open source emulator

An emulator is a program that effectively implements a hardware computer in software—well enough that binary programs for that computer can run in the emulator. For instance, Stella is an emulator that implements the Atari 2600. The actual sequence of bits stored on an Atari 2600 cartridge can be loaded into Stella and the program can run them as if it were that video game system with that cartridge inserted into it. The user uses the computer's keyboard or joystick rather than the famous black plastic Atari joystick, and the computer monitor is used as a display, not a TV. But otherwise the experience is quite similar to the original. Stella adjusts its timing automatically so that the speed at which games run is about the same as on an Atari 2600, no matter what computer is used to run Stella. An Atari 2600 game in Stella looks, feels, and functions much the same as the original on the authentic console. For a student of early-1980s culture or a scholar of game studies, the experience provided by Stella is far more valuable than documentation alone would be. It is possible to emulate more powerful computers today. For instance, there are several Apple II emulators available, providing access to Apple II software, including early electronic literature works.

Developing an emulator is usually more difficult than developing an interpreter because a host of new issues (including timing issues) emerge when the hardware level must also be considered (**Note 21**). Yet many emulators do currently exist, and readers, students, and scholars of electronic literature already use emulators to access works. Users will undoubtedly benefit from emulators in the future.

Summary of "per category" techniques

A digital preservation initiative for electronic literature would probably not by itself take on the development of a new emulator, since emulators are general-purpose instruments. Instead, such an initiative could contribute to existing emulator development efforts to help ensure that works of electronic literature function properly in their products. The case is different with interpreters, however. Some interpreters are mainly used to interact with electronic literature, or their uses along these lines are particularly important. The development of new interpreters could be an important function in a preservation initiative focused on electronic literature.

2.1.3 Conclusions of Technical Analysis of Interpretation/Emulation

Open Source software has the following condition attached to it by means of a software license: anyone who receives the executable program must also be given access to the source code. Open source software is not the norm for personal computer software sold in stores. Many commercial companies distribute software to their paying customers and do not provide access to the source code. However, source code has usually been provided along with academic computer software that is distributed freely.

Given the above alternatives, the highest priority is to develop a set of open source (GNU GPL, "General Public License") interpreters for important kinds of electronic literature. (Assisting open source communities in creating emulators is also important, but a lesser priority.) Such a development effort will have the benefit of a near-term payoff that will immediately make accessible a large number of important early e-lit works. Front loading development in this way will be important in winning acceptance for e-lit preservation efforts among stakeholder communities and funding organizations (**Note 3**).

2.2 Implementation Plans for Interpreter Initiative

The Interpreter Initiative could initially select at least two interpreter projects. Even if unforeseen difficulties (technical or legal) obstruct one project, it should be possible to complete one interpreter and see the result of increased access within a year. In addition, it is wise to develop two different interpreters simultaneously on the general principle (which may be called the "dual paradigm rule") that development within any category of a digital preservation plan should target at least two kinds of e-lit works simultaneously even if the second kind includes fewer works. Such a procedure will prove concepts on a broader baseline and so protect against fragile, narrowly premised approaches that break down the first time they encounter an unexpected variant (**Note 4**).

The two, specific interpreter projects that could be pursued are as follows:

2.2.1 Create an Interpreter for HyperCard

> A **platform** is the environment in which a program runs. The "system requirements" on the side of a box of commercial software describe a platform: what basic hardware is required, what operating system, what version of the operating system, and what special extensions or special-purpose hardware. Digital preservation efforts are concerned not only with today's platforms but also with platforms that may exist several decades in the future.

Apple's **HyperCard** for Mac was a favorite system for early electronic literature creators and is an obvious choice for an initial interpreter project. A free, open source HyperCard player could be developed for Windows XP, Linux, Mac OS X, and Java platforms. In a funded preservation project, one or two full-time software developers should be able to complete the project within a year.

2.2.2 Create Interpreters for other candidate systems

Storyspace 1.0

Many early electronic literature works were written in Storyspace for Mac and will not run on Windows. However, the most famous of these works are not freely available. The development of a free player should be contemplated most seriously if Eastgate Systems, Inc. were willing to allow free distribution of some of the early works for which it owns the rights (in exchange for the interpreter helping indirectly to publicize the Storyspace format in general) or if that company were to contribute to development.

Director

Macromedia's Director format is a mainstay of the electronic arts community and has been a primary tool for electronic literature authors working terrain that overlaps with multimedia-, timeline-, or script-based digital art (as in the case of M.D. Coverley's *The Book of Going Forth by Day*; Stephanie Strickland and Cynthia Lawson's *V: Vniverse*; Realworld Multimedia's *Ceremony of Innocence*; and some of Bill Seaman's works, including *The Exquisite Mechanism of Shivers* and *Passage Sets / One Pulls Pivots At the Tip of the Tongue*). Though Director is currently a live format on Mac and Windows platforms, files created in early versions of the program have already become difficult to use on current operating systems and prospects for future migration are uncertain (especially as Macromedia's Flash software occupies an increasing portion of the territory that was once Director's). A free, open source interpreter for this system would yield benefits in the future, and could also enable access to these works on Linux computers today. However, cooperation from Macromedia would be needed for this task to be tractable (for example, opening the source code for outdated versions of the Director player). While the benefits of a Director interpreter would be great, developing an open source interpreter for a multimedia system, especially one with proprietary multimedia elements and technologies, poses substantial technical challenges.

In addition to Storyspace 1.0 and Director, there are many other candidate systems that the Interpreter Initiative could possibly address at a later date. These systems, which include BTZ (Better Than Zork), HyperCard IIGS, mTropolis, Dynatext, Microsoft Windows Help, Authorware, and Supercard have a lesser priority because they affect fewer works of electronic literature (**Note 5**).

2.2.3 Create Related Services

Besides developing interpreters, a long-term digital preservation initiative can also develop related services to help make the results of preservation available to as wide a circle as possible. For example, a Web site could be created as a one-stop distribution point for open source interpreters and freely available electronic literature works restored by those interpreters. There could also be supporting documents—including X-Literature compatible metadata documents for particular e-lit works [see below on X-Literature], user guides for the interpreters, and teaching or research guides. Participating institutions might receive a periodic newsletter on "What's New in E-Literature Collecting?" together with annual updates of new interpreters, restored works, and so on.

Since these continuing services would extend beyond the time of any initial grant or other funding for the development of the digital preservation initiative, some portion (or level) of services would likely need to generate an income stream to sustain the non-profit effort. For example, the one-stop Web site could be free to all users and institutions. But supporting documents, annual updates, and other value-added services benefiting libraries or classrooms might be sponsored through modest institutional fees or subscriptions.

3 X-Literature Initiative

Obsolescence of electronic literature can be alleviated to some extent through the Interpreter Initiative described above. But it is clear that there are limitations to the purely reactive approach of building interpreters to keep up with the ceaseless mutation of technology. This is because any interpreters (and emulators) will restore to readability only a selected subset of older electronic literature; interpreters do not extend or enhance the usability of e-lit; and interpreters will themselves periodically need to be updated with little expectation of help from a broader or commercial development community.

For these reasons, the fight against electronic literature obsolescence must ultimately occur in a wider framework. Seen in a larger perspective, the problem is not the preservation of old or aging e-lit per se. It is the *description* and *representation* of electronic literature of any vintage in a neutral, open source, standards-based format—one capable of maintaining the essential experience of a work while allowing its presentation to adapt to evolving hardware and software channels through understood, regular, and automated methods of transformation. The problem of preserving electronic literature, in other words, takes its place within the general problem of the platform-neutral representation and transformation of digital media.

> **Metadata** is encoded information about a work that describes its intellectual status (author, copyright, date, terms of use, and other information), physical or digital status (for example, names, locations, and logical relations of files), and potentially also behavior (for example, dynamic or interactive interrelations of a work's elements). . . . (**More**)

Borrowing where possible from open source preservation efforts elsewhere, ELO proposes the creation of an integrated format for the representation and transformation of electronic literature. This format—to be called X-Literature (X-Lit, for short)—involves developing a rich, **XML**-based representation of electronic literature that will be human-readable and machine-playable (as well as machine-transformable) long into the future. Specifically, X-Lit will be a set of open source XML standards, metadata standards, XML applications, and related services designed to augment similar formats in the library or commercial worlds by providing specific extensions and implementations needed to handle electronic literature.

The X-Lit format will allow for the representation of media elements (including text, graphics, sound, and video) and of some interactive or computational effects. It will also provide a way to document the physical setup and material aspects of electronic literature. X-Lit will thus serve as a human- and machine-readable description of electronic literature and of the way the elements in such literature interact and operate. It will provide a uniform way to document works of all sorts so that they can be better managed by authors, publishers, editors, scholars, and others now and *also* be re-created in the future. When fully realized, X-Lit will be an open format that many different kinds of applications can directly play or run, or, at a minimum, export or save to. Indeed, ELO proposes developing a starter set of open source applications that use the X-Literature format—including an X-Lit Reader tool , an X-Lit Migrator tool (for converting electronic literature formats to the X-Literature format), and an X-Lit Muse tool (for authoring in the X-Literature format).

While the central goal of X-Lit is preservation, the ancillary benefits will include a wider dissemination of electronic literature and a broader scope of scholarly and creative activity (in the latter case, for example, through the development of XML or RSS applications that allow authors to include portions of other works dynamically or interactively in their own works).

It is useful to divide the preliminary technical analysis of X-Lit into three portfolios, one devoted to XML and metadata standards, a second to the types of electronic literature that could be represented by such standards, and a third to the e-lit tools that might be built to take advantage of the X-Lit format.

3.1 Technical Analysis of XML and Metadata Standards to Facilitate the Migration of E-Lit

Understanding how to describe and represent electronic literature for the purpose of standards-based migration requires grasping the underlying concepts of XML and metadata. (For the generalist reader, it will be sufficient to understand only the gist of these technologies and to pick up some of their terminology.)

3.1.1 XML (Extensible Markup Language)

XML is a markup language for the logical ("structured") representation of data that inherits much of the combined rigor and extensibility (or the ability to be adapted for various purposes) of its predecessor SGML. However, XML is especially adapted to distributed, networked environments. For example, XML is what allows so-called "Web services" and RSS readers to pull content out of one proprietary database or other application, send it through the Internet, and read or act upon it in another database or application not originally designed to talk to the content-source. (By comparison, HTML is a more limited subset of SGML that is far less robust or extensible and partially sacrifices representing the logical structure of content because it ties content more closely to formatting and display decisions. XML is designed to be a transparent medium between source and target applications, whereas HTML is a partially opaque medium because it is more focused on the browser-rendered experience of the interface medium itself.) Complemented by its various "schemas" (or use-specific vocabularies and grammars of markup tags), XML is rapidly becoming the dominant format for representing any information intended to reside for part of its life cycle on the Internet in a "live" form capable of being received flexibly and not just rendered passively. It has seen extremely widespread adoption in both the non- and for-profit realms, and there are many implementations both open source and proprietary.

(XML itself is an unencumbered format that can be freely and openly implemented.)

XML has a number of advantages as a means of describing and representing works of electronic literature. Especially beneficial is the fact that XML documents can be automatically transformed, processed, and analyzed using readily available methods. For example:

- The XML Style Language (XSL) produces print quality renderings of a work.
- The widely used XSL Transformation Language (XSLT) extracts parts of XML documents and presents (transforms) them in a different format—converting XML, for example, into XHTML for presentation on the Web.
- XML Query is a method for accessing XML documents in a manner comparable to the SQL (Structured Query Language) of relational databases.
- Existing tools to produce concordances, word lists, collocation lists and other analytical devices often either work with XML or can be made to work with intermediate files generated from XML through a fairly simple XSLT transformation.
- As an indication that XML is becoming mainstream: Microsoft made XML central to its Office suite beginning with Office System 2003 (which also supports user-defined XML schemas so that authors are not constrained to vendor-supplied XML tag sets). Office uses XSLT and XML-based Web services, and supports SVG graphics. Mainstream programs from other commercial vendors and open source developers have also moved toward XML native code or XML export/import capability.

XML is not restricted to purely textual information. Graphical information, particularly animations of the kind commonly found in Flash and Director, are addressed by the related Structured Vector Graphics (SVG) format and Synchronized Multimedia Integration Language specification (SMIL, pronounced "smile"). These graphical specifications are increasingly being adopted in mainstream applications. For example, Adobe has provided a freely downloadable SVG plug-in for Microsoft's Internet Explorer, and there are a number of open source SVG implementations, including the open source web browser Mozilla. Real Networks's widely used Real Player supports SMIL.

3.1.2 Metadata Standards (and Archival Reference Models)

Metadata is encoded information about a work that describes its intellectual status (author, copyright, date, terms of use, and other information), physical or digital status (for example, names, locations, and logical relations of files), and potentially also behavior (dynamic or interactive interrelations of a work's elements). When encoded in XML or other text markup schemes, metadata is both human- and machine-readable. METS and RDF are two especially relevant metadata standards from the library and information sciences community that might be extended for use with electronic literature. Governing the flow of metadata among the total network of preservation agencies, repositories, and activities is OAIS, the conceptual framework (or reference model) for archiving.

OAIS (Open Archival Information System)

(For more on OAIS, see **http://ssdoo.gsfc.nasa.gov/nost/isoas/ref_model.html**)

Already widely adopted as a starting point in digital preservation efforts, the Open Archival Information System, or OAIS was originally developed by the space data community but has since added the library, archival, and museum communities to its stakeholder group. Designed as an umbrella framework in which to administer the full range of archival operations, OAIS establishes a functional model for how archival metadata information flows between digital-work producers, archive designers, archive managers, and archive users. In particular, OAIS introduces the idea of "data packages," or integrated packages of metadata information specific to different stages in the archival lifecycle of digital artifacts and different relations between archival agents or institutions. There is the SIP (Submission Information Package), which is negotiated between a producer and OAIS. An AIP (Archival Information Package) is used for preservation, and includes a full set of the metadata and digital media files necessary to preserve the digital object within an archival repository. Finally, a DIP (Dissemination Information Package) is what might be sent to a consumer by the OAIS, and may include part or all of what is in the AIP.

METS (Metadata Encoding and Transmission Standard)

(For more on METS, see **http://www.loc.gov/standards/mets/**)

While OAIS defines a functional model and shared vocabulary for establishing the relations between producers, consumers, and archives, it does not provide an actual implementation model, or specific encoding format used to describe and manage the archival object. METS is a flexible and extensible encoding format capable of storing different aspects of a digital object, and can serve as the instantiated form in which OAIS passes metadata back and forth through the archival system. (SIPs, AIPs, and DIPs can be implemented as METS documents.)

> **Base-64** is a method of encoding binary data in ASCII plain-text form. For example, a binary file can be encoded as plain text for the purpose of transmission through email and then rebuilt in binary form at the other end.

METS is expressed in XML schema language, and provides a means of representing archivally relevant aspects of a digital object (defined here as digital media files plus metadata). The heart of the METS document is an optional file inventory and a structural map. The file inventory is essentially a list of all the digital media files that are included in the digital object. The file inventory can either point to where the files physically reside or provide a location where the files can be Base-64 encoded into the METS document. The structural map (the one thing that is required in a METS document) models how the digital files relate to one another. In addition, there are optional "buckets" for metadata that may be needed in order to interpret or run the digital object. These "buckets" are for descriptive metadata, administrative metadata, and behaviors metadata (as defined below).

- **Descriptive metadata**, or metadata useful for the discovery and identification of a digital object, can either be encoded using an extension schema (such as MARC XML, the Simple Dublin Core XML Schema, and so on), pointed to where it lives natively, or Base-64 encoded into the document. The first two means of expressing descriptive metadata within METS are referred to as "wrapping;" the third method is referred to as "referencing."
- **Administrative metadata** can include four subdivisions: Technical Metadata (information regarding the creation, format, and use characteristics of files); Intellectual Property Rights Metadata (copyright and license information); Source Metadata (descriptive and administrative metadata regarding the analog or other source from which a digital library object derives); and Digital Provenance Metadata (information regarding source/destination relationships between files).
- **Behaviors metadata** can be used to associate executable behaviors with content in the METS object. This is an aspect of METS that a digital preservation project focused specifically on e-lit could develop further.

RDF (Resource Description Framework)

(For more on RDF, see **http://www.w3.org/RDF/**)

As defined on the RDF Web site, RDF is "a framework for metadata; it provides interoperability between applications that exchange machine-understandable information on the Web. RDF emphasizes facilities to enable automated processing of Web resources and as such provides the basic building blocks for supporting the Semantic Web [on the Semantic Web, see **http://www.w3.org/2001/sw/**]. RDF metadata can be used in a variety of application areas—for example: in resource discovery to provide better search engine capabilities; in cataloging for describing the content and content relationships available at a particular Web site, page, or digital library; by intelligent software agents to facilitate knowledge sharing and exchange; in content rating; in describing collections of pages that represent a single logical "document"; for describing intellectual property rights of Web pages, and so on. RDF with digital signatures will be a key element in building the "Web of Trust" for electronic commerce, collaboration, and other applications." RDF is also encoded in XML.

3.1.3 Conclusions of Technical Analysis of XML and Metadata to Facilitate the Migration of E-Lit

Given the momentum behind XML and metadata standards, it will be important for authors, publishers, and archivists of electronic literature to help educate their communities in the most important standards and to adapt those standards for their purposes. But because electronic literature has special properties that distinguish it from much of the digital material that the standards are currently designed to handle, it will also be important for an e-lit preservation initiative (as well as other digital preservation projects dedicated to the arts, for example, Archiving the Avant-Garde; see **Related Initiatives**) to exploit the "extensibility" of the standards—that is, their ability to be implemented in ways specific to particular needs. The X-Lit format will be the extension of XML and metadata standards appropriate for e-lit. In particular, X-Lit can extend existing standards to represent the dynamic and interactive elements that do not figure prominently in static digital artifacts.

3.2 Technical Analysis of Types of Electronic Literature to be Represented in X-Literature Format

Because XML is well suited to document-style data and data structures, the X-Lit format will be able to represent media elements and their interrelationships in many works of electronic literature—especially those with a hypertext-like structure. Often the X-Lit representation of such a work could be rendered with full functionality through XSLT. (For instance, XSLT could transform a link-based hypertext document in an obsolete format into XHTML playable in current browsers.) If some functions of an obsolete hypertext system are not representable in X-Lit, the limitation can be indicated in the output and a supplementary implementation system possibly developed. Alternatively, X-Lit could follow the paradigm of the **METS standard** with its "buckets" for behaviors metadata by encapsulating the code for such functions. Applications capable of doing so could run the code, and other applications would merely treat it as part of the documentation of a work.

But many other works of electronic literature with a more complex computational character (that are primarily computer programs with media embedded in them, rather than the other way around) probably could not be restored to full functionality through just the X-Lit format itself, even with the METS-like encapsulation of code and even though in principle XML and XSLT are by themselves capable of universal computation (as proved by the Turing Machine Markup Language, TMML, which implements a Turing machine through XML and XSLT: **http://www.unidex.com/turing/**). Instead, it would be more realistic in these cases to think of X-Lit as facilitating the development of future **reimplementations**. (While interpreters and emulators may be more tractable options for some e-lit, reimplementations will be useful for important, unusual works; see **Interpreter Initiative** above.) In such a scenario, X-Lit would be used to model just those aspects of a computationally complex work for which XML description is best suited—for example, by encoding textual and other media elements (including lexia in link-based hypertext works with complex embedded behaviors, room descriptions in interactive-fiction-like works, text fragments that generate poems, and so on) together with *only* relatively simple relationships between these elements. Then the X-Lit representation would serve as the "resource fork" or data file for a new implementation. For instance, it would be possible to write a new program that runs such a work as John McDaid's *Uncle Buddy's Phantom Funhouse* or (anticipating a time when it may no longer run) Stuart Moulthrop's *Reagan Library*, which makes use of QuickTime VR, generated text, and a method of keeping state. The new program could use the X-Lit representation of the work's elements rather than the original data files, which would be much more difficult to handle than data in a standard format.

Whether or not a particular obsolete work can be restored to full function from its XML representation, the representation will still serve the purpose of enhancing the activities of archiving, searching, and studying. Such benefits would also accrue to new electronic literature created in conformance to X-Lit. In general, works represented in carefully designed XML are more amenable not just to preservation but to textual and critical analysis, propagation through multiple channels, adaptation to various uses and presentations, and so on.

The possible output from the representation of any work of electronic literature in XML and metadata depends on the type of electronic literature involved. The following is a preliminary analysis of three genuses of e-lit with different technical relations to XML:

3.2.1 Static Works

Static works do not change as a result of the reader's actions, presenting the same options whenever a user arrives at a "screen," for instance, no matter what has been read before. Such works may contain intertextual links (link-based "hypertext"), graphics, and movies or animations initiated when the user presses a button or actuates a link. They do not contain text generated by software in response to interaction. Static works are often produced from older print works, or by authors used to physical media. Examples might include an online version of Martin Gardner's *Annotated Alice*, or a critical edition of a Middle English poem. These works are best represented using the XML HyperText Markup Language (XHTML) in accordance with the markup scheme of the Text Encoding Initiative (TEI).

3.2.2 State-Based Computational Works

State-based works behave differently depending on the path the reader takes to explore them. One example would be Michael Joyce's *afternoon,* which uses "guard fields" to vary the links that are available to a user depending on which lexia have been visited before. Another example would be a simple "adventure" game in which one's character must possess an object in order to solve a puzzle. As an experiment to test the adequacy of XML to the adventure game genre, Liam Quin (a member of the ELO PAD Tech/Software committee) wrote a simple adventure game using XML and RDF to represent state (see **http://www.holoweb.net/~liam/rdfg/rdfg.cgi**). Here, an XML document is processed (via a cgi script) by an RDF engine, though the processing could also have been implemented by XSLT. What makes XML practical for this purpose is that a declarative, descriptive relationship exists between states in the game. A full programming language is not needed.

However, as the relationship between states grows large, this approach becomes less useful. By analogy: it is possible to write a program that tells the user whether an integer between 1 and 10,000 is a prime number simply by listing all 10,000 numbers as "states" that lead to the answer "prime" or "composite" as appropriate. But such would certainly not be a good way to write the program.

3.2.3 More Intensively Computational Works

The full, original experience of works of electronic literature that involve more elaborate computation—whether it is the physics of Jim Andrew's *Arteroids* or the parsing and world-modeling typical of interactive fiction—can currently best be preserved in the same (or equivalent) program rather than by representation in the X-Lit format alone. An example of a work that is more intensively computational can be found at the "random art" page created by Liam Quin titled "Pretentious Yet Pointless" (**http://www.holoweb.net/~liam/sol/**). Here, both the images and text are generated to simulate the work of art criticism. For such works, there are two main approaches possible. The first is to preserve the execution environment, either emulating the original computer system or replacing it with an interpreter. (See **Interpreter Initiative** above). The second approach is to document completely the workings of the program and represent its media elements using X-Lit. Then, the program could be **reimplemented** and the reimplementation would use the X-Lit file as data. Even if no one immediately develops such a reimplementation, the X-Lit format would document the media elements consistently and thus make future study and reimplementation easier.

In the future, of course, an increasing proportion of computationally intensive behavior may be representable in X-Lit. The problem might be visualized on the model of the first transcontinental railway in the U.S., which was built from the West and East simultanteously before joining with the driving of the "golden spike" in 1869. XML has the potential to extend in one direction to represent ever more programming behaviors, rather than simply serving as the container or wrapper for encapsulated programming. (A digital preservation initiative focused on electronic literature could boost such extensions considerably.) Meanwhile, programming environments are moving to meet XML by becoming simpler and more amenable to high-level abstraction (for example, to adapt to XML-based "middleware" or "Web services" connecting proprietary applications through the Internet). As standardization and interoperability proceed from both directions, the golden spike of today's successor to the transcontinental railway—the network—will at some point become conceivable. The golden spike would be a standard that ties XML to programming languages so intimately that X-Lit could become both a representational and programming environment for electronic literature.

Reality will fall likely somewhere between the use of XML just to document computationally intensive behaviors and to implement fully interoperable, high-level programming language. But the goal of a golden spike is worth stating to set the aim for a long-term digital preservation initiative.

3.2.4 Conclusions of Technical Analysis of Types of Electronic Literature to be Represented in X-Literature Format

The potential of XML and metadata is vast because these are the standards that large segments of both the non- and for-profit worlds have settled upon as the technical lingua franca of today's information—the common intermediary language that allows any one body of content locked in one format or program to send a version of itself through the Internet to any other format or program.

But electronic literature is challenging because of the complex nature of its dynamic, interactive, or network-aware presentation. The promise of X-Lit is not that it can provide a working version of every arbitrarily complex e-lit work for all of time. For some works, X-Lit will indeed be able to migrate the original experience to a new cross-platform, open source, and future-friendly format. For others, the gain will be more modest: the facilitation of scholarship and an easing of the task of reimplementation. And some aspects of complex works may not in the near future be preservable at all—just as it is "out of scope" for other media, for example, to preserve not only the image or sound of an amusement arcade but the smell of stale beer and cigarettes.

3.3 Technical Analysis of Electronic Literature Tools for the X-Literature Format

Ultimately, the purpose of X-Lit—like that of other open source, standards-based formats—is to make it possible for a diverse community of future developers to build conformant applications that not only meet the needs of particular audiences (for example, archivists, scholars, authors, publishers) but also improvise upon such needs in ways not predictable in advance. A digital preservation initiative can build a starter set of applications for the X-Lit format designed to enhance the experience of reading, editing, and authoring electronic literature. The following sorts of tools should be developed—though in the short term some will have a higher-priority than others:

3.3.1 Migration Tools: Applications for Migrating Existing Electronic Literature into X-Lit Format (X-Lit Migrator)

Where the source files used by the author of a work are available or the reading files are plain-text and the original format is common, the X-Literature Initiative could develop an X-Lit Migrator application (or set of applications) to facilitate the representation of existing electronic literature in X-Lit format. It seems likely, for example, that some relatively simple formats, such as HTML and Storyspace, may lend themselves to the creation of automated data extraction tools capable of completely or partially converting a work's content into XML that conforms to X-Lit standards for markup, metadata, and transformation into various formats (including, but not limited to, XHTML). (Probably the most efficient method of doing so will be to start in most cases with the files and make a first-pass automatic conversion—as when a word processor makes a conversion from another program's file format. If high fidelity is desired, then hand tweaking will be necessary.) Similar automated migration—but perhaps to a more limited extent (depending on vendor cooperation)—may be possible for more complex formats such as HyperCard and Director or Flash. A small number of migration tools for original formats should take initial focus—for example: for HTML, Flash, Director, HyperCard, Storyspace, and one interactive fiction authoring system (e.g., *Inform*).

More complicated is the case of electronic literature whose original format, though accessible through authoring or plain-text source files, is uncommon (for instance, *Califia*, authored in ToolBook; *Façade*, custom coded). It may not be possible in such circumstances to justify the investment of development resources necessary for automatic or semi-automatic translation. However, it should still be possible to create X-Lit documents that effectively articulate the components of the work (text, code, media elements, file map) and their interrelationship.

Most complicated of all is the case where all that is available are binary files. Migrations of such works into X-Lit format would have to be hand-created by scholars, students, artists, or archivists; and could be accomplished only for the most important works. However, works of this sort can at least be documented (for example, by capturing or transcribing text, taking screen shots, describing operations).

3.3.2 Reading and Editing Tools: Applications for Displaying, Querying, Annotating, Editing, and Teaching Electronic Literature in X-Lit Format (X-Lit Reader)

One of the priorities of the X-Literature Initiative is to support not just the preservation but the dissemination, scholarship, and pedagogy of electronic literature. It is thus desirable to build applications (or extend existing applications) for the X-Lit format that go beyond augmenting the activities of editors/archivists to enhancing those of presenters, scholars, and teachers of e-lit. All these activities can become simultaneously more sophisticated and interoperable by means of established methods of extracting and manipulating XML data (for example, XSLT and XLink; see **explanation of XML above**). Some combination or selection of the following X-Lit applications (referred to generically as an X-Lit Reader) might be built as part of the X-Literature initiative:

Advanced display and reading tools: Such applications would allow a user to "perform" a partial, canned, or otherwise special-purpose rendering of a work of electronic literature represented in the X-Lit format (for example, a selection of elements marked up by the author or scholar as pertinent to a specific theme; a specific sequence of events or images; a map of data elements and their relations).

Annotation and referencing tools: Such tools will probably (but not necessarily) be integrated with the reading or display tools described above. Users should ideally be able to mark discrete or sequential events in a work for study and replay. (Such referencing implemented through the X-Lit format would go a long way toward providing a granular, interoperable, and standardized way of citing electronic literature.) Users should also be able to attach annotations to elements of a work. A related goal is to generate from an X-Lit representation what amounts to a linear annotation of the whole work—for instance, a text print-out akin to a film script that could be used for close study or citation.

Query tools: Query tools would allow users to search electronic literature in advanced ways that have long been possible in structured documents (for example, via SGML readers) but are unavailable in other formats. For example, users might be able to search for all instances of a keyword within a certain kind of data element (e.g., chapter titles or section heading) and then see the results displayed in a variety of ways (for instance, as a visual map, a chart of statistical occurrences, and so on).

3.3.3 Authoring Tools: Applications for Creating Electronic Literature in X-Lit Format (X-Lit Muse)

The development of customized X-Lit authoring applications is possible, but at least initially may be a lesser priority because the level of polish required to create popular authoring tools is very high and there are vigorous commercial competitors who currently own the turf.

However, the X-Literature Initiative can take some steps in the direction of authoring tools. One step is to support the development of tools that extend or build on top of existing authoring tools. A pilot project titled X-Lit Muse, for example, might extend Robert Kendall and Jean-Hugues Réty's **Connection Muse** system, which provides tools for innovative Web authoring. Another pilot project could open the authoring of interactive dramas to many others by developing a version of the infrastructure of Michael Mateas and Andrew Stern's *Façade* (if its authors were willing).

Another step is to work with (or persuade) vendors to build X-Lit conformance into commercial authoring programs (for example, to ensure that the X-Lit format can be exported to or imported from). An argument that might be made to vendors is that conformance to a standard documentation and interoperability format could widen the use of authoring programs in the educational research, classroom, and student communities (the latter a possible sweet spot for vendors).

In addition, the X-Literature Initiative will want to evaluate circumstances after the launch of the X-Lit format to gauge its adoption. Some electronic literature authors may want to author in X-Lit as a native format. At a later date, X-Lit reading, annotation, referencing, and querying tools created by the X-Literature Initiative itself could be built up into a full authoring environment if there were demonstrated demand. Ultimately, the feasibility of developing authoring tools is not a technical issue (since it is entirely possible) but a matter of resource allocation. A digital preservation effort may or may not be funded at a level that allows it to put extensive resources into creating authoring tools as opposed to other tools.

3.3.4 Conclusions of Technical Analysis of Electronic Literature Tools for the X-Literature Format

Creating or extending the standards necessary for the X-Lit format will be an ambitious endeavor. Developing application software to take advantage of the format will add to the difficulty level, since it will require programming amid competition from commercial and other organizations with vaster resources. To demonstrate how the X-Lit format can be useful to electronic literature, however, it will be important for the X-Literature Initiative to develop pilot applications in categories not currently well served by other interests, beginning with migration and reading/editing tools.

3.4 Implementation Plans for X-Literature Initiative

The X-Literature Initiative can be developed in three main stages, with several deliverables at each stage ending in the building of X-Lit tools.

3.4.1 Stage One: Conduct Detailed Technical Studies

The initial stage of the X-Lit Initiative would be devoted to undertaking two detailed technical studies:

One study would create a census and typology of existing electronic literature (building on the ELO's **Electronic Literature Directory**), and then study representative works in depth from a technical perspective. The goal is to produce an enumeration of key technical challenges.

A second study would review existing XML and metadata standards for their usefulness in representing electronic literature. Some issues to be considered are the following:

- How to create limited-fidelity presentations of a work to assist scholarly examination.
- How to formulate reference standards that encompass the citation of specific text within a document.
- How to formulate reference standards able to reflect states of a presentation (for instance, game status in an interactive fiction).
- How to formulate reference standards able to cite a reading of a work, a trail through a link-based hypertext, or a presentation of a state-based work.
- How to create annotation standards that allow commentary and analytical apparatuses to be attached to any of the referenceable objects in a work.

The concrete outcome of these studies would be a set of technical working papers preparing for the creation of detailed X-Lit specifications (for standards, extensions, and applications).

3.4.2 Stage Two: Create XML Schemas

> As defined by the W3C, "**XML Schemas** express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the structure, content and semantics of XML documents." In essence, schemas are a more powerful and flexible way of accomplishing the tasks of SGML DTD's (Document Type Definitions). They allow use communities to extend XML by creating tag sets for specific purposes or kinds of digital artifacts.

Guided by the technical studies outlined above, the X-Literature Initiative would in its second stage create specific XML schemas and metadata standards for electronic literature. These schemas should also accommodate the representation of annotations, thus providing a platform for the scholarship and pedagogy of e-lit.

The design of the XML schemas should encompass some thought about what sorts of interface and interaction are intended. XML markup of phenomena that are interesting but that no conceivable application can use should be avoided. For instance, some presentational details may well need to be dealt with by emulation or simulation only. No practical markup system can capture every phenomenon of potential interest.

The usefulness and robustness of the schemas will be assessed by completely or partially encoding selected works in X-Lit format. The end result will be a suite of schemas in the **Relax NG**, **W3C XML Schema**, or XML DTD languages (in descending order of preference); documentation for those schemas and their intended application; and reports on tests of the schemas upon selected works (**Note 6**).

3.4.3 Stage Three: Create Tools and Associated Services

In a third stage, the X-Literature Initiative would create a set of open source applications that may be either production-quality tools or exemplary prototypes. As concluded above, the highest priority should go to migration and reading/editing tools. Authoring tools have a lower immediate priority. Mission-specific, open source migration and reading/editing tools are not only central to the goal of preserving, archiving, and disseminating electronic literature but are unlikely to be created by the commercial sector. Authoring tools, on the other hand, would be difficult to create at a level of quality that is competitive with tools already in existence, or are likely to be provided by commercial vendors.

Any applications created for X-Lit should be open source. In addition, wherever possible development efforts should try to build on top of existing or ongoing open source development efforts. For example, it should be investigated whether the X-Literature Initiative can use or extend the TidyLib project (**http://tidy.sourceforge.net/**), whose tool for automating the migration of idiosyncratic HTML into conformant HTML might serve as the starting point for an open source HTML-to-XHTML migration tool. *Eclipse* may also be relevant (**http://eclipse.org/**). *Eclipse* is an open source tool platform that has already gained authoring and GUI support, and that currently has plug-ins for many programming languages as well as basic XML tools. Freely available and commercial applications have both been built on top of the *Eclipse* project, including some of IBM's development tools. The X-Literature Initiative could develop new plug-ins to support file formats and authoring functions important to scholars, archivists, and artists of electronic literature.

Besides developing applications, the X-Literature Initiative could develop services that may be offered at no cost to users or by payment or subscription to institutions. Standards and open source applications could be distributed through a Web site, which would serve as a clearinghouse of the latest developments in X-Lit. In addition, applications could be bundled with interpreters, freely-available electronic literature works, and supporting documents as a kind of "starter kit" for institutions participating in the preservation or teaching of electronic literature. And institutions might receive an annual update of new or revised applications. (As in the case of similar services associated with the Interpreter Initiative, some revenue stream will be required because such continuing services intended to spread the results of the preservation effort to as many libraries, scholars, students, and others as possible would extend beyond initial development funding.)

4 Conclusion: Setting a Standard, Sharing the Labor

The long-term preservation of digital works—and especially of complex or experimental e-lit works that test the limits of new media—will require the labor of many stakeholder communities (authors, readers, editors, teachers, publishers, librarians, programmers) that presently do not have excellent means of coordinating with each other. Establishing a framework that can allow for the commitment of time and resources from distributed sources without everyone needing to reinvent the wheel is what the creation of standards—especially open source standards—is all about.

In its role as one of the few organizations representing electronic literature—and the only one focused on the breadth and history of such literature—ELO can initiate the building of such a standards-based framework in alliance with university, library, and other institutions.

Notes

Note 1. In this document "hypertext" is generally used in the limited sense popularized by applications such as HyperCard, Storyspace, and the World Wide Web—that is, to denote media organized in relatively-discrete nodes connected by links. However, it may be noted that in the longer history of new media such a definition was not employed either at the time of the term's coinage (by Theodor Holm Nelson) or by early pioneers of hypertext systems (such as Douglas Engelbart). Nelson defined hypertext as a subset of "hypermedia" (media that "branch or perform on request") and gave both link-based ("discrete hypertext") and level of detail-based ("stretchtext") examples. Engelbart used the term hypertext to refer to all the new document capabilities enabled by the fine-grained addressing of his oN-Line System (NLS). These included linking, but also dynamically-created views at mixed levels of detail, other new modes of navigation, and so on. See Noah Wardrip-Fruin, "What Hypertext Is."

Note 2. As mentioned in the case of the Atari 2600 emulator Stella, an older e-lit work running on a modern computer may not be using the same sort of hardware and controllers. For instance, very early electronic literature experiments were not displayed on computer monitors. Users operated remote print terminals as interfaces instead. Clearly, today's computers will not present exactly the same physical interface as these machines did and, likewise, computers fifty years from now cannot be expected to be like today's machines. However, a version of an old computer program running on a modern computer still provides a much better idea of what interaction was like than does any other sort of documentation.

Note 3. The particular incentive for choosing **open source** methods of building **interpreters** and **emulators** is as follows. Developing a new interpreter or emulator that is not open source may be useful for those who want access to electronic literature today, but it has no value as a preservation technique. A new interpreter or emulator that is proprietary, and for which the **source code** is not available, will be just as hard to deal with in the future as the original proprietary interpreter or computer system is now. Open source software, on the other hand, can be fairly easily **ported** in the future without undertaking elaborate reverse engineering or other new development. Porting will be even more feasible if such software is developed with portability in mind and is well documented. Another preservation effort in the future could undertake a port of an interpreter (or emulator) created today, or the porting could be done by a commercial company, independent scholars, authors, programmers, students, or other enthusiasts. Any single port of such a system—whoever does the porting—will make a whole category of electronic literature available on the target platform. Using a license such as the GNU Public License, a digital preservation initiative could ensure that future ports remain free for everyone, and that they, too, remain open source. Already, the interactive fiction community has access to hundreds of interactive fiction works thanks to free open source interpreters such as Frotz (which implements the Z-machine) that have been ported to numerous different platforms. (Note that for interpreters and emulators to work, the actual works of electronic literature that they access do not need to be open source. The source code for those works does not have to be available at all, and the works themselves do not have to be freely distributed.)

Note 4. Caveat emptor : With regard to systems owned by commercial vendors, there are some circumstances when it will not make sense to proceed with development of preservation systems unless it can be verified that there are a significant number of freely distributed works in the affected format or unless an arrangement can be negotiated with the vendor for free distribution of "obsolete" works (that is, the preservation initiative creates the interpreter and the vendor makes obsolete works available to the electronic literature and scholarly community). This is because while a preservation initiative may not necessarily mind doing work that also indirectly benefits commercial vendors (work that vendors might well be doing themselves to support their products), it should not do so if the lack of freely distributed, older works means that few users in the creative, artistic, scholarly, and other stakeholder communities of electronic literature will benefit.

Note 5.

BTZ (Better Than Zork)

Mindwheel and three other important works (packaged with hardback books and billed as "electronic novels") were created in the BTZ format at Synapse. The rights are owned by Broderbund. There are several options that could lead to wider access to these works. The critical issue is whether Broderbund would permit their free distribution. If free distribution of the works is granted, it may be possible to support the development of a BTZ interpreter by someone in the interactive fiction community at fairly low cost.

HyperCard IIGS

At least one important work, *Théorie des ensembles* by Chris Marker, was created in this system, which emerged in the wake of HyperCard for Mac. Without building a special interpreter, a preservation project could make a difference by supporting development of a free Apple IIGS emulator and by requesting that Apple allow free distribution of the Apple IIGS firmware required for the emulator. For instance, the KEGS Apple IIGS emulator is a free, open source emulator that already exists but has not reached the "release" (1.0) level. Helping this emulator project accommodate works of electronic literature, or making it more accessible to those interested in e-lit, would not be a major undertaking.

DynaText or Microsoft Windows Help

George Landow's "Hypertext in Hypertext" is the most famous work of interest to the electronic literature community published in DynaText. And business hypertext systems (for example, Microsoft Windows Help) have been used to create a few bizarre works of electronic literature (for instance, by Nick Montfort).

Note 6. The Relax NG schema language for XML, which is an ISO standard, can be converted into W3C XML Schema with some subtle differences that affect particular features. Though there is debate about which is preferable, Relax NG has been shown mathematically to be more expressive, and its specification is considerably shorter (and thus easier to learn). The next revision of TEI is using Relax NG as a key component.

Add dates of latest update

Bibliography

B.1 Related Initiatives

no date ✓
- **Archiving the Avant-Garde: Documenting and Preserving Digital / Variable Media Art** UofCA)
  Berkeley Art Museum and Pacific Film Archive, Solomon R. Guggenheim Museum, Rhizome.org, Franklin Furnace Archive, Cleveland Performance Art Festival and Archive
  < **http://www.bampfa.berkeley.edu/about_bampfa/avantgarde.html** >

no date ✓
- **CAMiLEON: Creating Archiving at Michigan and Leeds Emulating the Old On the New**
  U. Michigan and U. Leeds, supported ~~Underwritten~~ by JISC (Joint Info Systems Committee & the NSF
  < **http://www.si.umich.edu/CAMILEON/** >

no date ✓
- **Cedars: Curl Examplars in Digital Archives**
  U. Leeds
  < **http://www.leeds.ac.uk/cedars/** > supported by JISC's eLib and Digital Preservation Focus programs.

✓
- **Task Force on Archiving of Digital Information**
  " Preserving Digital Information: Report of the Task Force on Archiving of Digital Information," 1 May 1996.
  Latest updates

The Commission on Preservation and Access and The Research Libraries Group, Inc.
< **http://www.rlg.org/ArchTF/tfadi.index.htm** >

- **FEDORA: Flexible and Extensible Digital Object and Repository Architecture**
  "The FEDORA Project: An Open-Source Digital Repository Management System" (FEDORA homepage), ~~22 December 2003,~~ U. Virginia Library and Cornell U. Digital Library Research Group.
  < **http://www.fedora.info/** >
  ? ~~supported by the Andrew W. Mellon Found.~~
  Latest update: 6/25/04

- **METS (Metadata Encoding and Transmission Standard)**
  U. S. Library of Congress
  < **http://www.loc.gov/standards/mets/** > Latest update: 10/5/2004

- **OAIS (Open Archival Information System)**
  "ISO Archiving Standards - Reference Model Papers." Curator John Garrett. ~~19 April 2003,~~ NASA/Science Office of Standards and Technology (NOST)
  < **http://ssdoo.gsfc.nasa.gov/nost/isoas/ref_model.html** >
  ?
  Latest update: 7/1/2004. ~~[illegible]~~

- **PANDORA: ~~Archive: Preserving and Accessing Networked Documentary Resources of Australia~~** Australia's Web Archive
  National Library of Australia and Partners.
  no date
  < **http://pandora.nla.gov.au/index.html** >

- Cornell U. **Project Prism: Information Integrity in Distributed Libraries**
  Cornell U. 1 February 2001. – Latest update
  < **http://www.prism.cornell.edu/** >

- **~~RDF (~~Resource Description Framework~~)~~** (RDF)
  Eric Miller, Ralph Swick, and Dan Brickley, "RDF (Resource Description Framework)," (v 1.168, 17 August 2004) W3C ? ← latest update: v. 1.170 10/21/2004 worldwide web consortium
  < **http://www.w3.org/RDF/** >
  Tim Berners-Lee et al., "Frequently Asked Questions About RDF," ~~v 1.26 1 August 2003,~~ ? last update 9/10/04 v. 1.27 ~~[illegible]~~
  < **http://www.w3.org/RDF/FAQ** >

- **Semantic Web**
  Eric Miller et al., "Semantic Web," v 1.206 26 July 2004, W3C Worldwide Web Consortium
  < **http://www.w3.org/2001/sw/** >

- **Supporting Digital Scholarship (SDS)**
  U. Virginia Institute for Advanced Technology in the Humanities and U. Virginia Digital Library Research Group
  no date
  < **http://www.iath.virginia.edu/sds/** >

- **Text Encoding Initiative (TEI)** Last updated 9/6/2003
  TEI Consortium
  < **http://www.tei-c.org/** >

- **Variable Media Network**
  Berkeley Art Museum/Pacific Film Archives, Franklin Furnace, Guggenheim Museum, Daniel Langlois Foundation for Art, Science, and Technology, Cleveland Performance Art Festival + Archives, Rhizome.org, Walker Art Center
  no date
  < **http://variablemedia.net//** >

B.2 Selected Scholarship on Digital Preservation

- ✓ Alain Depocas, Jon Ippolito, and Caitlin Jones, ed. *The Variable Media Approach: Permanence Through Change*. Solomon R. Guggenheim Foundation, New York, and The Daniel Langlois Foundation for Art, Science, and Technology, Montreal, 2003 ✓ (in English and French), pdf ~~online~~ version at Variable Media Network < **http://variablemedia.net/e/preserving/html/var_pub_index.html** >

- ✓ Isaac Dimitrovsky, "Final report, 'Erl-King' project." Variable Media Network. Last update: 1 April 2004. < **http://variablemedia.net/e/seeingdouble/report.html** >

- ✓ "Digital Fever: Archiving Art and Poetry Online". Roundtable. Slought Foundation. 10 April 2003 ← event audio recording. 92 min. real media or windows media < **http://slought.org/content/11144/** >

- Robert Kendall, "The Hypertexts of Yesteryear." *Word Circuits*. 1998. < **http://www.wordcircuits.com/comment/htlit_3.htm** >

- Matthew G. Kirschenbaum, "The Anatomy of a Digital Object." Paper. Conference on *e(X)Literature: Archiving, Preserving and Disseminating Electronic Literature*. University of California, Santa Barbara (April 2003).

- Matthew G. Kirschenbaum, "Editing the Interface: Textual Studies and First Generation Electronic Objects." *Text: An Interdisciplinary Annual of Textual Studies*, 14 (2002): 15-51

- Brian Lavoie and Lorcan Dempsey, "Thirteen Ways of Looking at . . . Digital Preservation." *D-Lib Magazine* (July/August 2004) < **http://www.dlib.org/dlib/july04/lavoie/07lavoie.html** >

- Peter Lyman, "Archiving Digital Cultural Artifacts: Organizing an Agenda for Action." *D-Lib Magazine* (July/August 1998) < **http://www.dlib.org/dlib/july98/07lyman.html** >

- Catherine C. Marshall and Gene Golovchinsky, "Saving Private Hypertext: Requirements and Pragmatic Dimensions for Preservation." In *Proceedings of ACM Hypertext 2004*. Santa Cruz, CA, August 9-13, 2004, pp.130-138 < **http://www.csdl.tamu.edu/~marshall/p102-marshall.pdf** >

- Nick Montfort and Noah Wardrip-Fruin, *Acid-Free Bits*. Electronic Literature Organization. June 2004 < **http://www.eliterature.org/pad/afb.html** >

- PADI: Preserving Access to Digital Information (Subject Gateway to International Digital Preservation Resources), National Library of Australia < **http://www.nla.gov.au/padi/** >

B.3 Works of Electronic Literature Cited

Citations based on those in the Electronic Literature Organization's Electronic Literature Directory, database director, Robert Kendall < **http://www.eliterature.org/dir/** >

- *afternoon, a story*, by Michael Joyce (Eastgate Systems, 1990). Long fiction in English. Hypertext. Published on CD-ROM; for Macintosh and Windows; written in Storyspace.

- *Arteroids*, by Jim Andrews (v. 2.5, Vispo). "Literary computer game for the Web" in English, Portuguese. < **http://vispo.com/arteroids/index.htm** > [not in ELO Directory; can we insert it in the Directory?]
- *The Book of Going Forth by Day*, by M.D. Coverley [Marjorie Luesebrink] (self-published). Long fiction in English. Prominent graphics, hypertext, and other interaction. Published on the Web; plug-ins: Flash; uses DHTML; currently being redeveloped in Director for presentation with the Shockwave plug-in. < **http://califia.hispeed.com/Egypt/** >
- *Califia*, by M.D. Coverley [Marjorie Luesebrink] (Eastgate Systems, June 2000). Long fiction in English. Audio, animated text, prominent graphics, hypertext, and other interaction. Published on CD-ROM; for Windows; written in ToolBook.
- *Ceremony of Innocence*, by Nick Bantock (Real World Multimedia, 1997). Published on CD-ROM; for Windows and Mac. [not in ELO Directory; can we insert it in the Directory?]
- *The Exquisite Mechanism of Shivers*, by Bill Seaman (ZKM, 1990). Published on DVD; developed in Director and QuickTime [not in ELO Directory; can we insert it in the Directory?]
- *Façade, a One-Act Interactive Drama*, by Michael Mateas and Andrew Stern. Custom code (sometimes written in a custom language), audio files, OpenGL, the Java Virtual Machine, and Windows DLLs. (To be released as a free download / CD-Rom in late 2004). [not in ELO Directory; can we insert it in the Directory?]
- "Hypertext in Hypertext," by George Landow. DynaText. [not in ELO Directory; can we insert it in the Directory?]
- *Mindwheel*, by Robert Pinsky (Synapse & Broderbund, 1984). Long fiction in English. Interaction and generated text. Published on floppy disk; for Macintosh and DOS. Includes hardback book.
- *Passage Sets / One Pulls Pivots at the Tip of the Tongue*, by Bill Seaman (1995). Multiple versions, including videotape, installation, interactive installation, audio CD, CD-ROM, Web page. Developed in Director. < **http://digitalmedia.risd.edu/billseaman/poeticTextsPassage.php** > [not in ELO Directory; can we insert it in the Directory?]
- *Reagan Library*, by Stuart Moulthrop (self-published, *Gravitational Intrigue*, the *Little Magazine's* electronic anthology, 1999). Long fiction in English. Prominent graphics, hypertext, and other interaction. Published on the Web and on CD-ROM; plug-ins: QuickTime. < **http://iat.ubalt.edu/moulthrop/hypertexts/rl/** >
- *Théorie des ensembles*, by Chris Marker. [not in ELO Directory; can we insert it in the Directory?]
- *Uncle Buddy's Phantom Funhouse*, by John G. McDaid (Eastgate Systems, May 1993). Long fiction in English. Audio, prominent graphics, hypertext, and other interaction. Published on floppy disk; written in HyperCard. Includes two audiotapes.

- *V: Vniverse*, by Stephanie Strickland (*Iowa Review Web*, September 2002). Credits: Concept and text, Stephanie Strickland; concept development and design, Stephanie Strickland and Cynthia Lawson; programming and reimplementation, Cynthia Lawson; splash page design, Talan Memmott; splash page drawing, Guy Ottewell. Published on the Web; developed in Director for presentation with the Shockwave plug-in. Complemented by book: *V: WaveSon.Nets / Losing L'una* (Penguin, 2002)
  < **http://www.vniverse.com/** >

B.4 Other Resources Cited

- *The Annotated Alice: Alice's Adventures in Wonderland & Through the Looking Glass*. By Lewis Carroll; illustrated by John Tenniel; with an introduction and notes by Martin Gardner. New York: Bramhall House, 1960.
- Electronic Literature Directory. Database director, Robert Kendall. Electronic Literature Organization
  < **http://www.eliterature.org/dir/** >
- Connection Muse homepage. Created by Robert Kendall and Jean-Hugues Réty. 7 April 2003, *Word Circuits*
  < **http://www. wordcircuits.com/connect/** >
- Eclipse.org, Eclipse Foundation
  < **http://eclipse.org/** >
- HTML Tidy Library Project. SourceForge
  < **http://tidy.sourceforge.net/** >
- Liam Quin, "Pretentious Yet Pointless"
  < **http://www.holoweb.net/~liam/sol/** >
- Liam Quin, Simple Adventure Game
  < **http://www. holoweb.net/~liam/rdfg/rdfg.cgi** >
- Relax NG Homepage. James Clark. 24 September 2003
  < **http://www.relaxng.org/** >
- C. M. Sperberg-McQueen and Henry Thompson, "XML Schema." v 1.98. 30. June 2004. W3C
  < **http://www.w3.org/XML/Schema** >
- Storyspace homepage. 2003. Eastgate Systems, Inc.
  < **http://www.eastgate .com/Storyspace.html** >
- Turing Machine Markup Language (TMML) Homepage. Bob Lyons. Unidex Inc. 8 May 2001
  < **http://www.unidex.com/turing/** >
- Noah Wardrip-Fruin, "What Hypertext Is." *Proceedings of the Fifteenth ACM Conference on Hypertext and Hypermedia*. Santa Cruz, CA (2004): 126-27.

The Electronic Literature Organization
http://www.eliterature.org

Colophon · The template for the Web edition of this document was marked up by Nick Montfort in valid XHTML 1.1 with a valid CSS2 style sheet. It is screen-friendly and printer-friendly; a style sheet for printer output is provided which browsers should use automatically when users print the document. To cite a specific part of this document, give the section number (such as 3.2); it's also possible to link to specific parts of this document by using the links at the top, under the heading **"Contents."** ¶ The authors of *Born-Again Bits* thank the other members of the ELO board of directors for their numerous, detailed corrections and suggestions for revisions. ¶

CREDITS:
The festival co-organizers are Robert Nideffer, Antoinette LaFarge, and Celia Pearce. Nideffer and LaFarge are both UC Irvine Associate Professors of Digital Media and were the co-curators of the extremely successful 2000 exhibition SHIFT+CTRL that brought together a selection of contemporary game art for the first time in a U.S. exhibition. Pearce was responsible for *Entertainment in the Interactive Age*, a 2001 conference on game design and culture at USC.

ALT + CTRL was juried by Nideffer, LaFarge, Pearce, and an outside panel of jurors from the independent game scene, including Rebecca Cannon, Drew Davidson, Erkki Huhtamo, Paul Marino (who led the jurying for the machinima works), Jeannie Novak, Warren Spector, and Eric Zimmerman.

ALT + CTRL is a joint project of the Game Culture & Technology Lab, The Beall Center for Art and Technology, and
Cal-(IT)2, the California Institute for Telecommunications and Information Technology, University of California, Irvine.

In addition to our corporate sponsors, ALT+CTRL has also received generous support in the form of public relations assistance from the Entertainment Software Association (formerly the IDSA), producers of E3, the National Endowment for the Arts, and the Department of Studio Art, UC Irvine.

ADDRESS:
University of California, Irvine
712 Arts Plaza, Claire Trevor School of the Arts
Irvine, CA 92697-2775

HOURS:
Tuesday – Wednesday, 12:00-5:00
Thursday – Saturday, 12:00 – 8:00

CONTACT / INFO: (949) 824-4339
http://beallcenter.uci.edu

BeallCenter mailing list
BeallCenter@uci.edu
https://maillists.uci.edu/mailman/listinfo/beallcenter