

Managing an e-Pub: Workflow, Version Control, and Rights

Amy Friedlander, Ph.D.
Associate Director for Research
Center for Information Strategy and Policy
Science Applications International Corporation
McLean, Virginia

The principal impediments to scaling up at **D-Lib Magazine** – and, I suspect, elsewhere – are workflow and version control. Neither appears wholly amenable to technological solution. Rather, the technological tools must be embedded in appropriate organizational environments, procedures, and practices. This is very hard – and made all the more difficult because there are critical components in the production cycle over which the staff may have little control.

But first, a word of background on **D-Lib Magazine**. The magazine was begun in the spring of 1995 as part of a program at the Corporation for National Research Initiatives (CNRI), sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA) to provide technical and management support to the then-recently funded Digital Libraries Initiative (DLI).[1] The magazine's first issue was released on July 15, 1995. Initially, it was conceived of as a simple information vehicle principally for the six DLI projects. These university-led cooperative efforts involved multiple partners, and the magazine was intended to offer a forum for early discussion of research results from the DLI partnerships as well as a mechanism for identifying other projects, outside of the DLI, that had yielded or were pursuing congruent or complementary goals.

We were astonished at richness of the research environment – at the range of subject domains, from music and literature to astronomy and medicine, and at the global reach of interest. By the summer of 1998, we estimated that we had 4,000-4,500 readers. Part of this interest arose from the international character of the computer science researchers, which itself arose, at least partially, from its origins in networking and interoperable systems. For example, one of the earliest stories to come to **D-Lib** from the readership (rather than our soliciting the piece from the researchers) was Renato Iannella's discussion of TURNIP, which appeared in the March 1996 issue of the magazine.[2] TURNIP addressed the problem of URN interoperability, then a "hot" issue in the computer science research community. Over time, **D-Lib** came to play an important role in the dissemination of information in other key areas, notably, metadata. We were also early in calling attention to the importance of multi-linguality. Finally, we always understood that the scope of digital libraries research was broad: from near-term projects that were explicitly library-based – the work at the U.S. Library of Congress and the U.S. National Agricultural Library, for example, and many of the U.K. eLib projects – to advanced research projects that called on basic research in computer science (e.g., content analysis of images, perception recognition, and so on).

So, **D-Lib Magazine** grew in response to interest from two major sources: from writers who were willing to share their early and sometimes speculative research results, and from readers who wanted to read. This meant that the stories became longer and more sophisticated, and the magazine became larger. At the same time, web users, in general, became more sophisticated, implicitly demanding better

displays and improved design. For-example, because **D-Lib** maintains its back issues as part of the site. As the backlog of stories grew, more and more individual stories were picked up by such services as **Current Cites**[3], the **Scout Report**[4], and Charles Bailey's **Scholarly Electronic Publishing Bibliography**[5], and more and more readers entered **D-Lib** via individual stories (two levels down) rather than via the home page. We had initially conceived of the stories as having been accessed through first the home page and then the contents page of the magazine. Much of the functionality of the site (e.g., searching, indexes to back issues) was identified higher up and was, therefore, not obvious to readers who first encountered the site via the stories. This meant that these readers, who had been directed to the magazine's site via individual stories from other services, were unaware of "on-site" services available to them.

I first realized this when I began to receive messages asking why **D-Lib** did not maintain an archive of back stories – when I knew perfectly well that we did. Once we realized the behavior, we, then, re-designed the site, adding more links from the stories to other services within the magazine so that the stories became both more and less dependent on the structure of the site. From the readers' point of view, more functionality was available from the stories; searching, for example, was now one click, not two, away. But this meant more formatting, more editing, and more management for **D-Lib's** staff and more work during the two-week production cycle that precedes each issue.

D-Lib's production cycle.

Each issue of **D-Lib Magazine** consists of material that CNRI prepares itself (the editorial, clips and pointers, access pages, and other boiler plate material) and material that others contribute: guest editorials, stories, and notices for *Clips & Pointers*. The stories are the heart of the magazine's content – although we believe that the *Clips* column is more heavily read.

From a management perspective, putting **D-Lib** out once a month is a classic peaks and valleys problem. That is, many tasks, many of which are contingent on one another, are crowded into a short span of chronological time – the last few days before the release of the monthly issue on (or near) the 15th of the month. On the other hand, at least two weeks of every month – the two weeks not devoted to assembling the magazine – are relatively peaceful, and the editorial staff can work on other projects or can undertake long-term maintenance projects. During this "valley", tasks are also undertaken that anticipate and otherwise reduce the "peak" period of the magazine's production cycle.

Managing content may be divided into three major phases: acquisition, production, distribution. (This is intentionally based on a print model.) Story acquisition may begin several months before the month in which a story may run, and we try to have stories in the pipeline three months in advance of release. But this rule is not hard-and-fast. Depending on flow, topic, and "newsworthiness", we added a story or clip at the last minute. Or, at the request of an author, we delayed a story a month or two. As of September 1998, last minute additions were restricted to material in the *Clips* column. This decision resulted from tasks associated with the production phase – the activities that dominate the two-week period preceding release, which always takes place on the 15th of the month or the nearest business day.

Assuming that a story is submitted on day 1 of the production cycle (wherein day 15 is the day of release), the story is subjected to the following steps. It is first reviewed for content. Since **D-Lib**

Magazine is not peer reviewed, review is done by the editor and consists primarily of reading for flow, content, and obvious inaccuracies (e.g., 1997 instead of 1998, NFS instead of NSF, etc.). The reason for editing is to help the writer communicate with the reader. So, the content of the revision is less important than that writer be encouraged to revisit possibly rough patches in the text and to help her or him see the text the way a relatively uninitiated reader might see it. My own experience in having *been* edited is that the editor's comment more often than not identified a problem. But the content of the suggested fix was usually not to my liking, and I frequently found that I needed to re-organize the material or to set the argument up earlier so that the problem was pre-empted.

So, if there were odd wordings or content that did not seem to make sense in a story, I would consult my colleagues for advice, occasionally by e-mail. I, then, sent comments back to the writer. These were always considered purely advisory; in some cases, writers telephoned, and we walked through the solutions to problems in wording. Otherwise, writers are always encouraged to use what seems sensible and to ignore the rest.

The steps described up to this point – story acquisition and review of content – are known as "developmental" or "substantive" editing in the print world, which is the source of this model. There is also a level of copyediting, which focuses on spelling, punctuation, grammar, and usage. Both the developmental and copyediting become merged in the electronic world with the question of formatting. It is an example of the way in which tasks that are discretely defined in print become re-organized and fused in the electronic desktop. It is also a source of problems in workflow and version control.

Let us step back and review the principal steps as we understand them from the print world:

1. Stories are commissioned and received from authors.
2. Stories are subjected to substantive or developmental editing. Comments are returned to the author, who responds. This process is iterated as many times as necessary until the content is agreed-upon.
3. Stories are sent to copyediting for intensive review of grammar, spelling, and diction.
4. In many journals and magazine, stories are also subject to fact-checking (not done at **D-Lib**).
5. Marked-up copy may or may not be sent back to the author for final approval (practice varies).
6. Copy is converted to electronic galleys.
7. Galleys are pasted-up, and final cutting may be done for space reasons.
8. Page proofs are created and proofread.
9. The "document" is printed and distributed.

Two differences between the electronic and print worlds leap out: One, we are not necessarily constrained by real estate in the electronic world, so cutting content purely to achieve a specified layout is not necessary. Two, the distinction between editing and mark-up fuses, although the distinction between developmental and copyediting remains meaningful. As a practical matter, for the first two or so years, I typically did the substantive editing, mark-up, and first round of copyediting as a single episode. Only

after a copyeditor joined the staff did we add copyediting as a second step – which greatly improved the product.

Fusing editing (a content function) with mark-up (a production function) is one example of how the electronic tools and the web environment modified conventional workflow. However, since web authoring tools are widely distributed, you might well ask the question, why not let the authors do the mark-up? Well, in the beginning we did. But over time, we found that the tools, in fact, got in the way. Many were just too user friendly. Moreover, as our editing and as web tools both became more sophisticated, opting for web functionality, which can be limited, restricted our ability to work with the authors.

In our experience, many of the HTML editors either did not read files from different applications or over-wrote the tags, resulting in distorted displays. A story created in LaTeX on a powerful workstation and converted to HTML might look quite different when it is brought into an application running on a PC. Some applications interchanged `<blockquote>` with `<div>`, for example, and this affected paragraphing. Moreover, we found that sending files through some of the e-mail clients also introduced modifications. We also knew that our international readership might not be using the most sophisticated versions of the browsers, which meant that the tagging had to be absolutely clean and consistent within the site. `<Blockquote>` had to be `<blockquote>` (and not `<div>`) every time, so that the display would be consistent in whatever browser a user might have. As our internal display became more sophisticated – we re-designed three times in three years – our own display began to encounter problems, even when skilled HTML authors attempted to copy it. All and all, this meant a lot of cleaning up.

In addition, there were two tags that we wanted to see *added* to the header: The tag, `<META HTTP-EQUIV="content-type" CONTENT="text/html; charset=iso-8859-1">`, controls for font settings in browsers. It specifies ascii. If a browser defaults to ascii, this tag is redundant, and, in some older browser, is known to impede performance slightly. Indeed, some users complained when we introduced it. However, research in digital libraries is an international phenomenon, and **D-Lib**, indeed, is read globally. We believe that there is a very strong readership in Japan, China, and Thailand, where the native scripts are not Roman and the expectation that browsers default to ascii is, thus, unfounded.

The second issue concerned the title field. **D-Lib Magazine** shows the title of a story in at least three places: on the monthly contents page, as the title displayed on the first page of a story, and in the title field in the header (`<title></title>`). The first two places are thought of as the "display title", the third as the HTML title. The Excite search engine, which is used to augment conventional author and titles indexes (or lists), displays the information in the title field as the "title" of the story in the list it returns in response to a user's query. To reduce ambiguity for the reader, we decided that the HTML title and the display title had to map, and when we deployed the Excite search engine, back story files were manually inspected and corrected. An HTML editor will not necessarily recognize the display title as the HTML title, nor do authors either remember to provide information in this field or to use the identical wording. (I received countless files called "dlib_story" or some variant of the phrase.) In principle, reconciling all three instances of the notion of a title so that they were consistent could – and should be – automated. However, as the mark-up changed – the way, for example, we identified main titles and subtitles – it became more difficult to come up with a simple script that would extract the title from the display.

This fine-grained level of detail argued for close editing and ultimately for final editing at the text level. For working with the authors, many of whom were at least as proficient as we in the ways of HTML,

though, we learned that conventional word processing programs had advanced editorial functionalities, like Word's "track changes", that let us edit more freely and communicate our suggestions to them. So, word processing applications were, in fact, better for substantive editing. This, combined with issues of interoperability among HTML applications, led us to encourage authors to submit stories either as a Word file (*.doc) for the first draft, or as a simple HTML file, which we sometimes converted to a Word file, if a lot of editing seemed necessary, or which we converted to a text file for editing and corrected mark-up. Documents that were edited in Word were converted into HTML after the content was stable, but we always did the final editing at the text level – on the server in vi, if necessary, to make sure that the last stray characters were removed.

Before official release of the magazine, we put the finished stories in a public but undisclosed location to enable the authors to them in their final form. This step is equivalent to galleys, or blue lines, in the print world, and offered the authors a last opportunity to make silent changes. This is important for two reasons: First, the authors retain copyright to the material. There is, thus, an obligation to make a reasonable effort to let the authors see the versions of their work that the public would see. Second, after **D-Lib** is officially released, no changes to material in the magazine *are* made that cannot be clearly identified. Any changes that are made (for example, if we corrected transposed images or offered a version of a piece in a second language.) are noted at the end of the file, so that readers can have confidence that what they read today is what they read yesterday. Or they know explicitly how the document has been changed. This is a policy of version control through scheduling.

At first, authors routinely reminded us that the technology made it easy to change the stories and a few resisted the scheduling control. Over time, though, both authors and readers have told us that they *like* the policy. Like the rigorous attention to the schedule of monthly releases – the 15th or the nearest business day – it conveyed reliability and with reliability, confidence.

Four hands make accidental overwriting easy.

Once stories had arrived for a given issue and we had compiled the sites that would be described in the *Clips* column, we constructed the Table of Contents (TOC). The previous month's contents file became the template for the new month, unless we had redesigned the site. The TOC became the controlling document for the forthcoming issue. The TOC also forms the basis for organizing the electronic and print documentation (e-mail, correspondence, etc. are organized by *story*.), so the TOC, in fact, has become the key organizational document for the project as well as the controlling document for each issue.

From a systems point of view, **D-Lib Magazine** sat on a simple UNIX file structure of sub-directories and files. Each issue of the magazine was assigned a directory, typically named month-year (e.g., september97, october97, december97, and so on). Within each issue, stories that comprised one file were assigned a single file, and stories that comprised more than one file were given a sub-directory. The contents page was always month-contents (e.g., 10contents.html for the contents page in the October issue); the editorial was always month-editorial (e.g., 10editorial.html for the October editorial); and clips was always month-clips (10clips.html). A story by Peter Jones in the October issue that consisted of a simple text file would be named 10jones.html, but a story by Ann Jones that consisted of a text file and several

images would be organized as (jones/10jones.html and so on). So, this fictive October issue might look like this:

```
~/october99
  10contents.html
  10editorial.html
  10clips.html
  jones/
    10jones.html
    image1.gif
    image2.jpg
```

Names were mnemonic; that is, we gave objects names that made it easier for the editor to keep track of the content, not names that were semantic. In addition, we assigned persistent identifiers, first handles and subsequently Digital Object Identifiers (DOIs)[6]. But neither was initially intrinsic to the way we managed the magazine, either the content or the underlying system.

The impetus to change came from growth. The file names became part of the URLs, and when I built TOC each month, I established relative links because it was – and is – good web design. We built and tested the magazine on a set of private directories that duplicated the file structure on the publicly-accessible space. These two spaces came to be known as the private and public sides of **D-Lib**, and release each month meant shifting the completed files from the private directories to the public ones.

As each story changed, I over-wrote the previous version with the new one, keeping back versions either on my PC or in yet another set of directories on the server. Very quickly, I learned that I needed controls in the files themselves to check that the correct file had, indeed, been loaded. So, I began to experiment with various header tags and notes, rather like the commentary that software developers use, so that the files contain language that let us maintain control of the process but would not necessarily be displayed in the browser. (You can see this for yourselves; use the "view source" command on your browser!)

Or so I thought. While I was doing the magazine alone for the first two years, the system more or less worked. But as soon as I started to share the work with a copyeditor, the incidence of erroneous over-writes multiplied – an experience not unknown to software developers. We tried controlling through scheduling, which meant that at a certain point, I handed stories off to the copyeditor and agreed no longer to touch them. But this failed under a time crunch, which routinely happened as we tried to coordinate the writing schedules of busy authors who made last-minute changes. We also considered changing permissions so that only one person had access, but rejected this idea because of concern for single point failure – not just during the production cycle but also for routine maintenance on other parts of the site. What would we do in an emergency, for example, if the copyeditor got stuck in traffic, a common occurrence in Northern Virginia? And what about vacations? Yes, we could change permissions, but that seemed to create work rather than solve the problem.

When I left the magazine after the release of the October issue, we had begun to experiment with the notion of using persistent identifiers as a way to address version control while not substantially modifying the workflow. This was based partially on other CNRI experience with DOIs as a way of

building web sites. Philosophically, **D-Lib** had committed to the importance of persistent, unique, location-independent identifiers from the outset in 1995. Although we assigned and registered handles, CNRI's implementation of this technology[7]; we did not use them in the operation of the magazine. Our reasoning was that since **D-Lib** was primarily an information vehicle, and since we wanted to reach the broadest possible audience, we would use the ubiquitous technology – that is, URLs. Moreover, our use of handles was not restrictive; we recognized that there were several naming proposals and would have assigned and registered different names at the request of an author. Stuart Weibel's stories, for example, have been assigned PURLs and are accessible through that system.

By the summer of 1998, CNRI staff had begun to experiment with proxy handles, so that users need not necessarily download the client software to read complex documents composed of objects internally identified with persistent identifiers, that is, DOIs. There were some questions about performance, since it meant two jumps to resolve a DOI rather than one. Equally importantly, there was a question in my mind about the value of the identifier if it meant extra work for the editorial staff. We were now talking about registering identifiers for drafts – an extra step – in order to compromise performance for the reader. This did not seem a reasonable trade-off.

Why do this at all? One of the arguments on behalf of unique, persistent identifiers is that, unlike URLs, identifiers remain stable, even if the location of the stored content changes. But **D-Lib** is a carefully managed site. We did not change URLs, and we did make back issues available, so persistent identifiers seemed equivalent to URLs – at least in our case. It was explained to me that the value of persistent names from a management perspective came into its own when the content of some files changed, and that it was, therefore, useful to think of the identifier as a variable rather than an absolute value. This meant that we could abstract the persistent *identifier* of the story from the *content* of the story and could use an internal registration process as a means of tracking the versions of the content as it changed.

From **D-Lib**'s production perspective, we could still build the contents page, this time with identifiers rather than relative URLs, and use it as an overall control mechanism (see Figure 1). The identifier would point to a URL, which we would update as the versions of the content succeeded each other. Therefore, we would never overwrite files. Rather, we would update the pointers to identify the most recent version of a given story, and merely by looking at the succession of files, the editor and the copyeditor could tell who was working on the most version of a story. The file or object structure of each issue was thus abstracted out of the contents page and expressed as an intermediate step. And the trade-off was justified; the extra step became a management tool that might enable us to eliminate erroneous overwriting, maintain adequate version control, and reduce bottlenecks. More than one person could still work on the magazine at the same time.

When I left **D-Lib**, this approach was still at the idea stage. As a solution to workflow and version control issues, it seems promising. Concerns had arisen, though, over additional burdens this might place on **D-Lib**'s mirror sites and possible compromises in performance, particularly for users overseas with limited connectivity or where congestion tended to be severe.

Authors' rights, readers' access, and polyhierarchy

One advantage to the system of directories and sub-directories was that it mapped onto authors' rights. The author either held the copyright to the material in a clearly identified file or in an equally clearly

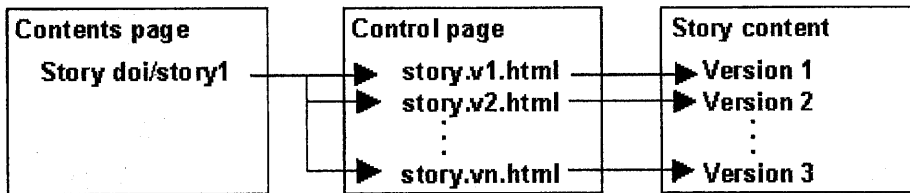


Figure 1. Version control using persistent identifiers
(Illustration, courtesy of Catherine Rey.)

identified sub-directory within a given month. This advantage migrated to the persistent naming approach, since it meant that the identifier pointed to the conceptual object – the story – unlike the URL, which only identified the *file*. [8] When the file and the story were identical, this was not an issue. But when a story consisted of more than one file, then the identifier applied to the conceptual entity and not merely one of a collection of constituent objects. This appears likely to become increasingly useful, not only for version control, as previously described, but also for rights management.

D-Lib embraces a two-step rights policy. At the site level, we made material available essentially “as is” and did not guarantee its accuracy. This meant that we did not do fact-checking in the conventional sense and also mandated close coordination with the authors, who approved all but the most trivial changes. This meant that at the copyediting stage, we still sent changes to authors for approval since changes in punctuation might imply changes in meaning. (Consider relative clauses. A non-defining relative clause is always set off by a comma and takes the relative pronoun, “which”; a defining relative clause may take a “that” or a “which” and is not set off by commas.)

At the story level, all rights remained with the authors or with their designated rights holders. We requested non-exclusive permission to make the stories available through **D-Lib**. I had initially seen this policy as a means of creating an incentive to authors to write for us, since most scholarly journals requested – or required – acquisition of copyright. However, over time, it was clear that this policy reduced our overhead. I did not have to manage the acquisition or track permissions. When permissions came in, we referred the request directly to the authors.

Over time, this policy gently evolved to include a request for novel submissions, meaning that the submission was new with us, although we did not restrict re-use after we did the release. Indeed, we encouraged authors to seek other venues but requested notification, as a courtesy. This became a means of tracking **D-Lib**’s progress since we could collect, at least informally, information about where **D-Lib**’s stories were appearing and the increasing level of acceptance that the magazine came to enjoy in the professional communities to which it is directed. [9]

We also extended the request to include the right to allow stories in **D-Lib** to be covered by abstracting and indexing services and by other digital libraries. Notable in this regard was the relationship with the Networked Computer Science Technical Reports Library (NCSTRL). [10] **D-Lib**’s research stories became part of the NCSTRL distributed collection in September 1998. In cooperation with researchers from Cornell University and the University of Virginia, we began to study the relationship the previous spring, and it turned out to be somewhat more challenging than we had expected.

The NCSTRL framework is available in two ways, as either Standard or Lite. Either way, it provides **D-Lib** potential access to an important audience of both potential authors and readers; thus, **D-Lib** places importance on NCSTRL as an access mechanism similar to abstracting and indexing services, inclusion in subject bibliographies, and functionally similar, analog library-based finding aids. We chose the NCSTRL-Lite (NCSTRL-L) approach rather than the NCSTRL-Standard (NCSTRL-S) approach because from our perspective, this is simpler and requires less local maintenance. Lite "expects" **D-Lib** to make bibliographic records available in a public location where the records can be polled periodically. The content of these records conforms more or less to RFC 1806 and represents a minimal element set that could potentially migrate to Dublin Core or one of the other evolving metadata standards.

Students at the University of Virginia graciously wrote a script that extracted the requested record from existing **D-Lib** material. Because we had modified the display several times, the results of this script were good but not infallible. Several rounds of proofreading and editing were required until the "bib file" was considered clean. In fact, this was a very useful exercise, because a number of typographical errors and inconsistencies were surfaced. (Trailing spaces was the principal offender.)

Several areas of concern arose of which the most significant – and interesting – was the display. In the reference implementation then hosted by Cornell, the display provided the user a uniform interface to the underlying, disparate reports, masking their physical location and reducing differences in display. Because the implementation relied on a sub-window in frames, the reader was unable to bookmark the document, which is irrelevant if the user has chosen to download and save a *.pdf or a *.ps file. By definition, the user has a copy on her/his machine for future use. Moreover, displaying the object as a sub-window within a frame preserves a sense of context considered analogous to reading a book or an article in a physical, analogue library.

Objects in HTML, which may have been created for use on the web are slightly different. **D-Lib** stories have been written with the expectation that they will be displayed within a conventional web browser and **D-Lib** has expended substantial effort on developing a consistent, functionally appropriate, and attractive display. This is the display, moreover, that the authors have approved. Moreover, in a digital world, this is the display that becomes the authoritative version of the story. (Recall that in print, authors are given a certain number of copies of the published article; in the on-line world, the authors must rely on the archived version as the authoritative instantiation.)

Display within a framed sub-window compromises that effort and reduces the "readability" of the content. Moreover, the ability to bookmark the item directly has been removed by virtue of the sub-window and there is some question as to how the service request would be reported in the server log. This matters to **D-Lib** because we used the server logs – inherent problems notwithstanding – as a means of understanding traffic and as a rough, first approximation on usage and circulation.

We understood, however, that there are several issues here. One is the cognitive transition that a reader makes from the global to the local environment. This has been described as analogous to the transition a reader makes from the sense of being in the physical library to the sense of being in the world that the writer creates with no small assistance from the publisher, book designer, printer, and so on. At one point, this reasoning goes, does the digital world also make this transition and how is this represented in the display and, if necessary, in the underlying architecture?

From **D-Lib**'s point of view, one of the values of NCSTRL is that stories in **D-Lib** become accessible

in the way that they are accessible through an A/I service such as PAIS, which already covers the magazine. For reasons related to readability, context, and identity, sub-ordination within a sub-window was not an acceptable alternative. We preferred to focus the discussion on the notion of the reader's desktop, and researchers at Cornell graciously agreed to modify the interface so that a second window, though independent, would appear as a cascaded window on the user's desktop. Thus, a user might have three or four windows open simultaneously, including the NCSTRL page that returned the list of "hits". Manoeuvring among several objects is a question of point-and-click rather than backing up. From the individual stories, readers could access **D-Lib's** back issues, indexes to titles and authors, and the Excite search engine that covers the site.

On balance, we believe that both **D-Lib's** authors and readers are served through including **D-Lib** in the NCSTRL federation. However, it has imposed a burden on the magazine. Stories are now explicitly more de-contextualized than before. While it is clearly true that individual stories were recognized by their URLs in other services and bibliographies, we have now agreed to provide a mechanism for accessing all of the stories as a collection without going through either the home page or the contents page. From a rights perspective, it means that from the *stories*, there has historically been no way for readers to know what **D-Lib's** access terms and conditions are. We had assumed that most readers would access the stories via the contents page, and a pointer to the access policy is identified there. Only a copyright notice was routinely added to the story. This was changed in October and a hyperlink from each story to the access terms and conditions. Internally, **D-Lib's** structure appears hierarchical. For readers who may arrive at stories from NCSTRL – or some other service – the structure is polyhierarchical with each story, rather than the contents page, as the apex.

Conclusion: Whither D-Lib?

I left my position as editor of **D-Lib Magazine** on October 21, 1998. It was a sad day. My reasons were many and, for the most part, private. But I believed that for **D-Lib** to grow, it needed new ideas. With a new grant starting, this was an opportune time to make the change. In the first three years, I believe we showed the value of running a project, which had independent value as a content vehicle but was simultaneously a means of surfacing technical issues. Version control and workflow had emerged as important considerations that were organizational rather than technological although the technological tools contributed to shape and form of the solution. Ultimately, it is the distinction between "name" and "content" that enables us to envision identifiers and the process of registering them as part of version control – but that tool is embedded in a variant of the traditional publisher's workflow of acquisition, editing, and production. Persistent identifiers are also valuable from the point of view of rights management; it potentially enables us to use single identifier for a cluster of digital objects, which *collectively* belong to a given author. Finally, attention to their writers – to their schedules, to their work, and to their intellectual property rights – proved to be a value that was threaded throughout the organizational and technical systems we evolved.

Workflow is a theme of the new research and proposed activities include metadata, copyright submission, and archiving. A year from now, I expect to visit the magazine and find it transformed in exciting ways. But let me leave you with a story that convinced me it was time to move on.

During the second European Conference on Digital Libraries in Crete in September 1998, I was listening

to an informal conversation between two colleagues; both had written for the magazine. Three of us were sitting around a table. One, a very well-known researcher in digital libraries, insisted that **D-Lib** is a journal of record, equivalent in substance, if not in technical fact, to a peer-reviewed journal. It is a point of view others have expressed. No, insisted the other, who is also active in the field, **D-Lib** is a *magazine*, definitely informative, but not juried.

Neither of them asked me.

I was delighted. Three years ago, we had set out to create something for the community, and the community had clearly made the periodical its own.

[1] Until July 16, 1998, **D-Lib Magazine** [<http://www.dlib.org>] was funded under Grant No. MDA972-95-1-0003. Thereafter, it was sponsored under Grant No. N66001-98-1-8908.

[2] Renato Iannella, TURNIP: The URN Interoperability Project, **D-Lib Magazine** (March 1996), [<http://www.dlib.org/dlib/march96/briefings/03turnip.html>].

[3] **Current Cites**, [<http://sunsite.berkeley.edu/CurrentCites/>].

[4] **The Internet Scout Project**, [<http://scout.cs.wisc.edu/scout/>].

[5] Charles Bailey, **Scholarly Electronic Publishing Bibliography** (version 21: 9/18/98), [<http://info.lib.uh.edu/sepb/sepb.html>].

[6] DOI: The Digital Object Identifier System, [<http://www.doi.org>].

[7] The Handle System, [<http://www.handle.net>].

[8] This point has been elaborated in Amy Friedlander, *D-Lib Magazine*: Publishing as the Honest Broker. In **E-Serials: Publishers, Libraries, Users, and Standards**, pp. 1-20, Wayne Jones, ed. (New York: The Haworth Press, 1998).

[9] By the time I left in October 1998, there had been one request to translate William Graves's All Packets Should Not Be Created Equal: The Internet2 Project (April 1998) into Spanish. In addition, references to **D-Lib Magazine** had appeared in **Scientific American** and in **Communications of the ACM**. We had also been told that citations to **D-Lib** stories were appearing in technical papers submitted for review in refereed journals and in the resumes of candidates for employment, promotion, and tenure.

[10] Networked Computer Science Technical Reference Library (NCSTRL), [<http://www.ncstrl.org>]