From Prototype To Production In Hypermedia Projects

Robert J. Glushko
Hypermedia Engineering
Alpharetta, GA 30202

Abstract

Many hypermedia projects exist as small-scale prototypes or so-called proof of concept demonstrations. But full-scale projects that are deployed and maintained in production are scarce, because problems can arise at various points in the project life cycle that cause the project to fail. Problems that arise when the project is started include the lack of realistic expectations and the difficulty of assembling a multidisciplinary project team. Problems that emerge during the design and development phases include few published case studies or design guidelines, poor quality or availability of source information, inappropriate and underpowered software technology, and uncertainty about intellectual property issues. Problems at the deployment and maintenance phases include installed base constraints and inadequate methods for maintaining the hypermedia system. Not all of these problems are specific to hypermedia projects, but they conspire with the novelty and immaturity of hypermedia to make hypermedia applications hard to design, develop, and deploy successfully.

Introduction: A Composite Case Study

Attracted by the excitement about hypermedia in the popular press and technical community, an organization decides that links, navigation features, and multimedia will bring enhanced usability to an information management problem that has traditionally been handled in a database or document archive. Since it is the first hypermedia project for the organization, start-up funds are easy to come by. All of the most talented and ambitious researchers and developers in the organization find their way onto the project.

No one keeps track of how much time and effort goes into it, but after a few months a carefully hand-crafted demonstration system or prototype emerges using "Hyper-X". Hyper-X is a highly-touted program that everyone is talking and reading about as a revolutionary software advance.

The prototype system is flashy and compelling on the 19-inch workstation screen. It contains only a tiny amount of the information contained in the proposed full-scale application, but even a casual observer is impressed by how appealing it looks. The colors, graphics and images scanned from documents and books, animation, and sound effects come together seamlessly to create a compelling proof-of-concept.

A senior executive in the organization compares the hypermedia prototype with the original text-based system and declares the Hyper-X version a smashing success. All that remains is to "scale it up" to a production version by converting the remainder of the information to hypermedia.

How you react to this case study depends on where you are in your hypermedia project, or whether you are part of the prototype team or a member of the team tasked to put it into full-scale production. But the prototype is almost certain to fail to make it into production (Glushko, 1991).

Problems for Hypermedia Projects

The case study has followed a typical scenario for hypermedia projects that has set the stage for failure. The problems that face hypermedia projects can be categorized by the phase in the application life cycle at which they arise. Each set of problems represents a barrier to a successful project. Not all of the problems are specific to hypermedia projects, but they conspire with the novelty and immaturity of hypermedia to make hypermedia applications hard to design, develop, and deploy successfully.

In this paper I consider problems of three types: problems that arise when the project is started, problems that emerge during the design and development phases, and problems at the deployment and maintenance phases. Loosely put, these are problems with getting started, getting it done, and getting it to work.

Getting Started

Unrealistic Expectations About Scale and Readiness

Much of what one hears and reads about hypermedia has an artistic or whimsical flavor that presents a captivating vision in which text, graphics, voice, video, and other diverse information sources are much easier to use, "seamlessly integrated" by links with each other, and more entertaining to explore. Hypermedia is often cast as a revolutionary form of writing, posed in stark contrast with reactionary printed text. Hypermedia is said by visionaries to liberate readers from the restrictive confines of the printed page, which herds them through text in a strict, predefined sequence. This perspective is exemplified by authors of literary criticism and interactive fiction, as well as by advertisements for hypermedia programs that suggest that the more clever or creative the link between pieces of informa-

tion, the better. An early brochure for Apple's HyperCard program juxtaposed a map of Italy with pictures of pasta and a boot.

The potent combination of hype by vendors and enthusiasm in the popular press often leads to unrealistic expectations about how hard it is to design and develop a hypermedia application of realistic scale. Most organizations have no experience base that allows them to make realistic resource and schedule estimates.

The novelty of hypermedia often makes it relatively easy to obtain funds for a small-scale prototype. Many hypermedia projects rely on a combination of internal funding and external sponsorship from computer vendors, government agencies, and foundations. But when the start-up funds run out and the time comes for full-scale development, it is usually much harder to find the resources to pay for the tedious "grunt work" of data preparation and testing, and harder still to pay for ongoing maintenance.

Simple extrapolation from the effort required by the prototype is usually sobering. If a demonstration project takes three months to convert five articles from an encyclopedia into an interactive hypermedia form, how long will it take to convert a thousand articles using the same techniques? Automatic, semiautomatic, or template-based techniques are the only realistic option for large projects, but these are almost never applied during the initial prototyping effort, which usually focuses on user interface concerns. Structural analysis of the information, data modeling, and designing for acceptable performance as the database scales up are not as exciting as user interface concerns, but they turn out to be far more important for projects that aspire to succeed beyond the prototype stage.

Missing Skills on Design and Development Team

Project novelty also makes it easy at first to bring together top-notch people whose efforts benefit from ad hoc inter-organizational cooperation, since everyone wants to be part of an exciting new project at the beginning. Nevertheless, hypermedia projects may require (or at least benefit from) a broader mix of skills than are often readily available. Appropriate skills in a text-intensive project team include:

- Software designers
- User interface designers
- Usability testers and potential users
- Technical writers and editors
- Indexers
- Database designers

Hypermedia projects can require still other talents, including graphic artists and people who understand how to integrate multiple media, sometimes called "art directors". For projects involving the conversion of existing information from printed form, the participation of the author or editor of the original information can be invaluable in understanding the presentation conventions of the existing format. If neither is available, the project should allow additional time to understand the design and design rationale of the existing

information. Likewise, the people who run the museum or factory, repair the airplanes, or otherwise are expert in the application area must be involved, or the project runs the risk of building a carefully-designed user interface to the wrong information.

Few organizations or individuals have expertise in more than a few of these areas. Even if the needed mix of skills for a project organization can be assembled by matrix management or by bringing in appropriate consultants, making such an interdisciplinary team work effectively together is no small challenge.

Getting it Done

Few Published Case Studies or Design Guidelines

Because hypermedia is a relatively new design field, there are few detailed published case studies or design guidelines that designers can readily use. Published reports about hypermedia are not representative, typically biased toward small-scale demonstrations or research projects. While hypermedia applications of practical scale have been successfully designed and implemented, in general such projects are not documented in the literature because of resource constraints in development organizations, proprietary considerations, or because they are classified for security reasons.

While there is a growing body of empirical research evaluating particular hypermedia systems or specific design options, this work does not usually generalize well. In addition, what formal experiments have been able to establish is that most of the design choices, when considered in isolation, have only small percentage impacts on system usability (Nielsen, 1989). To complicate matters, sometimes small changes in the user's task or the structure of the hypermedia can lead to conflicting answers about the relative merit of design features (Wright & Lickorish, 1990). Individual differences in users, especially motivational differences, and the effects of different tasks seem to have major effects on system usability. Yet who the users are and the tasks they want to carry out are often not something the hypermedia system designer can control.

Poor Quality or Availability of Source Information

Many hypermedia conversion projects are plagued by the poor quality or availability of source information. Many documents have no digital form, and even when one exists, unless a hypermedia version was planned or contracted for when the documents were created, the existing digital form may not be readily usable.

One common limitation derives from the language with which the digital version of the text was composed or marked-up. The ideal situation is when the markup language is SGML (Standard Generalized Markup Language) or some other language that specifies the logical structure of the document rather than its presentation (Barron, 1989; Goldfarb, 1990).

Optical character recognition (OCR) technology is rapidly improving, and new OCR devices that output text in SGML form are especially promising. Nevertheless, error rates are non- negligible, so proofreading is always required, and the nature of the residual errors in OCR documents makes manual text entry viable unless correct recognition rates exceed 98% (Cushman, Ojha, & Daniels, 1990).

Likewise, images might exist in numerous incompatible formats, and while converting them to a single format is always an option, some of the most common formats are optimized for particular platforms.

Taken together, potential problems with source information make it essential that hypermedia projects carefully investigate source quality and availability before committing to a project schedule. Before converting information to standard formats, the project should consider the range of computing platforms on which the application is likely to be deployed during its lifetime.

Lack of Appropriate Software Tools

Most off-the-shelf hypermedia software is oriented toward creating small amounts of new information and is not well-suited for converting existing information (Alschuler, 1989; Glushko, 1990a; Glushko, 1990b). Demonstration projects often use this software to create expectations about the look and feel of a full-scale implementation, and it often comes as a harsh shock to discover fundamental limitations in the capacity or performance of the software.

User programmability is a prominent feature of this class of software, but the price for a high-level programming language is often poor performance for large amounts of information. Furthermore, most of these programs bundle their user interfaces and the information they present, an architecture that is incompatible with networked applications. Hence, supporting simultaneous users requires duplication of systems and software with corresponding expense and maintenance headaches.

It may be worth waiting for the next generation of hypermedia software that directly supports conversion and that employs a client-server architecture. Alternatively, some database programs or expert system shells may better support hypermedia features than programs that call themselves hypermedia.

If off-the-shelf software must be used for a hypermedia project, it is imperative that any demonstration or proof-of-concept phase carefully address pragmatic issues of scaling up. These include both capacity concerns - can the program manage significantly larger amounts of information with acceptable performance - and resource concerns - does the program imply or impose design methods for defining units, links, or other features that are infeasible when applied on a large scale (Glushko, 1990a, 1990b)?

Legal Uncertainties

In recent years there has been a rash of "look and feel" copyright infringement lawsuits and similar claims for software patents. These legal controversies have arisen because software has been defined both as a kind of "literary work," which makes it copyrightable, and as a kind of machine or method of operating one, which makes it patentable. While these legal analogies may be wrong and may someday be corrected by a new intellectual property law that recognizes the special character of software (Samuelson, 1989b), today software designers and developers are faced with chaos, uncertainty, and legal action.

As unclear as the situation is for software in general, the novel character of hypermedia and hypermedia software raises still more complexities for intellectual property law. For example, if copyright law has different rules for "literary works", "audiovisual works", "sound recordings", and "pictorial works", into what legal category does an interactive hypermedia encyclopedia or a talking book fall? Are new links or notes in a hypermedia system considered "derivative works" under copyright law? These and other issues are not just legal curiosities; they will have considerable impact on the legal protection available and hence the economic viability of hypermedia systems (Samuelson & Glushko, 1991).

The best defense against a copyright infringement claim is to be able to prove independent development, so keeping careful documentation of design decisions is essential. In addition, designs based on experiments or evaluations give the design a "functional" character that narrows the scope of copyright infringement claims. It is best to follow the golden rule when designing a system: Borrow from others no more than you would have them borrow from you. An alternative formulation of this principle can be found in a well-reasoned paper that presents both sides of the look-and-feel debate: Let he who has never borrowed cast the first lawsuit (Samuelson, 1989a).

One aspect of copyright infringement that confronts hypermedia designers and developers is clear and well-known, but new technology has made it easier to break the law. OCR, scanners, digital samplers and video "frame grabbers" are among the wide variety of technology that makes it possible to copy almost anything and incorporate it into a hypermedia system. But having the technology does not imply the right, and a sure way to invite a lawsuit is to assume that it does.

But while it is clearly necessary to obtain the rights to use copyrighted material, the mechanisms for obtaining rights to use copyrighted information are oriented toward the use of works or archives in their entirety. Negotiating the rights to assemble a heterogeneous collection of bits and pieces for a hypermedia application is emerging as a substantial barrier. It is not a coincidence that many hypertext applications have used government documents like standards and regulations that are free of copyright restrictions. For hypermedia applications, publishers and other "content-owners" are being courted by computer hardware, software, and telecommunications companies who are developing business strategies and product lines for multimedia systems..

Getting it to Work

Installed Base Constraints

Hypermedia demonstration projects are often done in research organizations that have advanced technology, including workstations and high-resolution 19-inch monitors. HyperCard on the Macintosh computer is also a very popular environment for demonstration projects.

In contrast, the users for whom full-scale versions of these demonstration systems must be targeted often work with an older or different installed base of computing equipment. This installed base may consist predominantly of IBM AT-compatible processors with small display screens having limited graphics resolution.

This situation often poses a dilemma for hypermedia projects. Advanced technology may be needed to demonstrate the benefits of hypermedia capabilities, but the presentation of these capabilities in the demonstration projects exceeds what the installed base will support. It is essential that the funding or marketing organization promoting the project know the costs and tradeoffs implied by various technology alternatives. Which is more successful, a project that uses less-advanced technology to create lower expectations that can be met, or a project that uses state-of-the art technology that is not readily available for the average user? There is no right answer, but it is essential to ask the question when project goals are being established.

In any case, the installed base slowly changes, so hypermedia designs should be prepared to take advantage of new technology capabilities as they become available. A key consideration is separating the user interface and information management portions of the hypermedia system so that the user interface can be enhanced as the installed base permits it.

Maintaining the deployed application

A hypermedia system deployed in a museum, factory, classroom, or elsewhere inevitably needs to be changed and maintained. Sometimes it is difficult to anticipate the uses and users of a system until it is built, and some tuning of the user interface and content is required. More generally, however, there is no way to get everything right at the outset because things simply change. The museum changes its exhibits, the factory upgrades its machinery, and so on.

Nevertheless, most hypermedia applications to date have been deployed with static and unalterable snapshots of information. Most hypermedia authoring software encourages this practice with separate authoring and run-time versions, with expensive price tags for the former and relatively inexpensive distribution fees for the latter. Hypermedia applications that contain indexes and images that occupy significant space are often attracted by the high-capacity but static storage provided by CD-ROM technology. Furthermore, it is certainly

easier to build a static system than to design data structures and user interfaces that enable easy system maintenance.

However, systems that are hard to keep up-to-date fall into disuse. In some hypermedia applications, systems that contain incorrect or inaccurate data can even become dangerous, and the users themselves must assume the primary responsibility for keeping the information "evergreen" (Warren, 1990).

Designing an application for maintenance might also involve redesigning the process by which the information contained in the application is created. For example, hypermedia projects that involve periodic publication of text created elsewhere should define formatting standards and quality control procedures for the organization that produces the information. These measures can lead to substantial improvement in the productivity of hypermedia conversion by enabling the development of automatic conversion software.

Summary

Hypermedia is an attractive vision, but practical hypermedia applications are hard to build. Disciplined approaches to analyzing information, identifying constraints in its structure and in the task environment, and using the appropriate implementation technology are required. Successful hypermedia projects are those that take a cautious approach to problems of scale and that make the right tradeoffs along the way.

References

Alschuler, L. (1989). "Hand-crafted hypermedia: Lessons from the ACM experiment." In E. Barrett (Ed.), *The Society of Text: Hypermedia, Hypermedia, and the Social Construction of Information*(pp. 343-361). MIT Press.

Barron, D. (1989). "Why use SGML?" *Electronic Publishing*, 2(2), 3-24.

Cushman, W., Ojha, P., & Daniels, C. (1990). "Usable OCR: What are the minimum performance requirements." *Proceedings of the CHI'90 Conference on Human Factors in Computing Systems*, 145-151.

Glushko, R.J. (1990a). "Using off-the-shelf software to create a hypermedia electronic encyclopedia." *Technical Communication*, 37(1), February 1990, 28-33.

Glushko, R.J. (1990b). "Visions of grandeur?" *Unix Review*, 8(2), 70-80.

Glushko, R. J. (1991). "Seven ways to make a hypertext project fail." *Technical Communication*, 38(4).

Goldfarb, C.F. (1990). *The SGML Handbook.* Arlington, VA: Graphic Communications Association.

Nielsen, J. (1989). "The matters that really matter for hypermedia usability." *Hypermedia '89 Proceedings*, 239-248.

Samuelson, P. (1989a). "Protecting user interfaces through copyright: The debate." *Proceedings of the ACM Conference on Computer-Human Interaction - CHI '89*, 97-103.

Samuelson, P. (1989b). "Why the look and feel of software user interfaces should not be protected by copyright law." *Communications of the ACM*, 32(5), 563-572.

Samuelson, P., & Glushko, R. (1991). "Intellectual property rights for digital library and hypertext publishing systems: Reflections on Xanadu." *Hypertext '91 Proceedings.*

Warren, B. (1990). "The Object the Metaphor the Power and Evergreen." *Hypermedia and Information Reconstruction.* University of Houston-Clear Lake and NASA/Johnson Space Center.

Wright, P., & Lickorish, A. (1990). "An empirical comparison of two navigation systems for two hypermedias." In C. Green & R. McAleese (Eds.), *Hypermedia: Theory into practice II.* Intellect Press.

