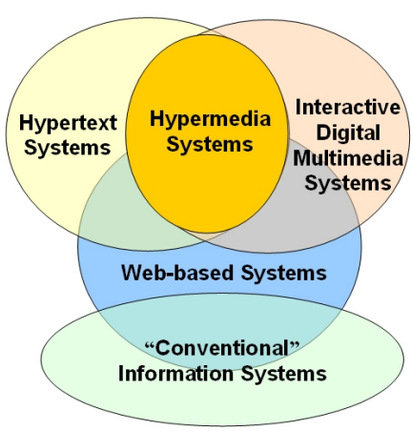
Designing Web-Based Hypermedia Systems

Michael Lang
National University of Ireland, Galway, Ireland

INTRODUCTION

Although its conceptual origins can be traced back a few decades (Bush, 1945), it is only recently that hypermedia has become popularized, principally through its ubiquitous incarnation as the World Wide Web (WWW). In its earlier forms, the Web could only properly be regarded a primitive, constrained hypermedia implementation (Bieber & Vitali, 1997). Through the emergence in recent years of standards such as eXtensible Markup Language (XML), XLink, Document Object Model (DOM), Synchronized Multimedia Integration Language (SMIL) and WebDAV, as well as additional functionality provided by the Common Gateway Interface (CGI), Java, plug-ins and middleware applications, the Web is now moving closer to an idealized hypermedia environment. Of course, not all hypermedia systems are Web based, nor can all Web-based systems be classified as hypermedia (see Figure 1). See the terms and definitions at the end of this article for clarification of intended meanings. The focus here shall be on hypermedia systems that are delivered and used via the platform of the WWW; that is, Web-based hypermedia systems.

Figure 1. Hypermedia systems and associated concepts



There has been much speculation that the design of Web-based hypermedia systems poses new or unique challenges not traditionally encountered within conventional information systems (IS) design. This article critically examines a number of issues frequently argued as being different—cognitive challenges of designing non-linear navigation mechanisms, complexity of technical architecture, pressures of accelerated development in "Web-time" environment, problems with requirements definition, the suitability of traditional design methods and techniques, and difficulties arising out of the multidisciplinary nature of hypermedia design teams. It is demonstrated that few of these issues are indeed new or unique, and clear analogies can be drawn with the traditions of conventional IS design and other related disciplines.

CRITICAL REVIEW OF PRINCIPAL DESIGN ISSUES AND CHALLENGES

Visualizing the Structure of Hypermedia Systems

Essentially, hypermedia attempts to emulate the intricate information access mechanisms of the human mind (Bush, 1945). Human memory operates by associating pieces of information with each other in complex, interwoven knowledge structures. Information is later recalled by traversing context-dependent associative trails. Hypermedia permits the partial mimicry of these processes by using hyperlinks to create non-linear structures whereby information can be associated and retrieved in different ways. Otherwise put, hypermedia facilitates multiple paths through a network of information where there may be many points of entry or exit. This especially is the case with Web-based hypermedia, where users can enter the system through a variety of side doors rather than through the front "home page". The undisciplined use of

hyperlinks can lead to chaotic "spaghetti code" structures (de Young, 1990). As systems scale up, this causes the substantial problem of "getting lost in cyberspace," whereby it becomes very difficult to locate information or navigate through the labyrinth of intertwined paths (Otter & Johnson, 2000; Thelwall, 2000).

Two principal reasons explain why difficulties in visualizing the structure of a hypermedia system may arise. First, non-linear navigation mechanisms lead to intricate multi-dimensional information architectures that are hard to conceptualize. Second, Web-based hypermedia systems are typically an amalgam of many different interconnected components, such as static Hypertext Markup Language (HTML) pages, client-side applets or scripts (e.g., Java, Javascript), dynamically generated pages (e.g., PHP, Perl, Active Server Pages, ColdFusion), media objects (e.g., JPEG, VRML, Flash, Quicktime) and back-end databases. Flows and dependencies are not as visible in Web-based hypermedia systems as they are for most conventional systems, and it can be quite difficult to form a clear integrated picture of the technical architecture (Carstensen & Vogelsang, 2001).

However, the phenomenon of systems being constructed using a multiplicity of components is not unique to Web-based hypermedia. In conventional systems design, tiered architectures that separate data, logic and interface layers are commonly used to assist seamless integration. One such approach is the Model-View-Controller (MVC) framework, which has also been found beneficial in Web-based hypermedia design (Izquierdo, Juan, López, Devis, Cueva & Acebal, 2003). Nor is the difficulty of designing non-linear navigation mechanisms unique to hypermedia. Within traditional printed media, certain types of material are intentionally designed to be used in a random-access non-linear manner, such as encyclopediae, thesauruses and reference works. According to Whitley (1998), hypermedia systems are different from other types of software applications because "the developers have to set up a number of alternatives for readers to explore rather than a single stream of text" (p. 70). This may be a new concept in software design, but elsewhere, technical writers have long experience of setting up multiple navigable paths in the design of electronic documentation, such as online help systems. It has been found that technical writing techniques can readily be adapted to navigation design for Web-based hypermedia systems (Eriksen, 2000).

Accelerated Development Environment

The capacity of organizations to respond and adapt quickly to rapidly changing environments is a well-recognised strategic issue. Accordingly, IS need to be flexible and able to adapt to changing business needs. Looking at trends in IS development over the past 20 years, project delivery times have dramatically shortened. In the early 1980s, Jenkins, Naumann and Wetherbe (1984) reported that the average project lasted 10.5 months. By the mid-1990s, the duration of typical projects had fallen to less than six months (Fitzgerald, 1997), and average delivery times for Web-based systems are now less than three months (Barry & Lang, 2003; Russo & Graham, 1999). These accelerated development cycles have given rise to the notion of "Web time" or "Internet speed" (Baskerville, Ramesh, Pries-Heje & Slaughter, 2003; O'Connell, 2001; Thomas, 1998), a development environment that is supposedly characterized by "headlong desperation and virtually impossible deadlines" (Constantine & Lockwood, 2002, p. 42).

Such compressed timeframes are made possible by the combined effect of two factors. First, modern-age, rapid-application development tools greatly speed up the development process, although it is sometimes argued that What-You-See-Is-What-You-Get (WYSIWYG) visual design tools invite a reckless "just-do-it" approach without much, if any, forethought. Second, the Web is an immediate delivery medium which, unlike traditional IS and off-the-shelf software applications, is not impeded by production, distribution and installation delays. Web-based systems can be easily and quickly launched by developing functional front-end interfaces, powered by crude but effective back-end software, which can later be modified and enhanced in such a manner that end users may be oblivious to the whole process.

Again, however, this phenomenon of reduced cycle times is not specific to Web-based hypermedia design, for it also affects the design of conventional systems (Kurata, 2001). Yourdon (1997) defined "death march" projects as those for which the normal parameters of time and resources are re-