A Critical Review of Challenges in Hypermedia Systems Development

A CRITICAL REVIEW OF CHALLENGES IN HYPERMEDIA SYSTEMS DEVELOPMENT

Michael Lang *

1. INTRODUCTION

Over the past few years there has been considerable academic interest in the development of "Web-based" systems, much of it surrounding the contention that it is somehow different from "traditional" or "conventional" systems development. However, the debate is clouded because of confusion over the meaning of the phrase "Web-based system".[1] Indeed, it may be argued that "Web-based" is just an adjective which does not essentially alter the meaning of the term "information system" when prefixed to it.[2] This view is upheld by an examination of definitions from the literature, which clearly make no fundamental distinction:

> *"Web-based information systems (WIS) are information systems (IS) that are based on Web technology and they are likely to be tightly integrated with conventional IS such as databases and transaction processing systems"* [3]

> *"Web Information Systems are Systems, not Pages. WISs are information systems first, and Web systems second"* [4]

As Lockwood and Constantine put it, "current development tools make it easy to 'browserize' almost anything",[5] so little if any redesign may be required to effect basic migration of a system to the Web. For example, some intranet projects have been as straightforward as Web-enabling existing back-end applications such as Lotus Notes databases. It is therefore obvious that although a system may be said to be "Web-based", this doesn't necessarily imply it is any different from a non-Web-based system as regards software design considerations.

Some clarity can be introduced by thinking of the Web as a "global hypermedia system".[6] Hypermedia technologies support much richer user interfaces, more complex navigation mechanisms and more varied forms of information than conventional systems. As Web-based systems assume hypermedia functionality, they become distinct from conventional systems. Although the Web is not an ideal hypermedia environment, it is nevertheless the most common platform for hypermedia systems at the present time. In

* Department of Accountancy & Finance
NUI Galway, University Road, Galway, Ireland
Telephone: +353 91 750301 Fax: +353 91 750565 Email: Michael.Lang@nuigalway.ie

recognition of this point, the ACM Special Interest Group on Hypertext and Hypermedia (formerly SIGLINK) now goes by the acronym SIGWEB. This paper therefore considers interactive Web-based systems within the broader classification of hypermedia systems. This interpretation is also guided by Lee's point that IS researchers should aspire to produce timeless contributions.[7] "Hypermedia systems" is a better term than "Web-based systems" not just because it is less ambiguous, but also because it is a more enduring concept and embraces technologies that pre-date the Web (*e.g. on-line help, encyclopaedia CD-ROMs*) as well as those that follow the Web (*e.g. WAP and interactive TV applications).* Indeed, the term "Web-based information system" may soon become a redundant anachronism. The scope of the Web is expanding with the range of delivery devices, - witness "e-commerce" giving birth to offshoot terms such as "m-commerce" and "t-commerce", - what Botterweck and Swatman call "Web-like applications".[8] It is likely that over time most systems will be ported to the Web and that Web-based interfaces will become the norm, as testified by Microsoft's zeal to win the "browsers war". If most systems were to become "Web-based", the adjective would probably be dropped, as has happened with "*multimedia* PC", a popular term from the early 1990s that is now defunct.

Before proceeding, a number of other terms warrant clarification. Where used in this paper, *traditional systems development* refers to popular practices in the period from the 1970s to the late 1980s, until displaced by new approaches upon the arrival of visual programming, RAD tools, GUIs, object-orientation, and open systems architectures. The subsequent period from the early 1990s to present shall be referred to as the *modern age. Conventional systems* refer to systems and applications of the standard types encountered within organisations, such as transaction processing systems and management information systems. Lastly, *traditional hypermedia development* refers to applications from the pre-Web era, such as interactive CD-ROMs, online help systems, and Apple Hypercard applications.

The objectives of this paper are to critically review a number of issues encountered within hypermedia systems development that are often argued with little if any justification to be radically different, and to assess if these issues are indeed *new* (that is, not experienced in traditional systems development or in other disciplines) or *unique* (that is, not experienced in the development of conventional systems in the modern age, or in other disciplines). In brief, these issues are: cognitive challenges of designing non-linear navigation mechanisms, pressures of accelerated development in a "Web-time" environment, problems arising out of the external "virtual" nature of the end-user population, the appropriateness of traditional design methods and techniques, and difficulties attributable to the multidisciplinary composition of hypermedia design teams.

2. ISSUES AND CHALLENGES IN HYPERMEDIA DEVELOPMENT

There has been much speculation that the development of "Web-based" information systems poses new or unique challenges.[9-11] However, the assumption of "newness" is a common weakness in both systems development research and practice.[12-14] Of late, there is a growing posse of dissenters who argue that there is nothing substantially different about Web-based systems development.[1,2,15,16] Given the problems earlier alluded to concerning the definition of "Web-based systems", this ought not be surprising. In the following sub-sections, each of the purported new/unique challenges shall be considered

within the broader umbrella of hypermedia systems development. Just as it is appropriate to situate a discussion of Web-based systems within the context of hypermedia, we should also consider hypermedia within the traditions of its contributory root disciplines, which include graphic design, information science, technical writing, literary theory, media production, and database systems.

2.1. Complexity of Navigation Mechanisms

Some definitions of hypermedia emphasise non-linearity as its essential differentiating characteristic. Because of this non-linearity, navigation mechanisms within hypermedia systems can become quite complex. Arbitrary hyperlinking results in chaotic structures, leading to problems such as "getting lost in cyberspace", locating information, visualising system organisation, and managing content. This is where diagrammatic modelling techniques become useful for system developers, as they help to overcome the cognitive difficulties of understanding complex, abstract structures. However, it has been argued that diagramming techniques from traditional systems development are inadequate for modelling hypermedia systems.[9,17]

Structural complexity arising from interconnectedness is not a new problem in software design. In the early days, there was the 1960s practice of 'go-to' programming, which Dijkstra criticised as being "*an invitation to make a mess*".[18] In the 1970s, complex navigation and relationship structures were a feature of network databases. Out of those experiences arose a number of principles and techniques, such as modularity and normalisation, which are readily applicable to the design of information nodes within hypermedia systems.[19] More recently, the flow of control in visual event-driven and object-oriented programming languages is such that traditional techniques such as structured flowcharts are of limited use. As a consequence, modern age software diagramming techniques such as UML are being applied, as well as techniques drawn from traditional dynamic media (e.g. storyboarding). Storyboarding is borrowed from the film industry, where non-linear narrative has long been used for dramatic effect. Both storyboarding and UML can also be applied to hypermedia systems modelling, - indeed, a number of UML extensions have been proposed specifically for hypermedia design.[20,21]

Looking to traditional literature, many authors have experimented with non-sequential interactive fiction (e.g. Borges [22]), and there are numerous examples stretching back to antiquity of branching stories and interlocking commentaries. The discipline of library information science also has long experience of non-linear systems that require the investigation of side links, and librarians' skills are relevant to the design and evaluation of Web sites.[23,24] Likewise, techniques from technical writing and electronic documentation can be applied to hypermedia design.[25,26] Even within traditional printed media, there are certain types of material that are specifically designed to be used in a random-access non-linear manner, such as encyclopaediae, thesauruses and reference works. Physically these are linear sequences of content units, but logically they are more complex because the units may be indexed and cross-referenced. According to Whitley, hypermedia systems are different from other types of software applications because their design involves "*a process of structuring ideas, describing the order of presentation, and conceptual exploration … the developers have to set up a number of alternatives for readers to explore rather than a single stream of text*". [27] However, technical writers have long had to set up navigable paths in designing online help systems, and likewise have authors of non-linear printed materials.

2.2. Accelerated Development Cycles

A much-cited "new" challenge is the pressure of accelerated development cycles, often referred to as "Web time" or "Internet speed".[28,29] Certainly, looking at trends in IS development over the past 20 years, delivery times have dramatically shortened. In the early 1980s, Jenkins et al reported that the average project then lasted 10.5 months.[30] By the mid-1990s, the duration of typical projects had fallen to less than 6 months,[31] and average delivery times for Web-based systems are now less than 3 months.[9,32-34] Such compressed timeframes are unprecedented in traditional systems development or traditional hypermedia development, and are made possible by the combined effect of two factors. Firstly, 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. The second enabling factor is modern age rapid application devlopment tools. Arguably, this second factor is the more important, and the jargon term "Web time" is misleading because it tends to suggest that the coming of the Web alone brought about these accelerated timescales.

Ever-shortening product cycles has always been an observable fact of life within the IT industry, even before the advent of the Web.[35]. Back in the 1960s "space age", NASA were rushing to produce software in the race to the moon. Short deadlines and limited resources have long been the bane of IS project managers.[36,37] As such, "Web time" may be said to be just a continuation of this general tendency. Moreover, it is a phenomenon that is not specific to Web or hypermedia development, but applies also to conventional systems.[38] This is reflected by the growing interest in high-speed approaches such as agile methods, RAD, timeboxing, and COTS configuration amongst the general community of software developers.

Indeed, one could say that this trend is reflective of a greater urgency in business in general, brought about not just by the Web but also by other advances in telecommunications, transportation, and computing technologies, as well as practices such as JIT and BPR. Business in the modern age is characterised by a faster metabolism, time-based competition, shorter windows of opportunity, rapidly changing and uncertain environments, and a need for greater flexibility and adaptability.[39,40] Considered thus, the phenomenon of "Web time" is not unique to Web, hypermedia, or conventional systems development, although it must be acknowledged that product life cycles are much shorter in the IT industry than in many other industries.[35] However, for many "new media" companies, "Web time" is not at all new, because they have always faced strict, pressing deadlines and have accordingly adapted their processes to the pressures of Web delivery.[41,42]

2.3. Virtual End-User Population

Russo and Graham make the point that "*Web applications differ from traditional information systems [because] the users of Web applications are likely to be outside of the organization, and typically cannot be identified or included in the development process*".[9] This, like the previous issue, is not hypermedia-specific and arises out of the nature of the Web. However, because most modern hypermedia applications are delivered via the Web or "Web-like" platforms (such as PDAs and interactive TV), it shall be considered under the banner of hypermedia development.

It is plainly true that for most Web-based information systems, with the obvious exception of intranets, end-users are external to the organisation. This is indeed new territory for IS development, because information systems traditionally served internal functions.[43] Collecting requirements from a virtual population is difficult, and traditional ISD techniques cannot be easily applied if at all.[44] As a consequence, requirements are often vague. Baskerville and Pries-Heye state that "*an inability to pre-define system requirements is the central, defining constraint of Internet time development … often a project starts without a requirement specification*".[29]

Let us examine both these aspects more closely. Firstly, the notion of a virtual population, although new to IS developers, is quite typical for mass-market off-the-shelf software production and new product development.[43] In such situations, the marketing department fulfils a vital role as the voice of the customer.[44] Established marketing research techniques can be used in conjunction with traditional requirements elicitation techniques and Web usage analysis techniques to define requirements for a virtual population.[45] For example, Tognazzini describes how a team of designers, engineers, and human factors specialists used scenarios to define requirements based on an understanding of the profiles of target users as communicated by marketing staff.[46]

Secondly, the phenomenon of vague requirements is by no means new. Glass remarks that "*back in the earliest days of software development … [specifications] often didn't exist, or when they did, they were written on the back of an envelope*",[47] Walz et al observe that users often "don't know what they want" and fuzzy requirements are common,[48] and of course there is the classic problem of "I'll know it when I see it" requirements that are not pre-specifiable.[49]

Another aspect of a virtual user population, which is different from traditional IS development, is that end users often cannot be personally trained how to use the system. Constantine and Lockwood assert that "*much more so than standard software, Web applications must focus on the user experience*".[50] Certainly, issues such as usability and interaction design are paramount for Web/hypermedia systems, but they are arguably no less a consideration for off-the-shelf software applications. Indeed, the importance of ease-of-use for mass-produced goods has long been emphasised by industrial designers. There is however a further element of Web-based user experience design which traditionally has not been considered by software designers, - branding and corporate image. Unlike off-the-shelf applications and conventional systems development, Web systems have a "public relations" aspect.[51] This new aspect of IS development is best handled by those with specialist skills, - marketing personnel, graphic designers and communications consultants.

2.4. Need for Specialised Development Methods and Approaches

It is often argued that approaches and methods from traditional systems development are inappropriate for Web/hypermedia development.[3,9,17,42] Murugesan et al speak of "*a pressing need for disciplined approaches and new methods and tools*",[52] taking into account "*the unique features of the new medium*". As demonstrated in this paper, it is arguable if many of the features of hypermedia are indeed unique, for many parallels may be drawn with traditional/conventional software design and other root disciplines. Merely because an application is based on new technologies, its design should not necessarily require an altogether new or different approach.[4,13] It may well be true that traditional systems development methodologies are ill-suited to hypermedia development. However,

for the same reasons, those methodologies can be argued to be inappropriate for conventional systems development in the modern age.[53] Modern approaches, methods and techniques, - such as rapid prototyping, incremental development, agile methods, use case modelling, class diagrams, GUI schematics, model-view-controller framework, and heuristic evaluations, - may be said to be as applicable to hypermedia development as to conventional systems development.

However, traditional methodologies ought not be entirely discarded for hypermedia development. Some authors maintain that an adapted form of the classical SDLC remains the most appropriate process model,[54,55] an assertion that is further supported by empirical evidence that the SDLC and variants thereof remain in popular use.[56-58] Furthermore, Barry and Lang reveal that traditional approaches from other root disciplines, such as graphic design and media production, are also being used in hypermedia systems development.[34] It would therefore seem that new, specialised hypermedia-specific approaches are not required. Although many such approaches have been proposed in the literature, - such as RMM, OOHDM, VHDM, EORM, WSDM, WebML, and OntoWebber, - a recent study of hypermedia development practice reveals that these are not being used, but that methods from traditional and conventional systems development are being used.[58]

2.5. Multidisciplinary Design Teams

A notable aspect of hypermedia systems development is that design teams typically involve members from a diversity of professional backgrounds.[32,34,59] Of course, skills diversity is not unique to hypermedia systems development, - many conventional projects, particularly large ones, necessitate the integration of various knowledge domains.[48] However, in conventional systems development, designers tend to be primarily "computer professionals", which is typically not the case with hypermedia systems development. This is especially true of Web-based hypermedia systems, where many developers do not have a background in traditional software design or programming.[9,33,51] The challenge of managing communication and collaboration within multidisciplinary design teams is by no means trivial, and if mismanaged is potentially disastrous. Experiences from Web and interactive multimedia development projects reveal that discrepancies in the backgrounds of team members can give rise to significant communication and collaboration problems.[51,60] Kim makes the point that:

> *"disciplines are like cultures: for disciplines to work well together they must learn to appreciate one another's language, traditions, and values … Different disciplines have different priorities, different thinking styles, different values. When people from different disciplines get together, their values collide."* [61]

This is analogous to Kuhn's notion of "paradigms".[62] It has been noted that separate paradigms can co-exist within the same field yet be mutually ignorant of each other, either deliberately or inadvertently.[63] Although "paradigms" and "disciplines" are not strictly equivalent, they are closely related. Within hypermedia systems development, two such disciplines are software engineering and graphic design. Despite being the two foremost disciplines in hypermedia development,[58] it has been observed that they appear to operate in distinctly different worlds and have quite different value systems.[64,65] For example, software engineering and graphic design have markedly different interpretations of concepts such as "quality", "elegance" and "structure".

On the face of it, this appears to be a new challenge, although one could argue that communication problems are not new for they have traditionally plagued systems development projects.[66] The need to address the balance between functionality and aesthetics has previously arisen in other disciplines, such as civil engineering versus decorative architecture, automobile design, and computer game design. Within the realm of conventional systems development, user interface design has also long been a multidisciplinary, collaborative activity.[67,68] Kautz and Nørbjerg argue that the move towards multidisciplinary teams is a continuation of an observable trend across systems development in general.[16] This position is supported by the contention of Fafchamps and Garg that flexible teams, characterised by composite membership and roles and diverse disciplines and skills, are "*a new type of organizational entity that will become more prevalent in the future*".[69] Whereas hypermedia systems development seems to call for a greater level of multi-/cross-skilling than traditional systems development,[42,56,70] this may likewise be argued to be an increasingly common phenomenon in the digital networked economy (i.e. "reprogrammable labour").

3. SUMMARY AND CONCLUSIONS

Jakob Nielsen has commented that "*software design is a complex craft and we sometimes arrogantly think that all its problems are new and unique*".[71] As presented in Table 1, many of the challenges of hypermedia systems development are neither unique nor entirely new. IS researchers have often been guilty of jumping onto the bandwagon of the lastest fad, leading Keen to remark that many "*issues seen as 'new' turn out to have long roots*".[12] With the advent of so-called "new media", much of the discourse amongst communications scholars has likewise been at fault of lacking a "historical consciousness".[72]

It is clear that hypermedia development is a multidisciplinary domain which can potentially draw from lessons and experiences across a variety of disciplines – traditional IS development, software engineering, HCI, graphic design, visual communications, marketing, technical writing, library information science, media production, architecture, and industrial design. For now, this is what most distinguishes hypermedia development from traditional/conventional systems development, although this distinction is likely to disappear in the future as software development in general is headed in this direction.

The multidisciplinary nature of design teams must be acknowledged in devising mechanisms to resolve challenges of hypermedia development. Design approaches, integrated working procedures, diagramming techniques, toolset selection, and methods for specifying and managing requirements must all take this central aspect into consideration.

The software engineering and ISD literature is filled with a multitude of methods and techniques aimed towards Web/hypermedia design, very few of which are used in practice. This might be explained in a variety of ways: lack of awareness, lack of tool-based support, lack of guidance on how to use them, or inertia being obvious possibilities. However, there is a much more fundamental matter which has received very little attention, – that of the narrow world-view of the method developers. Morgan makes the point that researchers in all branches of science "*often approach their subject from a frame of reference based upon assumptions that are taken-for-granted*".[73] For methods and techniques to be of value, the assumptions upon which they are founded should be in

Table 1. Summary of Key Points.

Challenge	Related Trends, Technologies, and Experiences
Complexity of Navigation Mechanisms *(designing and visualising non-linear structures)*	Traditional systems development • go-to spaghetti code (structured programming techniques). • network and relational databases (data modelling techniques). Conventional systems development in the modern age • visual, event-driven, object-oriented programming (UML). Other disciplines • technical writing techniques for production of electronic documentation. • cataloguing mechanisms in library information science. • planning of scenes and shooting order for film pre-production (storyboarding).
Accelerated Development Cycles *(pressures of "Web time")*	Traditional systems development • short deadlines and tight resources have long been a reality. • ever-shortening product cycles has been a notable trend. Conventional systems development in the modern age • high speed approaches such as rapid prototyping, RAD, agile methods, COTS configuration are growing in popularity. Modern business environment • faster metabolism, greater urgency, need for agility are norms. Other disciplines • "old media" firms moving to "new media" have long experience of pressing deadlines e.g. news production.
Virtual End-User Population *(requirements definition, user experience design)*	Traditional/conventional systems development • traditional IS development was in-house, but traditional off-the-shelf software development has always been for an external population. • initially vague requirements and fuzzy specifications are an age-old issue. • "I'll know it when I see it" requirements are common in the modern age. • usability is recognised as an important consideration for all modern systems. Other disciplines • marketing researchers have always had an external focus. • visual communications discipline has experience of developing brands in diversified marketplaces. • in manufacturing, industrial designers have long emphasised intuitive, easy-to-use features in the development of mass-produced goods.
Need for Specialised Methods and Approaches	Traditional/conventional systems development • traditional IS development methods may be inappropriate for hypermedia development, but likewise are inappropriate for conventional systems development in modern age. • modern methods, approaches and techniques for conventional systems development can be used for hypermedia development. Hypermedia development • variants of classical SDLC remain in popular use in hypermedia development. • specialised approaches e.g. OOHDM, WSDM are not being used in practice Other disciplines • traditional approaches from graphic design and media production can also be adapted to hypermedia development.
Multidisciplinary Design Teams	Traditional/conventional systems development • traditional IS project teams have long been cross-functional (e.g. accounting, marketing, etc.) but developers were primarily software engineers. • design teams for interactive systems with visual GUIs are often multidisciplinary. Modern business environment • multi-skilling and cross-skilling are facets of digital networked economy. • flexible teams are becoming an increasingly common organisational entity. Other disciplines • trade-offs between functionality and aesthetics have been experienced elsewhere e.g. building design, automobile design., computer game development

congruence with the context within which they are intended to be used. Methods and techniques for hypermedia systems development should therefore recognise its multidisciplinary nature and take a cosmopolitan world-view, integrating and adapting approaches from the various root disciplines. To paraphrase a well-known quotation, those who choose not to draw from the well of cumulative knowledge are bound to foolishly repeat past mistakes. This paper therefore concludes with a call to hypermedia design researchers to reach out and explore the historical experiences of other related disciplines.

4. FURTHER RESEARCH

This literature review paper is the product of an ongoing doctoral research project and draws upon insights gained from exploratory empirical work.[58,74] The next stage of this project involves the definition of a conceptual framework to analytically compare the various disciplines that contribute to hypermedia systems development. This comparison shall analyse data captured from literature trawls and interviews with developers. Researchers with similar or related interests are welcome to contact the author.

5. REFERENCES

1. J. Holck, 4 perspectives on Web information systems, in: *Proceedings of 36th Annual Hawaii International Conference on System Sciences (HICSS)*, (IEEE Computer Society Press, Hawaii, USA, January 2003).
2. C. Barry, Web-based information systems - time for the revisionists, in: *Proceedings of Collaborative Electronic Commerce Technology and Research (CollECTeR) Conference*, edited by T. Acton and P. Swatman (CISC, National University of Ireland, Galway, June 24 2003), pp. 37-50.
3. S. Wang, Toward a general model for web-based information systems, *International Journal of Information Management* **21**(5), 385-396 (2001).
4. A. R. Dennis, Lessons from three years of Web development, *Communications of the ACM* **41**(7), 112-113 (1998).
5. L. Lockwood and L. Constantine, Taming Web development, *Software Development Magazine*, (April 1999), http://www.sdmagazine.com/documents/sdm9904h/.
6. M. Andreessen and E. Bina, NCSA Mosaic: a global hypermedia system, *Internet Research* **4**(1), 7-17 (1994).
7. A. S. Lee, *The timeliness of publications in MIS Quarterly*, (ISWorld Mailing List, June 5, 1999).
8. G. Botterweck and P. A. Swatman, Towards a contingency based approach to Web engineering, in: *Proceedings of 7th Australian Workshop on Requirements Engineering (AWRE'2002)*, edited by J. L. Cybulski *et al.* (Melbourne, Australia, December 2-3 2002), pp. 47-64.
9. N. L. Russo and B. R. Graham, A first step in developing a Web application design methodology: understanding the environment, in: *Methodologies for Developing and Managing Emerging Technology Based Information Systems: Proceedings of 6th International BCS Information Systems Methodologies Conference*, edited by A. T. Wood-Harper *et al.* (Springer, London, 1999), pp. 24-33.
10. T. Isakowitz, M. Bieber and F. Vitali, Web information systems, *Communications of the ACM* **41**(7), 78-80 (1998).
11. D. B. Lowe and J. Eklund, Client needs and the design process in Web projects, *Journal of Web Engineering* **1**(1), 23-36 (2002).
12. P. G. W. Keen, Relevance and rigor in information systems research: Improving Quality, Confidence, Cohesion and Impact, in: *Information Systems Research: Contemporary Approaches and Emergent Traditions*, edited by H.-E. Nisson *et al.* (Elsevier Science Publishers, 1991), pp. 27-49.
13. E. Yourdon, Developing Applications for the Internet: advice for the Java generation, *American Programmer*, 36-41 (December 1996).
14. R. S. Pressman, What a tangled web we weave, *IEEE Software* **17**(1), 18-21 (2000).
15. T. Butler, An institutional perspective on developing and implementing intranet- and internet-based information systems, *Information Systems Journal* **13**(3), 209-231 (2003).

16. K. Kautz and J. Nørbjerg, Persistent problems in information systems development: the case of the World Wide Web, in: *Proceedings of 11th European Conference on Information Systems (ECIS)*, edited by C. Ciborra *et al.* (Naples, Italy, June 16-21 2003).
17. K. Siau and M. Rossi, Information modeling in the Internet age - challenges, issues and research directions, in: *Information Modeling in the New Millennium*, edited by M. Rossi and K. Siau (Idea Group Publishing, Hershey, PA, 2001), pp. 1-8.
18. E. W. Dijkstra, Go to statement considered harmful, *Communications of the ACM* **11**(3), 147-148 (1968).
19. M. J. Taylor, S. Wade and D. England, Informing IT system Web site design through normalisation, *Internet Research: Electronic Networking Applications and Policy* **13**(5), 342-355 (2003).
20. H. Baumeister, N. Koch and L. Mandel, Towards a UML extension for hypermedia design, in: *UML'99: The Unified Modeling Language - Beyond the Standard, Second International Conference, Fort Collins, CO, USA, October 28-30, 1999, Proceedings. Lecture Notes in Computer Science 1723*, edited by R. B. France and B. Rumpe (Springer, 1999), pp. 614-629.
21. J. Conallen, *Building Web Applications with UML*, (Addison Wesley, Reading, MA, 2000).
22. J. L. Borges, The garden of the forking paths, in: *Labyrinths: Selected Stories and other Writings*, edited by D. A. Yates and J. E. Irby (Penguin Books, Harmondsworth, 1981), pp. 44-54.
23. J. Conklin, Hypertext: an introduction and survey, *IEEE Computer* **20**(9), 17-20;32-41 (1987).
24. S. Shropshire, Beyond the design and evaluation of library web sites: an analysis and four case studies, *The Journal of Academic Librarianship* **29**(2), 95-101 (2003).
25. B. B. Zimmerman, Applying Tufte's principles of information design to creating effective Web sites, in: *Proceedings of 15th ACM Conference on Systems Documentation*, (ACM Press, Snowbird, Utah, USA, October 19-22 1997), pp. 309-317.
26. L. B. Eriksen, Limitations and opportunities for system development methods in Web information system design, in: *Organizational and Social Perspectives on Information Technology, IFIP TC8 WG8.2 International Working Conference on the Social and Organizational Perspective on Research and Practice in Information Technology, June 9-11, 2000, Aalborg, Denmark*, edited by R. Baskerville *et al.* (Kluwer, Boston, MA, 2000), pp. 473-486.
27. E. A. Whitley, Method-ism in practice: investigating the relationship between method and understanding in Web page design, in: *Proceedings of 19th International Conference on Information Systems (ICIS)*, (Helsinki, Finland, December 13-16 1998), pp. 68-75.
28. D. Thomas, Web time software development, *Software Development Magazine*, 78-80 (October 1998).
29. R. Baskerville and J. Pries-Heye, Racing the e-bomb: how the Internet is redefining information systems development methodology, in: *Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspective. Proceedings of International Federation for Information Processing (IFIP) Working Group 8.2 Conference, Boise, Idaho, USA, 27-29 July 2001*, edited by N. L. Russo *et al.* (Kluwer Academic Publishers, Boston, MA, 2001), pp. 49-68.
30. M. A. Jenkins, J. D. Naumann and J. C. Wetherbe, Empirical investigation of systems development practices and results, *Information & Management* **7**(2), 73-82 (1984).
31. B. Fitzgerald, The use of systems development methodologies in practice: a field study, *Information Systems Journal* **7**(3), 201-212 (1997).
32. P. R. Vora, Designing for the Web: a survey, *ACM interactions* **5**(3), 13-30 (1998).
33. S. McClure, *Web Application Development Developer Perspectives: An IDC White Paper*, (International Data Corporation, Framingham, MA, 1998).
34. C. Barry and M. Lang, A comparison of "traditional" and multimedia information systems development practices, *Information and Software Technology* **45**(4), 217-227 (2003).
35. H. Mendelson and R. R. Pillai, Clockspeed and informational response: evidence from the information technology industry, *Information Systems Research* **9**(4), 415-433 (1998).
36. T. Gilb, Deadline pressure: how to cope with short deadlines, low budgets and insufficient staffing levels, in: *Information Processing*, edited by H. J. Kugler (Elsevier Science Publishers B.V., 1986), pp. 293-299.
37. E. Yourdon, *Death March: The Complete Software Developer's Guide to Surviving "Mission Impossible" Projects*, (Prentice Hall, Upper Saddle River, NJ, 1997).
38. D. Kurata, Do OO in "Web time", *Visual Basic Programmer's Journal* **11**(1), 70 (2001).
39. J. C. Wetherbe and M. N. Frolick, Cycle time reduction: concepts and case studies, *Communications of the AIS* **3**(13), 1-42 (2000).
40. G. Scott, Internet/Web systems development: what can be learned from hi-tech new product strategic planning, in: *Proceedings of 36th Annual Hawaii International Conference on System Sciences (HICSS)*, (IEEE Computer Society Press, Hawaii, USA, January 2003).
41. V. Bellotti and Y. Rogers, From Web press to Web pressure: multimedia representations and multimedia publishing, in: *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems*, (ACM Press, Atlanta, Georgia, USA, March 22-27 1997), pp. 279-286.

42. J. Greenbaum and D. Stuedahl, Deadlines and work practices in new media development: its about time, in: *PDC 2000 Proceedings of Participatory Design Conference*, edited by T. Cherkasky *et al.* (New York, USA, November 28 - December 1 2000), pp. 70-77.
43. J. Grudin, Interactive systems: bridging the gaps between developers and users, *IEEE Computer* **24**(4), 59-69 (1991).
44. J. Lazar, E. Hanst, J. Buchwalter and J. Preece, Collecting user requirements in a virtual population: a case study, *WebNet Journal* **2**(4), 20-27 (2000).
45. M. S. Lane and A. Koronois, A balanced approach to capturing user requirements in business-to-consumer Web information systems, *Australian Journal of Information Systems* **9**(1), 61-69 (2001).
46. B. Tognazzini, *Tog on Software Design*, (Addison Wesley, Reading, MA, 1995).
47. R. L. Glass, Who's right in the Web development debate?, *Cutter IT Journal* **14**(7), 6-10 (2001).
48. D. B. Walz, J. J. Elam and B. Curtis, Inside a software design team: knowledge acquisition, sharing, and integration, *Communications of the ACM* **36**(10), 63-77 (1993).
49. B. Boehm, Requirements that handle IKIWISI, COTS, and rapid change, *IEEE Computer* **33**(7), 99-102 (2000).
50. L. L. Constantine and L. A. D. Lockwood, Usage-centered engineering for Web applications, *IEEE Software* **19**(2), 42-50 (2002).
51. P. H. Carstensen and L. Vogelsang, Design of Web-based information Systems - new challenges for systems development?, in: *Proceedings of 9th European Conference on Information Systems (ECIS)*, (Bled, Slovenia, June 27-29 2001), pp. 536-547.
52. S. Murugesan, Y. Deshpande, S. Hansen and A. Ginige, Web engineering: a new discipline for development of Web-based systems, in: *Proceedings of 1st ICSE Workshop on Web Engineering*, (ACM Press, Los Angeles, California, USA, May 16-17 1999), pp. 1-9.
53. B. Fitzgerald, Systems development methodologies: the problem of tenses, *Information Technology & People* **13**(3), 174-185 (2000).
54. T. A. Powell, D. L. Jones and D. C. Cutts, *Web Site Engineering: Beyond Web Page Design*, (Prentice Hall, Upper Saddle River, NJ, 1998).
55. C. Urquhart, Exploring methodologies for Web based design: a case study of a design business, in: *Proceedings of 1st AIS SIGeBiz Workshop on e-Business (WeB 2002)*, (Barcelona, Spain, December 14-15 2002), pp. 1-9.
56. C. Britton, S. Jones, M. Myers and M. Sharif, A survey of current practice in the development of multimedia systems, *Information and Software Technology* **39**(10), 695-705 (1997).
57. J. Paynter and M. Pearson, *A case study of the Web-based information systems development*, (Department of Management Science and Information Systems, University of Auckland, New Zealand, 1998).
58. M. Lang, Hypermedia systems development: a comparative study of software engineers and graphic designers, *Communications of the AIS* **12**(16), 242-257 (2003).
59. L. L. Scarlatos, R. P. Darken, K. Harada, C. Heeter, R. Muller and B. Shneiderman, Designing interactive multimedia (Panel), in: *Proceedings of 5th ACM International Conference on Multimedia*, (ACM Press, Seattle, Washington, USA, November 9-13 1997), pp. 215-218.
60. O. Koechlin, *Methods and Tools to Improve Software Quality for Multimedia Productions. Final Report ESSI Project No. 21545*, (European System and Software Initiative, December 1997).
61. S. Kim, Interdisciplinary cooperation, in: *The Art of Human-Computer Interface Design*, edited by B. Laurel (Addison Wesley, Reading, MA, 1990), pp. 31-44.
62. T. S. Kuhn, *The Structure of Scientific Revolutions*, (University of Chicago Press, Chicago, IL, 1970).
63. M. Dogan, Paradigms in the social sciences, in: *International Encyclopedia of the Social & Behavioral Sciences*, edited by N. J. Smelser and P. B. Baltes (Elsevier Science, Oxford, 2001), pp. 11023-11027.
64. L. Vertelney, M. Arent and H. Lieberman, Two disciplines in search of an interface: reflections on a design problem, in: *The Art of Human-Computer Interface Design*, edited by B. Laurel (Addison Wesley, Reading, MA, 1990), pp. 45-55.
65. S. Gallagher and B. Webb, Competing paradigms in multimedia systems development: who shall be the aristocracy?, in: *Proceedings of 5th European Conference on Information Systems (ECIS)*, edited by R. D. Galliers *et al.* (Cork Publishing Ltd., Cork, Ireland, June 19-21 1997), pp. 1113-1119.
66. B. Curtis, H. Krasner and N. Iscoe, A field study of the software design process for large systems, *Communications of the ACM* **31**(11), 1268-1287 (1988).
67. J. Grudin and S. E. Poltrock, User interface design in large corporations: coordination and communication across disciplines, in: *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems*, (ACM Press, Austin, Texas, USA, April 30 - May 4 1989), pp. 197-203.
68. J. Preece, *Interaction Design*, (Wiley, New York, 2002).
69. D. Fafchamps and P. Garg, Computing environments for flexible teams, in: *Software Engineering and Human-Computer Interaction. ICSE '94 Workshop on SE-HCI: Joint Research Issues, Sorrento, Italy, May*

16-17, 1994, Proceedings (LNCS 896), edited by R. N. Taylor and J. Coutaz (Springer-Verlag, Berlin, 1995), pp. 174-184.
70. N. P. Kotamraju, Keeping up: Web design skill and the reinvented worker, *Information, Communication & Society* **5**(1), 1-26 (2002).
71. J. Nielsen, Learning from the real world, *IEEE Software* **14**(4), 98-99 (1997).
72. E. Huhtamo, From cybernation to interaction: a contribution to an archaeology of interactivity, in: *The Digital Dialectic: New Essays on New Media*, edited by P. Lunenfeld (MIT Press, Cambridge, MA, 1999), pp. 96-110.
73. G. Morgan, Paradigms, metaphors, and puzzle solving in organization theory, *Administrative Science Quarterly* **25**(4), 605-622 (1980).
74. M. Lang, Reconsidering the "software crisis": a study of hypermedia systems development, in: *Proceedings of IADIS International WWW/Internet 2003 Conference*, edited by P. Isaías and N. Karmakar (IADIS Press, Algarve, Portugal, November 5-8 2003), pp. 307-313.