



## **“Crisis” or “wonderland”: An appraisal of hypermedia systems development practice**

Title	“Crisis” or “wonderland”: An appraisal of hypermedia systems development practice
Author(s)	Lang, Michael
Publication Date	2004
Publisher	International Association for the Development of the Information Society

## **“CRISIS” OR “WONDERLAND”: AN APPRAISAL OF HYPERMEDIA SYSTEMS DEVELOPMENT PRACTICE**

**Michael Lang** *National University of Ireland, Galway*

*Michael.Lang@nuigalway.ie*

### **ABSTRACT**

With the advent and growth of the Web, hypermedia information systems have propagated within and beyond organisations. Much concern has been expressed about the quality of hypermedia systems being developed and the apparent absence of disciplined development practices. There has been talk that the infamous “software crisis” is afflicting hypermedia systems development, allegedly characterised by shoddy project management, inadequate requirements analysis and planning, and ad hoc “quick and dirty” development approaches. This paper reports on the findings of a survey of 438 organisations in Ireland, the objectives of which were to test the validity of assertions of a “hypermedia crisis”, and to explore what mechanisms if any are being used to guide and control hypermedia systems development. The preliminary conclusions are that hypermedia development is much more disciplined than popularly believed, and the state of practice is much healthier than depicted by many academic researchers.

### **KEYWORDS**

Hypermedia; Web development; systems development methods; software crisis.

## **1 INTRODUCTION**

In the early days of software development, computer applications were typically constructed in an ad hoc, unsystematic fashion (or at least, so the story goes!). As applications began to grow in complexity, major difficulties in managing software development projects soon came to the fore. And so, at the 1968 NATO Conference on Software Engineering in Garmisch-Partenkirchen, the infamous phrase “software crisis” was first uttered (Naur & Randell, 1969). Brooks (1987) uses the metaphor of “a monster of missed schedules, blown budgets, and flawed products” to convey the essence of this problem. Some authors have argued that it is contradictory to describe a phenomenon which spans a long number of years as a “crisis”, and that “software depression” (Floyd, 1979) or “chronic affliction” (Pressman, 1997 p. 16) would

be more correct. Nevertheless, “software crisis” is now firmly established in the colloquial lexicon.

Over the years, there has been a rather condescending bias in the academic literature that the state of practice is deplorable and shameful. For example, Boehm (1976) expressed his opinion that the average developer was “sloppy, inflexible, in over his head, and undermanaged”, while Ward (1992) lambasted the “lack of professional discipline among the great unwashed masses of systems developers”. A riposte by the Irish writer, Oscar Wilde, springs to mind: “we are all in the gutter, but some of us are looking at the stars!” In my view, it is not at all helpful that some academics have chosen to take the moral high ground, pontificating to practitioners from the lofty pedestals of the ivory towers. There has been one notable dissenting voice, that of Robert Glass, in defence of the state of practice. He takes the position that the tales of abominable failures which are repeatedly cited as evidence of the existence of a “software crisis” are in fact “exception reporting”, because if one were to look at things in a positive frame the reality is that “today’s world rolls on the wheels of software solutions”. All about us in everyday life are visible examples of software successes, which he celebrates as a “veritable wonderland” (Glass, 1998).

This paper shall consider the “software crisis” within the specific context of hypermedia systems development. Although its conceptual origins can be traced back a few decades (Bush, 1945), it is only recently that hypermedia has become popularised through its ubiquitous incarnation as the Web. Of course, not all hypermedia systems are Web-based, nor can all Web-based systems be classified as hypermedia (this point is briefly elaborated in the next section). Nevertheless, the Web is the most common platform for hypermedia systems today, - as acknowledged by the fact that the ACM Special Interest Group on Hypertext and Hypermedia, initially set up in 1987 before the arrival of the Web, now goes by the acronym SIGWEB (formerly SIGLINK). Because of the newness and dynamism of hypermedia technologies and the apparent lack of disciplined development methods, concern has been expressed about the quality of systems being delivered. The soothsayers of doom are at it again, likening the maturity of present-day hypermedia development practice to that of software development in the 1960s, some going so far as to speak of an imminent “hypermedia crisis”:

*“with the demand for more complex multimedia systems in a wide range of areas, the prospect for a multimedia equivalent to the software crisis looks increasingly likely” (Britton et al., 1997)*

*“... [hypermedia applications development] is usually quick&dirty, resulting in low correctness, robustness, and maintainability of the end products” (Pauen et al., 1998)*

*“The potential Web crisis could be more serious and widespread than the software crisis” (Murugesan et al., 1999)*

*“In many cases, the development approaches used for Web-based systems have been ad hoc, reminiscent of early days of application software development ... Overall, software development for the Web lacks rigour and a systematic approach” (Murugesan & Deshpande, 1999)*

*“[Internet technology] has become a breeding ground for important WebApps that are hacked in much the same way as important applications software was hacked a few generations back – in the 1960s and 1970s” (Pressman, 2000)*

*“Despite the existence of software engineering methods for hypermedia development, this process is not as systematic as it could be expected and, in fact, the ‘hypermedia software crisis’ still remains” (Aedo & Díaz, 2001)*

It is difficult to accept these harshly generalised assertions given that there is not much firm objective data to support them. So far, there has been very little rigorous wide-scale empirical research into Web or hypermedia systems development; to the author’s knowledge, only four survey-based studies have been previously published in the mainstream literature (Vora, 1998; Russo & Graham, 1999; Barry & Lang, 2001; Lowe & Eklund, 2002).

The survey described herein therefore aims to contribute to a better understanding of the realities of hypermedia systems development practice. Its objectives were:

- to investigate the extent to which the problems that characterise the alleged “hypermedia crisis” actually exist in reality;
- to explore what, if any, mechanisms are used to guide and control the practice of hypermedia systems development.

## 2 DEFINITION OF TERMS

### 2.1 “Hypermedia” and “Web-based” Systems

Before proceeding, I feel it is necessary to justify why I prefer to speak of “hypermedia systems” rather than “Web-based systems”, given that there has been a flurry of interest in recent years in the design of “Web-based” systems, and that many authors, as in the aforementioned excerpts, tend to refer to the “Web” rather than “hypermedia”. My aversion to the term “Web-based system” rests upon a contention that it is neither meaningful nor enduring. The adjective “Web-based” merely communicates that a system is based upon a Web platform, - no more, no less. For example, some intranet projects have been as straightforward as Web-enabling existing back-end applications such as Lotus Notes databases, with little or *no* redesign necessary. It is therefore clear that although a system may be said to be “Web-based”, that doesn’t necessarily imply it is *any* different from a non-Web-based system as regards software design considerations.

Hypermedia technologies support much richer user interfaces, more complex navigation mechanisms and more varied forms of information than conventional systems. I argue that it is only when Web-based systems assume hypermedia functionality that they become distinct from conventional systems. Thus, interactive Web-based systems should properly be considered within the broader traditions of hypermedia systems design, a more enduring legacy which includes previous generations (e.g. electronic encyclopaediae such as Encarta) as well as future generations (e.g. advanced interactive TV, WAP). As depicted in Figure 1, hypermedia systems are seen as a media-rich form of hypertext and a specialised sub-class of interactive digital multimedia systems. Standard definitions of hypermedia emphasise critical aspects such as flexible structuring of information, support for loosely restricted non-linear navigation by means of hyperlinks, and support for multiple media and various data formats. Most, but not all, modern hypermedia systems are Web-based; conversely, most, but not all, modern Web-based systems can be categorised as hypermedia.

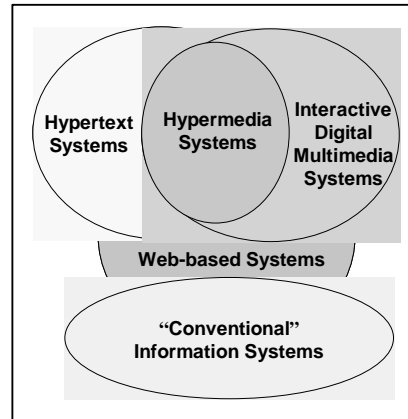


Figure 1. Hypermedia and associated concepts.

For the purposes of this study, hypermedia is defined as “any interactive software system that permits a user to navigate through hyperlinked information by means of various user-selected paths”. This includes such applications as interactive Web sites, electronic catalogues, intranets, courseware / CBT, interactive e-commerce systems, portals, and online information services.

## 2.2 “Process”, “Approach”, and “Method”

Terms such as “process”, “approach”, and “method” are difficult, if not impossible, to define in such a manner that one can always clearly distinguish between them. Nevertheless, there have been fervent disputes in the literature over the precise meanings of these terms. Iivari et al (1998) define an information systems development [ISD] approach as “a set of goals, guiding principles, fundamental concepts, and principles for the ISD process that drive interpretations and actions in ISD”. Simply put, an “approach” can be thought of as “a way of going about [doing something]” (Galliers, 1992). “Methods” are generally seen as being more concrete than approaches, as exemplified by the following definitions:

*“[a method is] a systematic approach to conducting at least one complete phase (e.g design or requirements analysis of software production, consisting of a set of guidelines, activities, techniques and tools, based on a particular philosophy of system development and the target system” (Wynekoop & Russo, 1995)*

*“A method is an approach to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products.” (Brinkkemper, 1996)*

On the other hand, one sometimes encounters loose definitions of “method”, such as “a generic guide to help people perform some activity” or even simply “a mixed bag of guidelines and rules” (Rumbaugh, 1995).

Meanwhile, a “software process” might be defined as:

*“the series of activities regarding a software product from the time the need to which that product is identified until the time the product is retired (which may occur before installation).” (Blum, 1994)*

*“the sequence of stages (e.g requirements analysis, specification, planning, design, implementation, integration, maintenance and retirement) through which a software product evolves” (Wynekoop & Russo, 1995)*

Clearly, these are not tight definitions. One person’s “process” could be considered by another to be an “approach” or “method”, and the point at which an “approach” becomes a “method” is not easy to pin down. In a sense this is rather academic, for in practice the terms are used quite loosely and interchangeably. Accordingly, I did not attempt to “rigorously” define these terms, but rather, as I expected respondents would do, adopted a common sense interpretation.

### 3 RESEARCH METHOD

A dual-mode survey was conducted in Ireland, by post and on the Web, in the winter of 2002/2003. Appropriate authentication mechanisms were engaged to assure instrumental rigour and validity for both modes. As is often the case with organisational surveys, the definition of an accurate sampling frame was difficult. Here, the population included companies engaged in general bespoke systems development; those specialising in Web, multimedia, or hypermedia systems development; those from traditional media that have branched into “new media”; and those that have internal IS departments (e.g. financial services firms and banks). The initial sample was compiled from a number of classified industry databases. It was then systematically reduced, based on descriptions of activities and portfolios of work as described on Web sites and in secondary data sources. As an additional sifting mechanism, a number of questions were introduced into the questionnaire to ensure that only those respondents that developed hypermedia systems of substantive scale and complexity, as defined by a combination of check variables, were included in data analysis. Prior to distribution, the survey was pilot tested with a purposefully selected group of mixed experience from mixed professional backgrounds, using the “talk aloud protocol” advocated by Dillman (2000). In addition, professional technical writers assisted with the wording and visual layout of the questionnaire so as to reduce the possibility of measurement error.

The final population consisted of 417 organisations, but a few of these had multiple divisions that were separately included, giving an overall tally of 438. It was decided to sample the entire population. In 425 cases the names of individuals in designated positions were known. For the remaining 13, the questionnaire was addressed to the “Head of Software Development”. The cover letter requested that the questionnaire be completed by someone in a design role, such as software design, information architecture, or creative design. Only one response was solicited from each organisation because they were mostly quite small and it was felt that to ask for multiple responses might have led to outright refusal.

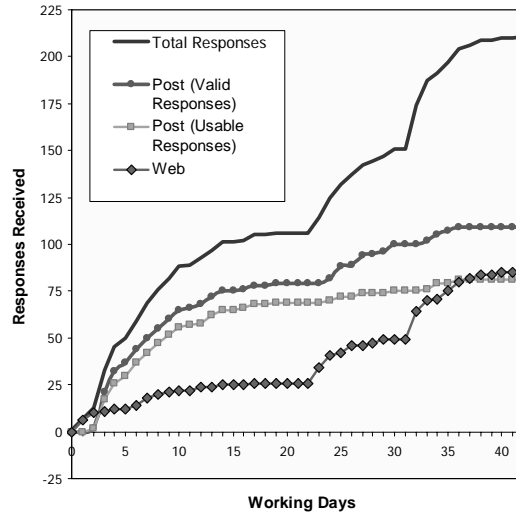


Figure 2. Survey response patterns.

Reminders were sent out by post after 4 weeks (20 working days), and again by e-mail after a further 2 weeks, the affects of which are clearly seen in Figure 2. A total of 214 valid responses were received, - 112 by postal mail, 88 via the Web, 12 by e-mail, and 2 by other media. One organisation responded twice, which was taken into consideration for those questions where the unit of analysis is the organisation rather than the individual. In addition, 23 questionnaires were returned undelivered or with a note that the organisation had ceased operations. This gave an overall organisational response rate of 51.3% (213 from 415 active). However, 42 respondents indicated that they had no significant experience of hypermedia systems design. Another 5 responses were insufficiently complete. Thus the usable response rate was 44.5% (166 from 373) based on the size of the true population. This greatly exceeds response rates for previous surveys of systems design (typically of the order of 10%-20%) and strengthens the validity of the findings.

## 4 SYNOPSIS OF FINDINGS

### 4.1 Profile of Respondents' Organisations and Development Environment

The primary business, size of organisation, and size of development teams of respondents' organisations are shown in Table 1. As can be seen, most of the organisations were small, consisting of 50 employees or less. A search of the Kompass industry database ([www.kompass.com](http://www.kompass.com)) was conducted to compare these findings with the structure of industry in other countries in Western Europe. It was found that this distribution profile is quite typical, within 5% of the regional aggregate for each category.

“CRISIS” OR “WONDERLAND”: AN APPRAISAL OF HYPERMEDIA SYSTEMS  
DEVELOPMENT PRACTICE

Table 1. Profile of respondents' organisations (n=166)

Primary business		Size of organisations	
Web Development	26%	1-20 employees	67%
IT / Software Development	14%	21-50 employees	10%
Graphic Design / Media Production	10%	51-100 employees	4%
Multimedia Development	7%	101-500 employees	6%
Portals	7%	more than 500 employees	13%
Interactive Communications / Advertising	6%	Size of development teams	
e-Learning/CBT	5%	1 developer	5.4%
Financial Services	5%	2 to 4 developers	57.8%
Management Consultancy	5%	5 to 10 developers	30.7%
Miscellaneous	14%	More than 10 developers	6.0%

Table 2. Duration and cost of projects.

Project duration		Project cost	
5% Trimmed Mean = 14.3 weeks Median = 10.5 weeks (n=140)		5% Trimmed Mean = € 41,382 Median = € 18,000 (n=77)	
Less than 4 weeks	6.4%	Less than € 5,000	15.6%
4 to 8 weeks	21.4%	€ 5,000 to € 9,999	19.5%
8 to 12 weeks	23.6%	€ 10,000 to € 19,999	15.6%
12 to 20 weeks	18.5%	€ 20,000 to € 49,999	22.1%
20 to 30 weeks	15.0%	€ 50,000 to € 99,999	10.4%
30 weeks or more	15.0%	€ 100,000 or more	16.9%

(Exchange rate for Northern Ireland data UK£1.00 = €1.60)

Respondents were asked to indicate the actual duration and costs of their most recently delivered project of non-trivial complexity (as opposed to simple “brochureware” development). To avoid speculative responses, a “Don’t Know” category was provided (understandably, many respondents did not know what the costs were, presumably because management did not tell them). It was found that 51.4% of projects are delivered in 12 weeks or less, with a median delivery time of 10.5 weeks (Table 2), which is consistent with findings in other studies (Vora, 1998; Barry & Lang, 2001).

As seems to be the general trend in systems development, most teams in this study were small. Only 6% of respondents indicated that they normally work in teams of more than 10 developers, and in the majority of cases there are less than 5 team members (Table 1). Taking this finding alongside those of Table 2, one sees that the so-called “3 x 3” profile typifies hypermedia systems development in Ireland, - teams of 3 or so developers working to deliver a project in about 3 months.

Most of the systems developed by respondents were quite large, with 66.8% of them being over 50 pages/screens in size, - almost a third of which consist of more than 200 pages/screens. As one would expect, the most popular type of hypermedia systems developed was simple “brochureware”. Most respondents also had substantial experience of developing more complex hypermedia systems, such as transactional applications, portals / Web directories, and electronic catalogues. Looking at the characteristics of systems developed, most were database-driven, featured dynamically generated pages, and had frequently changing content.

## 4.2 Project Management

*“Most hypermedia applications are developed using an ad hoc approach. There is little understanding of development methodologies, measurement and evaluation techniques, development processes, application quality and project management”* (Lowe & Hall, 1999 p. 14)

At first glance, the data in Table 2 might convey a sense of a hectic work schedule, so-called “Web time”. This environment is supposedly characterised by “frenzied development” (Yourdon, 1996), “headlong desperation” (Constantine & Lockwood, 2002), “a handful of developers working frantically against the clock amid the litter of take-out food containers to churn out code” (Reifer, 2002), and “guerilla programming in a hostile environment using unproven tools, processes, and technology” (Thomas, 1998). In constrained timeframes, one would expect developers to resort to shortcuts and other time-saving devices that may not be as considered as ought be. Unconsidered actions are inherently risky and likely to cause problems over time, but when respondents were presented with a list of project management issues that typify the “software crisis”, it appears that few major problems are being experienced in practice (see Table 3). This suggests that either these problems are not as severe as popularly believed, or else that whatever mechanisms respondents are engaging to tackle them are highly effective.

Table 3. Experiences with project management issues in hypermedia systems design.

	n	No problems	Minor problems	Moderately problematic	Major problems
Controlling project scope / Feature creep	161	1.2%	39.1%	42.9%	16.8%
Preparing accurate time and cost estimates	156	3.8%	43.6%	45.5%	7.1%
Coping with accelerated timescales of Web environment	140	13.6%	55.7%	26.4%	4.3%
Controlling and coordinating project tasks	164	11.6%	64.6%	20.7%	3.0%
Managing communication between team members from different professional backgrounds	166	14.5%	62.0%	21.7%	1.8%

“CRISIS” OR “WONDERLAND”: AN APPRAISAL OF HYPERMEDIA SYSTEMS  
DEVELOPMENT PRACTICE

Not surprisingly, the most acute problem is the old classic: controlling project scope / feature creep. However, over 40% of respondents have minor or no problems here, and it is only a major concern for 1 in every 6. The next most significant issue is the preparation of time and cost estimates. As remarked by one of the pilot test participants, “formulating project plans is easy, it is doing so accurately that is difficult, and things always take longer to do than you at first imagine”. Nevertheless, project managers seem to be faring very well. 65.9% of projects are delivered within the agreed budget, and 32.2% are delivered on time, whereas time and cost over-runs of more than 50% arise in only 16.7% and 2.6% of cases respectively (Table 4). A likely explanation as to why there is more variance in the duration than in the cost of projects is that many systems seem to be delivered according to fixed price contracts. Coping with “Web time” delivery schedules, controlling and co-ordinating project tasks, and managing communication between project team members present problems for just a few respondents, and even then at a moderate level. In reply to another question (see Table 9), 62.9% of respondents indicated that their organisations had guidelines in place to cover project planning and estimation. All of these findings convey an impression that project managers in hypermedia systems development are competently discharging their responsibilities and are in reasonable control of their situation.

Table 4. Variance in project duration and costs [actual / planned].

	<b>Variance in project duration</b> n = 137 5% Trimmed Mean = 26.6% OVER	<b>Variance in project costs</b> n = 76 5% Trimmed Mean = ON TARGET
Between 50% and 10.01% UNDER	3.0%	10.6%
Between 10% and 0.01% UNDER	0.0%	7.9%
Exactly ON TARGET	29.2%	47.4%
Between 0.01% and 10% OVER	5.1%	7.9%
Between 10.01% and 25% OVER	24.1%	14.5%
Between 25.01% and 50% OVER	21.9%	9.2%
More than 50% OVER	16.7%	2.6%

### 4.3 Requirements Analysis

*“Most WWW developers delve directly into the implementation phase, paying little or no attention to requirements acquisition and specification and going through a very informal design phase (if any)” (Coda et al., 1998)*

*“Most of the web sites are created opportunistically without prior planning or analysis.” (De Troyer, 2001)*

*“Most Web developers pay little attention to requirements elicitation and analysis” (Ginige & Murugesan, 2001)*

There have been many allegations that requirements analysis is often shoddily done or bypassed altogether in hypermedia systems development. It therefore came as a surprise to find that 86.7% of respondents had actually used a written requirements specification document for their most recently completed project. Moreover, these specifications would seem to be more substantial than mere sketchy tenders padded with “sales pitch” forematerials, because the 5% trimmed mean length was 40 pages. The reported lengths of requirements specification documents were as follows: less than 10 pages (18.5%); 10 to 24 pages (29.0%); 25 to 49 pages (20.2%); 50 to 99 pages (16.1%); 100 pages or more (16.1%). It was also found that the level of guideline usage for Requirements Documentation was 63.5%. On the negative side, coping with volatile and changing requirements is a source of problems (Table 5), but it is well acknowledged that this is an irreconcilable “wicked problem”, for end-users can never quite get what they want (Paul, 1994).

Table 5. Experiences with requirements management issues in hypermedia systems design.

	n	No problems	Minor problems	Moderately problematic	Major problems
Coping with volatile and changing requirements	164	1.8%	38.4%	46.3%	13.4%

#### 4.4 Use of Processes, Methods, Approaches, Procedures and Guidelines

*“Many Web development shops have little structure or process in place to meet the need for sound engineering and maintainability ... many practitioners often emerge from self-taught ‘hacking’ climates that repel any process as overwhelmingly burdensome red tape” (Norton, 1999)*

*“The current state of application development on the Web is characterised by anarchy and ad hoc methodologies” (Enguix & Davis, 1999)*

*“quick and dirty development by means of various tools – if any –, ... that are driven by the underlying technology, is the state of practice” (Retschitzegger & Schwinger, 2000)*

*“even large mission-critical intranet projects are being started without any regard for methodology.” (De Troyer, 2001)*

*“Web developers often use ad hoc, hacker-type approaches, which lack rigor, systematic techniques, sound methodologies, and quality assurance” (Ginige & Murugesan, 2001)*

Contrary to the sentiments expressed in the above excerpts, the findings of this survey would seem to indicate that hypermedia systems development is actually quite disciplined. 83.6% of respondents said that their organisation uses a hypermedia development process that has clear tasks and/or phases within it. In slightly more than a half of these organisations, these processes are explicitly documented (Table 4). Only 16.4% of organisations do not have a clear process, 59.1% of whom consider this a problem.

“CRISIS” OR “WONDERLAND”: AN APPRAISAL OF HYPERMEDIA SYSTEMS  
DEVELOPMENT PRACTICE

Table 6. Organisation's hypermedia development process (n = 165).

There is no clear process	16.4%
Clear tasks and/or phases, though the process used is not explicitly documented	41.8%
Clear tasks and/or phases, according to an explicitly documented process	41.8%

A more revealing picture emerged in response to an open-ended question that asked respondents to “list the names of any hypermedia development methods or approaches that you have used”. Whereas previous studies revealed the prevalence of in-house methods for Web and multimedia systems development (Barry & Lang, 2001; 2003), it becomes apparent from this study that these are mostly not “methods” in the pure sense, but rather they are an eclectic mélange of approaches, process models and toolkits of techniques drawn from right across the board. Because many of the responses received were ambiguous, it was difficult to code them accurately and the categories overlap (see Table 5). Quite a few of the responses which indicated that an internal method or approach was used did not provide any details on its orientation, so caution should be taken in interpreting the table as the percentages in some categories may be understated.

The top response category was in-house methods (22.8%). For those in-house methods about which some detail was provided, they were mostly hybrids, including such peculiar blends as SSADM or SDLC/Waterfall with Extreme Programming, RUP, or RAD. This suggests that hypermedia systems developers, rather than shunning method, actually assemble fragments of methods, sometimes from apparently incompatible paradigms (e.g. traditional versus agile, structured versus object-oriented), and distil the most useful elements into a home-cooked in-house approach.

Given claims in the literature that traditional methods are ill-suited to hypermedia systems development (Greenbaum & Stuedahl, 2000; Siau & Rossi, 2001; Wang, 2001), it is somewhat of a surprise that this was the second highest response category (21.5%). Most of these were derivatives of SSADM or SDLC/Waterfall, though a few also mentioned Yourdon or Jackson Structured Programming.

The third highest category was rapid / agile methods. This was not surprising, except in so far as the incidence is probably lower than one would expect, but in all likelihood some of the in-house methods which provided no detail on their orientation would probably fall into this category, so it is under-represented.

There is also substantial incidence of development approaches that are focused around the use of tools, - a finding that lends some support to the assertion that developers “delve directly into the implementation phase” (Coda et al., 1998). However, there is widespread acceptance of the necessity for explicitly documented plans and considered action in preference to “ad hoc” just-do-it approaches (see Table 8). 93.9% of respondents agreed that there is an essential need for planning, and 79.3% agreed that plans and working methods should be clearly documented. Of course, one cannot infer actual behaviour from these idealised attitudinal values, but it is noteworthy that 68.3% of organisations use documented guidelines or procedures for some or other purpose (Table 9), which is suggestive of a broadly favourable disposition towards the virtues of order and consistency.

Table 7. Use of methods and approaches in hypermedia systems design (n = 79).

Hybrid or proprietary in-house method or approach	22.8%
Traditional “legacy” software development methods and approaches, or variants thereof e.g. <i>SSADM, Yourdon, JSP, SDLC / Waterfall</i>	21.5%
Rapid or agile development methods and approaches e.g. <i>RAD, Extreme Programming</i>	17.7%
Approaches that are focused around the use of tools and development environments e.g. <i>PHP, Java, Flash, ASP, J2EE</i>	15.2%
Object-oriented development methods and approaches e.g. <i>RUP, OOA&amp;D</i>	11.4%
Approaches that are focused around the use of techniques e.g. <i>Storyboards, Flowcharts, Wireframes, UML</i>	7.6%
No method used / development approach is “ad hoc”	7.6%
Incremental or evolutionary methods and approaches e.g. <i>Spiral Model, Staged Delivery, Iterative Design, Code &amp; Fix</i>	7.6%
HCI / Human Factors Engineering methods e.g. <i>User Centred Design, Interaction Design, Goal-based Requirements</i>	6.3%
Specialised non-proprietary methods for Web and hypermedia systems development e.g. <i>Fusebox, WSDM, OOHDM</i>	5.1%

Table 8. Attitudes to planning in hypermedia systems development.

	n	Firmly disagree	Disagree	Neutral	Agree	Firmly agree
Ad hoc “improvised” hypermedia development approaches <u>generally</u> result in systems of poor quality	153	5.2%	18.3%	7.8%	39.2%	29.4%
To combat system complexity and time pressures, there is <u>an essential need</u> for planning and considered action	165	0.0%	1.2%	4.8%	31.5%	62.4%
To ensure efficient and effective collaboration within the development team, plans and working methods should be <u>explicitly</u> documented	165	1.8%	5.5%	13.3%	36.4%	42.5%

Table 9. Use of documented procedures and guidelines.

Requirements documentation	63.5%	System testing & debugging	39.5%
Project planning & estimation	62.9%	Coding practices	34.7%
Interface design / Usability	50.9%	Use of diagramming techniques	24.0%
Technical design documentation	49.7%		

## 5 CONCLUSIONS

On the basis of the findings reported herein, there seems to be little evidence to believe that hypermedia systems development is in a “crisis”; on the contrary, developers seem to be competently dealing with the pressures they face. Although requirements management issues pose some problems, - which could be said to be an inevitable reality for all software developers, - projects are generally under control, with few major time or cost over-runs. Most organisations have some clearly visible “process”, and although that process is often undocumented, the value of documented plans and working methods is well accepted. Systems are being developed mainly using traditional software development methods or variants, in-house development methods, or rapid / agile approaches. Though there are many hypermedia-specific methodologies set forth in the academic literature (e.g. *RMM*, *OOHDM*, *WSDM*, *W3DT*), the findings of this survey reveal that only 2 of 94 respondents have ever used any of these and just 4 others are otherwise aware of them.

In view of this, it must be asked if not the real “crisis” is within academia? Wynekoop & Russo (1995) have warned that “by failing to evaluate current methodologies, practices and needs, researchers may develop methodologies that are not only irrelevant, but flawed”. The academic literature is already strewn with hundreds of development methods, many of which are arcane, impractical, and unworkable. With the emergence of Web and hypermedia systems there has been talk of a “pressing need for new methods and tools” (Murugesan et al., 1999). Oinas-Kukkonen et al (2001) claim that “systematic analysis and design methodologies for developing Web information systems are necessary and urgently needed among practitioners”. However, it is doubtful if there is a genuine need for *new* methods. As this study reveals, traditional software development methods can be readily adapted to the new challenges of hypermedia systems development.

Narayanan (1998) has warned that:

*“This situation of practice in the marketplace far out pacing the development of theoretical foundations can potentially lead to a situation where the ‘hype’ surrounding hypermedia wears off in the light of effectiveness, usability and other kinds of problems uncovered from the massive proliferation and use of such systems in all walks of life”*

In truth however, if one were to look at the history of technology stretching back to the Industrial Age, the state of practice has often led the way and informed theory, rather than vice versa (Glass, 1989). Now once again, developers in industry are going about their business, successfully producing hypermedia systems for, literally, the world to behold (on the Web), without recourse to academic solutions. Bearing this observation in mind, if academic

researchers wish to make useful contributions to hypermedia development practice, perhaps the best place to start is by learning from practice through grounded empirical research. A number of potential research objectives are:

- to produce guidance on how to adapt and apply existing methods and techniques from traditional and conventional systems development to the domain of hypermedia, rather than unnecessarily inventing wholly new methods and techniques;
- to investigate how methods and techniques from other contributory disciplines, such as visual design, media production, and technical writing, can be adapted to hypermedia systems development;
- to closer investigate the nature of hybrid and in-house hypermedia development methods, - in particular, the rationale for combining fragments of methods from apparently inconsistent paradigms;
- to better understand why traditional methods continue to be popularly used for hypermedia systems development, even though they appear to be ill-suited;
- to devise useful models and frameworks to enable more accurate time and cost estimation;
- to strive for better integration between tools and methods, so that methods are supported by useful tools, and tool usage is guided by useful methods.

## ACKNOWLEDGEMENT

I must thank Professor Brian Fitzgerald of the University of Limerick for his assistance with this study. Grant aid was received from the Millennium Research Fund of National University of Ireland, Galway.

## REFERENCES

- Aedo, I. & Díaz, P. (2001) Applying software engineering methods for hypermedia systems. *ACM SIGCSE Bulletin*. 33(3), 5-8.
- Barry, C. & Lang, M. (2001) A Survey of Multimedia and Web Development Techniques and Methodology Usage. *IEEE Multimedia*. 8(3) July-September, 52-60.
- Barry, C. & Lang, M. (2003) A Comparison of "Traditional" and Multimedia Information Systems Development Practices. *Information & Software Technology*. 45(4), 217-227.
- Blum, B. I. (1994) A Taxonomy of Software Development Methods. *Communications of the ACM*. 37(11) November, 82-94.
- Boehm, B. (1976) Software engineering. *IEEE Transactions on Computers*. 25(12), 1226-1241.
- Brinkkemper, S. (1996) Method engineering: engineering of information systems development methods and tools. *Information and Software Technology*. 38(4), 275-280.
- Britton, C., Jones, S., Myers, M. & Sharif, M. (1997) A survey of current practice in the development of multimedia systems. *Information & Software Technology*. 39(10), 695-705.

“CRISIS” OR “WONDERLAND”: AN APPRAISAL OF HYPERMEDIA SYSTEMS  
DEVELOPMENT PRACTICE

- Brooks, F. P. (1987) No Silver Bullet / Essence and Accidents of Software Engineering. *IEEE Computer*. 20(4) April, 10-18.
- Bush, V. (1945) As We May Think. *The Atlantic Monthly*. 176(1) July, 101-108.
- Coda, F., Ghezzi, C., Vigna, G. & Garzotto, F. (1998) Towards a Software Engineering Approach to Web Site Development. In *Proceedings of 9th International Workshop on Software Specification & Design*, Ise-Shima (Isobe), Japan, April 16-18, pp. 8-17. IEEE Computer Society Press.
- Constantine, L. L. & Lockwood, L. A. D. (2002) Usage-Centered Engineering for Web Applications. *IEEE Software*. 19(2) March-April, 42-50.
- De Troyer, O. (2001) Audience-Driven Web Design. In Rossi, M. & Siau, K. (eds), *Information Modeling in the New Millennium*, pp. 442-461. Hershey, PA: Idea Group Publishing.
- Dillman, D. A. (2000) *Mail and Internet Surveys: The Tailored Design Method*, 2nd Edition. New York: Wiley.
- Enguix, C. F. & Davis, J. G. (1999) Filling the Gap: New Models for Systematic Page-based Web Application Development and Maintenance. In *International Workshop on Web Engineering at 8th International World Wide Web Conference*, Toronto, Canada, May 11.
- Floyd, R. W. (1979) The Paradigms of Programming. *Communications of the ACM*. 22(8) August, 455-460.
- Galliers, R. D. (1992) Choosing Information Systems Research Approaches. In Galliers, R. D. (ed), *Information Systems Research: Issues, Methods and Practical Guidelines*, pp. 144-162. Oxford: Blackwell Scientific.
- Ginige, A. & Murugesan, S. (2001) Web Engineering: An Introduction. *IEEE Multimedia*. 8(1) January-March, 14-18.
- Glass, R. L. (1989) The Temporal Relationship Between Theory and Practice. *Journal of Systems and Software*. 10, 65-67.
- Glass, R. L. (1998) In Praise of Practice. *IEEE Software*. 15(1) January-February, 30-31.
- Greenbaum, J. & Stuedahl, D. (2000) Deadlines and Work Practices in New Media Development: Its about time. In Cherkasky, T. et al. (eds), *PDC 2000 Proceedings of Participatory Design Conference*, New York, USA, November 28 - December 1, pp. 70-77.
- Iivari, J., Hirschheim, R. & Klein, H. K. (1998) A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies. *Information Systems Research*. 9(2), 164-193.
- Lowe, D. & Hall, W. (1999) *Hypermedia & the Web: An Engineering Approach*. Chichester: Wiley.
- Lowe, D. B. & Eklund, J. (2002) Client Needs and the Design Process in Web Projects. *Journal of Web Engineering*. 1(1), 23-36.
- Murugesan, S. & Deshpande, Y. (1999) Preface to ICSE'99 Workshop on Web Engineering. In *Proceedings of 21st International Conference on Software Engineering (ICSE)*, Los Angeles, California, USA, pp. 693-694.
- Murugesan, S., Deshpande, Y., Hansen, S. & Ginige, A. (1999) Web Engineering: A New Discipline for Development of Web-based Systems. In *Proceedings of 1st ICSE Workshop on Web Engineering*, Los Angeles, California, USA, May 16-17, pp. 1-9. ACM Press.
- Narayanan, N. H. (1998) Hyped-media to hyper-media: toward theoretical foundations of design, use and evaluation. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems*, Los Angeles, California, USA, April 18-23, pp. 202-203. ACM Press.

- Naur, P. & Randell, B. (eds) (1969) *Software Engineering: Report on a Conference Sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 October 1968*. Brussels: Scientific Affairs Division, NATO.
- Norton, K. S. (1999) Applying Cross-Functional Evolutionary Methodologies to Web Development. In *Proceedings of 1st ICSE Workshop on Web Engineering*, Los Angeles, California, USA, May 16-17.
- Oinas-Kukkonen, H., Alatalo, T., Kaasila, J., Kivelä, H. & Sivunen, S. (2001) Requirements for Web Engineering Methodologies. In Rossi, M. & Siau, K. (eds), *Information Modeling in the New Millennium*, pp. 360-382. Hershey, PA: Idea Group Publishing.
- Pauen, P., Voss, J. & Six, H.-W. (1998) Modeling Hypermedia Applications with HyDev. In Sutcliffe, A. et al. (eds), *Proceedings of IFIP Working Group 13.2 Conference on Designing Effective and Usable Multimedia Systems, Stuttgart, Germany, September 9-11, 1998*, pp. 23-40. London: Kluwer Academic Publishers.
- Paul, R. J. (1994) Why Users Cannot 'Get What They Want'. *International Journal of Manufacturing Systems Design*. 1(4), 389-394.
- Pressman, R. S. (1997) *Software Engineering: A Practitioner's Approach (European Adaptation)*. New York: McGraw-Hill.
- Pressman, R. S. (2000) What a Tangled Web We Weave. *IEEE Software*. 17(1) January-February, 18-21.
- Reifer, D. (2002) Ten Deadly Risks in Internet and Intranet Software Development. *IEEE Software*. 19(2) March-April, 12-14.
- Retschitzegger, W. & Schwinger, W. (2000) Towards Modeling of DataWeb Applications: A Requirements Perspective. In *Proceedings of 6th Americas Conference on Information Systems (AMCIS)*, Long Beach, California, USA, August 10-13, Vol. 1, pp. 149-155.
- Rumbaugh, J. (1995) What is a method? *Journal of Object Oriented Programming*. 8(6) October, 10-16;26.
- Russo, N. L. & Graham, B. R. (1999) A First Step in Developing a Web Application Design Methodology: Understanding the Environment. In Wood-Harper, A. T. et al. (eds), *Methodologies for Developing and Managing Emerging Technology Based Information Systems: Proceedings of 6th International BCS Information Systems Methodologies Conference*, pp. 24-33. London: Springer.
- Siau, K. & Rossi, M. (2001) Information Modeling in the Internet Age - Challenges, Issues and Research Directions. In Rossi, M. & Siau, K. (eds), *Information Modeling in the New Millennium*, pp. 1-8. Hershey, PA: Idea Group Publishing.
- Thomas, D. (1998) Web Time Software Development. *Software Development Magazine*. October, 78-80.
- Vora, P. R. (1998) Designing for the Web: A Survey. *ACM interactions*. 5(3) May-June, 13-30.
- Wang, S. (2001) Toward a general model for web-based information systems. *International Journal of Information Management*. 21(5), 385-396.
- Ward, P. (1992) The evolution of structured analysis: Part II - maturity and its problems. *American Programmer*. 5(4), 18-29.
- Wynekoop, J. L. & Russo, N. L. (1995) Systems development methodologies: unanswered questions. *Journal of Information Technology*. 10(2) June, 65-73.
- Yourdon, E. (1996) Developing Applications for the Internet: Advice for the Java Generation. *American Programmer*. December, 36-41.