

NEW BRANCHES, OLD ROOTS: A STUDY OF METHODS AND TECHNIQUES IN WEB/HYPERMEDIA SYSTEMS DESIGN

Michael Lang and Brian Fitzgerald

Despite claims in the literature that “traditional” methods and techniques are inadequate and poorly matched to the demands of the Web design context, the findings from a field survey of Web/hypermedia systems designers in Ireland indicate quite the contrary. In general, the notion of “method” in the traditional sense seems to have been largely displaced by hybrid aggregations of techniques and other method fragments — selected on the basis of usefulness and purposefully blended within an overarching framework of an in-house development process.

MICHAEL LANG is a lecturer in information systems at National University of Ireland, Galway. His main interests are systems analysis and design, information systems education, and enterprise information systems.

BRIAN FITZGERALD holds the Frederick A. Krehbiel II Chair in Innovation in Global Business and Technology at the University of Limerick and is also a Science Foundation Ireland Investigator.

OF LATE, THERE HAS BEEN MUCH INTEREST in the design of Web-based systems. Unfortunately, “Web-based system” is a very loose term that, in its broadest sense, could be interpreted as embracing all applications that somehow rely on the World Wide Web as a platform for execution. Indeed, Lockwood and Constantine (1999) make the point that “current development tools make it easy to ‘browserize’ almost anything.” We take the position that it is only when Web-based systems assume hypermedia functionality (e.g., enhanced navigation and interaction mechanisms, highly visual interfaces, multimedia content, etc.) that they raise substantive challenges not traditionally experienced in “conventional” systems design (Lang, 2005). Therefore, in this article we use the term “Web/hypermedia systems” rather than “Web-based systems” in recognition

that interactive Web-based applications fall within the wider and more enduring concept of hypermedia.

It has been common throughout the history of computer-based systems design for avant-garde technologies to be hailed as profound advances that warrant entirely new design approaches. By the end of the 1990s, we saw the same reaction for Web/hypermedia systems. Murugesan et al. (1999) speak of “a pressing need for disciplined approaches and *new* methods and tools,” taking into account “the unique features of the new medium.” Other authors claim that methods and techniques from “traditional” or “conventional” systems design are inappropriate for Web/hypermedia design (Russo & Graham, 1999; Siau & Rossi, 2001; Oinas-Kukkonen et al., 2001). Accordingly, many new Web/hypermedia-specific methods and

TABLE 1 Size and Primary Business of Survey Respondents ($n=164^*$)

Primary Business	Organization Size (Number of Employees)					Total
	1–20	21–50	51–100	101–500	>500	
Web development	42	2				44 (27%)
IT/software development	9	4		4	5	22 (13%)
Graphic design/visual communications	22					22 (13%)
Multimedia development	13	1				14 (9%)
Management consultancy	4			2	3	9 (5%)
E-learning/CBT	4	3	2			9 (5%)
Financial services		1	1	1	6	9 (5%)
Public sector			1		6	7 (4%)
Traditional media	1	1		1	3	6 (4%)
Miscellaneous	15	4	2		1	22 (13%)
Total	110	16	6	8	24	164 (100%)

* One organization returned two responses; hence n is 164, not 165.

techniques have been proposed, an extensive list of which is included in Appendix A. On the other hand, dissenters argue that the design of Web/hypermedia systems does not present any fundamentally new or unique challenges (Kautz & Nørberg, 2003), whereas Howard et al. (1999) go so far as to assert, based on the result of an experiment that tested the efficacy of matching methods with application characteristics, that endeavors to devise specific methods for Web/hypermedia systems are not valuable because generic methods are likely to be just as effective.

Now that the hysteria of the dot.com era has subsided and Web design practices have matured, it is a timely juncture at which to reflect on the state of practice. The objective of this article is to explore *what processes, methods, and techniques are being used for Web/hypermedia systems design in practice*. This extends our previous work in this and related areas (Lang & Fitzgerald, 2005; Barry & Lang, 2001, 2003). The research method is described in Appendix B. As can be seen from Table 1, the majority of the respondents were from small organizations. Although these characteristics limit the generalizability of our findings, we feel our results provide significant insights about the evolution of methods and techniques for Web/hypermedia systems design.

FINDINGS

Respondent Groupings

The cover letter dispatched with the questionnaire requested that it be completed by someone in a design role — such as software design, information architecture, or graphic design — the rationale being to capture a random cross-section of respondents across the various disciplines

that contribute to Web/hypermedia systems design. An examination of these responses yielded the three separate groups of respondents identified below. We will use these groupings in our analyses to interpret some of the survey results.

- Those primarily from a computer-based systems development (CBSD) background (55 respondents; 33 percent of overall)
- Those primarily from a visual design (VD) background (43 respondents; 26 percent of overall)
- Those with similar degrees of knowledge of CBSD and VD, as well as miscellaneous, ambiguous, or missing responses (67 respondents; 41 percent of overall)

Design Teams

Consistent with the findings of previous studies of Web and multimedia design (Britton et al., 1997; Vora, 1998; Russo & Graham, 1999; Glass, 2001; Barry & Lang, 2001), we found that most teams are small. Only 7 percent of respondents indicated that they normally work in teams of more than ten developers, and in almost two-thirds of cases there are less than five team members. These findings also compare closely with those of a survey of software engineering practice conducted two decades ago by Beck and Perkins (1983), which is interesting because it is often assumed that the profile of modern systems development is altogether different from that of the so-called “structured methodologies era.”

Generally, problems with team management and communications are affected by team size. Because most teams in this study were small, it is understandable that team communications were found to present no or minor problems in

TABLE 2 Size of Development Teams (*n* = 164)

1 developer	6%
2 to 4 developers	57%
5 to 10 developers	31%
11 to 20 developers	5%
21 to 50 developers	1%
More than 50 developers	1%

77 percent of organizations. One respondent commented that:

Project management skills are the most lacking; keeping a team small is the best way to control the chaos.

Tests (Spearman's, Kendall's tau) were run to see if team size and the incidence of communication problems were correlated. No such correlation was found, but this was understandable given the heavily skewed distribution of the data; that is, there were very few cases of medium- to large-sized teams.

The literature suggests that as teams become larger, there is a greater need for formalized processes and procedures. Conversely, "lighter" methodologies may be better suited to small teams (Cockburn, 2000). The findings of this survey uphold this generalization. As team size increased, there seemed to be a greater propensity to use documented guidelines and procedures ($p < .01$; $r_s = .25$). Regarding the use of methods, one respondent commented that the in-house approach was a hybrid, tailored so as to be "small enough to be useful for a small company." Another remarked, "we work in small groups and this tends to obviate formal working methods."

Project Timeframes

It was found that 63 percent of projects are delivered in 16 weeks or less, with a typical delivery time on the order of three months (see Table 3). This contrasts with an early 1980s study of general systems development (Jenkins et al., 1984), which reported that the average project then lasted 10.5 months, and it also contrasts with the situation in the mid-1990s, when the duration of typical projects was about six months (Fitzgerald, 1997). Taking the findings of Table 2 and Table 3 alongside each other, it seems that the so-called "3x3" profile (Glass, 2001) typifies Web/hypermedia systems design in Ireland — teams of three or so designers working to deliver a project in about three months.

TABLE 3 Actual Project Duration (*n* = 138)

Less than 4 weeks	9 (7%)
4 to 8 weeks	29 (21%)
8 to 12 weeks	33 (24%)
12 to 16 weeks	16 (12%)
16 to 20 weeks	10 (7%)
20 to 30 weeks	21 (15%)
30 weeks or more	20 (15%)

Note: Median = 11.0 weeks;
5% trimmed mean = 14.4 weeks.

Requirements Management

There have been allegations in the literature that requirements definition in the traditional sense is often shoddily done or even bypassed completely in Web/hypermedia systems design practice (Coda et al., 1998; De Troyer, 2001; Murugesan et al., 1999). In view of such claims, it is interesting that 87 percent of respondents in our survey indicated that for their most recently completed project, there *was a written* requirements specification document. A statistically significant difference was found here between the CBSD and VD groups (Mann-Whitney test, $p < .01$). Almost all (94 percent, $n = 54$) of the respondents whose professional background was CBSD had produced a written specification document for their most recently completed project, whereas a smaller percentage (74 percent, $n = 42$) of the VD group had done so. This is perhaps reflective of traditional practices within these fields, because most traditional CBSD projects would have required that a requirements specification document be produced, whereas the design brief for many traditional VD projects (e.g., advertising campaigns, brand design) would have been quite loose and informal.

These specification documents are often quite detailed, with 53 percent of them being 25 or more pages long (see Table 4). They seem to be more substantial than mere sketchy tenders padded with sales pitch forematerials, because the 5 percent trimmed mean length is 41 pages.

In response to a separate question, 64 percent of respondents indicated that they use organizational guidelines to assist requirements documentation (Table 8). These findings suggest that requirements analysis practices are not, as alleged, opportunistic, but in fact are quite systematic. However, reading between the lines, we see another explanation here — by way of feedback to a summary report that

TABLE 4 Size of Written Requirements Specification ($n = 123$)

Less than 10 pages	22 (18%)
10 to 24 pages	36 (29%)
25 to 49 pages	25 (20%)
50 to 99 pages	20 (16%)
100 pages or more	20 (16%)

Note: Median = 26 pages;
5% trimmed mean = 41 pages.

was distributed to survey participants, one respondent pointed out that requirements specification documents may exist primarily as a form of defense:

Most new media development is still a marginal activity within businesses. You need signoff on a project to control creep, cost, and scheduling — but mostly to ensure that you've got a clear brief that you can defend.

As such, the primary purpose of the requirements specification may be as a contractual bargaining chip.

Design Processes and Project Plans

It has been frequently asserted in the literature that Web/hypermedia design practice is in a state of crisis, characterized by sloppy, "quick and dirty" activity (Murugesan et al., 1999; De Troyer, 2001). On the contrary, the findings of this survey suggest that Web/hypermedia design can be quite disciplined and systematic. In reply to a closed multiple-choice question, 83 percent of respondents indicated that their organization uses a development process that contains clear tasks or phases (Table 5).

Interestingly, only half of these organizations have an explicitly documented process. This is consistent with the view that systems design activity, on the face of it, may often appear to be laissez-faire or perhaps even somewhat chaotic, but beneath the surface it is guided by the expertise and tacit knowledge of the development team (Ciborra, 1999). Only 17 percent of organizations do not have a clear process. Of these, the majority (59 percent) regard the absence of a process as a problem.

Kruskal-Wallis tests were performed to check for differences in the level of formality of development processes across organizations of different sizes, organizations from different industrial sectors, and development teams of different sizes. No statistically significant differences were revealed for any of these three factors. However, when looking only at cases where there was a clear process (the bottom two rows of Table 5), we found ($p < .05$) that processes tend to be more formalized and explicit in Web development companies than in traditional IT/software development companies. A possible explanation for this is the sales-driven, high-speed nature of Web design, as illustrated by the following excerpt from a follow-up interview:

You have to streamline how you do things. You have to build processes, put them in place, and just follow them. It depends on the price as well, which is determined by how many days you spend on it. So if it costs X, you work out how many days you can spend on that job, and that's all you spend on it. So when a Web design project comes in, you know exactly what to do, you take it, and you go bang-bang-bang.

TABLE 5 Organizations' Web/Hypermedia Development Process (by Primary Business)*

	Web Development ($n = 43$)	IT/Software Development ($n = 22$)	Graphic Design/Visual Communication ($n = 22$)	Overall ($n = 163$)
There is no clear process	8 (19%)	4 (18%)	4 (18%)	27 (17%)
<i>Cleartasks and/or phases, but the process used is not explicitly documented</i>	11 (26%)	12 (55%)	10 (46%)	67 (41%)
<i>Cleartasks and/or phases, according to an explicitly documented process</i>	24 (56%)	6 (27%)	8 (36%)	69 (42%)

* Just the three main business sectors are shown, as well as the overall figures.

TABLE 6 Use of Methods and Approaches in Web/Hypermedia Systems Design

	Overall (n = 78)
Hybrid, customised, or proprietary in-house method or approach	18 (23%)
Traditional "legacy" software development methods and approaches, or variants (e.g., SSADM, Yourdon, JSP, SDLC/Waterfall)	17 (22%)
Rapid or agile development methods and approaches (e.g., RAD, eXtreme Programming)	12 (15%)
Approaches that are focused around the use of tools and development environments (e.g., PHP, Java, Flash, ASP, J2EE, InterDev)	11 (14%)
Incremental or evolutionary methods and approaches* (e.g., Spiral Model, RUP, Staged Delivery, Iterative Design)	10 (13%)
Object-oriented development methods and approaches (e.g., OOAD, UML, J2EE)	6 (8%)
No method used/development is ad hoc	6 (8%)
HCI/Human Factors Engineering methods (e.g., user-centered design, goal-based requirements)	4 (5%)
Technique-driven development (e.g., storyboarding, flowcharts, UML, prototyping)	5 (6%)
Specialized nonproprietary methods for Web/hypermedia systems design (e.g., Fusebox, WSDM, HDM)	4 (5%)

* Rational Unified Process (RUP) is not counted under "object-oriented development methods and approaches" because the RUP framework is not necessarily object oriented (e.g., see Evans, 2003).

The level of formality of development processes was found to be negatively correlated to the level of severity of problems raised by a number of selected development issues, suggesting that formalized processes and procedures can help reduce the incidence of such issues. The statistically significant findings were as follows: designing the user interface ($p < .001$; $r_s = -.28$); designing and visualizing information architecture/navigation structures ($p < .01$; $r_s = -.25$); mapping conceptual design schemas to actual physical implementation ($p < .05$; $r_s = -.20$); controlling and coordinating project tasks ($p < .05$; $r_s = -.17$); preparing accurate time and cost estimates ($p < .05$; $r_s = -.19$); coping with accelerated timescales of the Web environment ($p < .05$; $r_s = -.19$); lack of guidance in the use of design methods and techniques ($p < .05$; $r_s = -.21$); and lack of adequate design documentation ($p < .01$; $r_s = -.25$).

Design Approaches and Methods (Based on 78 Responses)

It is difficult to clearly define and distinguish between concepts such as "approach," "method," and "technique," because in the language of everyday practice these terms overlap to the point of almost being interchangeable. Variations in interpretation may in part explain why previous research on the use of methods and approaches in Web/hypermedia systems development has been somewhat inconsistent. Whereas Britton et al. (1997) found that "the 'big bang' approach to system development is

rare," and Barry and Lang (2001) tell of an eclectic mix of approaches in use drawn from a number of reference disciplines, Russo and Graham (1999) reported that none of their respondents used a formalized system development method.

Respondents were asked, in an open-ended free-form question, to "list the names of any hypermedia development methods or approaches that you have used." Ninety-three of the 165 respondents completed this question, but 15 of these were discarded as unusable. Readers should therefore be cautioned that this relatively low item response rate (56 percent, as opposed to an average of 97 percent for tick-the-box items) may in part be explained by *participants' difficulty in understanding what was meant by "methods or approaches."* The classified responses are shown in Table 6. Because the number of responses broken down under each of the various categories is small, it is not appropriate to conduct statistical analysis here, and the capacity to draw generalizable conclusions is therefore diminished.

Because some of the responses were imprecise, it was also not possible to accurately classify them all and some categories may be undercounted. For example, rapid/agile methods (15 percent), incremental/evolutionary approaches (13 percent), and object-oriented approaches (8 percent) do not feature to the extent that might be expected. This can be explained in a number of ways.

First, many respondents indicated that they were using in-house methods but provided little or no detail on the nature of these methods. It is reasonable to assume that quite a few of these would involve at least some component of rapid/agile, incremental/evolutionary, or object-oriented methods. Indeed, these three categories are not mutually exclusive, so respondents who indicated that they are using a rapid/agile (e.g., eXtreme Programming [XP] or rapid/agile development [RAD]) or an incremental/evolutionary (e.g., Rational Unified Process [RUP]) method might be, indeed probably are, also using object-oriented methods at some level. There is a thin line between rapid/agile and incremental/evolutionary: we counted RUP under incremental/evolutionary and XP under rapid/agile, although these two could arguably have been placed into either category. As it turns out, most respondents who are using RUP are also using some other form of rapid/agile method, but if those who are not were included under "Rapid or agile development methods and approaches," the count for that category would rise from 12 (15 percent) to 15 (19 percent). Similarly, RUP and Fusebox may be, but are not necessarily, object oriented. If these two methods were counted within the object-oriented category, the tally would rise from 6 (8 percent) to 11 (14 percent). However, we decided to adhere to the principle of prudence when tallying the categories, relying on hard facts rather than second guesses.

Second, a possible explanation for the low reported instance of object-oriented methods is provided by Iivari and Maansaari (1998):

[T]he use of "methods" may be at the level of approaches rather than of specific methods, possibly using relevant techniques to support the approach. To illustrate, a systems developer may apply all the basic principles of the OO approach, applying specific techniques such as CRCs, state transition diagrams, object interaction diagrams, etc., without specifically using any specific OO method.

In this study, it is notable that although only 8 percent of respondents indicated that they use an object-oriented method, a substantially greater proportion have used object-oriented techniques such as use-case diagrams (72 percent), class diagrams (62 percent), and state diagrams (50 percent), as shown in Table 7. This

TABLE 7 Use of Techniques for Conceptual Modeling of Web/Hypermedia Systems ($n = 165$)

Screen prototypes/mockups	159 (97%)
Flowcharts	154 (95%)
2D site mapping techniques	149 (91%)
Storyboards	140 (85%)
Entity-relationship diagrams	123 (74%)
Use-case diagrams/scenarios	118 (72%)
Object-oriented class diagrams	102 (62%)
3D site mapping techniques	86 (52%)
Statecharts/state diagrams	81 (50%)

seems to lend support to Iivari and Maansaari's proposition.

Third, some evidence suggests that the low overall usage of object-oriented methods is at least partly related to the respondents' backgrounds. Specifically, of those respondents from a VD background, not one mentioned an object-oriented method. Amongst the CBSD cohort, 5 of 39 (13 percent) listed the names of methods that could unequivocally be regarded as falling under the category of "object oriented." If RUP and Fusebox were also included, this would be 9 of 39 (23 percent), although this is still quite low.

Use of In-House Methods or Approaches. The top response category was in-house methods (23 percent), which is consistent with the findings of previous comparable studies (Vora, 1998; Barry & Lang, 2001). In the main, these were either proprietary methods or tailored hybrids. For those about which some detail was provided, they tended to consist of internal procedures built around HCI best practices and guidelines, or else were combinations of a number of public domain methods. Some of these were interesting and unusual, involving hybrids such as [SSADM+Yourdon+XP], [Waterfall+Spiral+Prototyping], and [RUP+XP]. In quite a few cases, respondents indicated that their in-house method was founded on research, experimentation, and experience, and where such experientially founded in-house methods are in place, they tend to be the only method being used. This suggests that Web/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 *distill the most useful elements into a home-cooked in-house approach*.

It is also likely that the classical SDLC is being used as a basic method "template" around which a development process and a set of techniques are built.

Use of Traditional Software Development Methods and Approaches. Given claims in the literature that traditional methods are ill suited to Web/hypermedia systems design (Russo & Graham, 1999; Siau & Rossi, 2001; Oinas-Kukkonen et al., 2001), it is of note that this was the second highest response category (22 percent). Most of these were derivatives of SSADM or SDLC/Waterfall, although a few also mentioned Yourdon or Jackson Structured Programming. However, notwithstanding their continued popularity, concerns about the suitability of traditional software development methods appear to be valid because, in response to a separate question, almost 39 percent of respondents were of the opinion that such approaches are impractical for Web/hypermedia systems design.

Oddly, of the 17 respondents who use traditional approaches, 5 (29 percent) were of the opinion that they are inappropriate for designing Web/hypermedia systems. This begs the question: why then do they use these methods? Possible explanations for this apparently dysfunctional behavior are provided by Fitzgerald (1996), who sets forth a number of covert as well as overt reasons for using systems development methods. Another plausible explanation is given by Barry and Lang (2001), who contend that where the SDLC/Waterfall model is being used for Web/hypermedia systems development, it is as a project management and pseudo-legal framework rather than an endorsement of any underlying philosophy.

It is also likely that the classical SDLC is being used as a basic method "template" around which a development process and a set of techniques are built. In a study of Web development practices in New Zealand, Paynter & Pearson (1998) found that "traditional methodologies are being adapted within organizations, to accommodate the needs of the WWW environment." Notwithstanding its long-acknowledged failings, the SDLC/Waterfall model has proven to be quite robust and resilient over time, and it has been around in some shape or form for at least the past 30 years. Powell et al. (1998, pp. 18-19) make this point:

The model that we believe is the most appropriate for Web development today is a form of the waterfall model. . . . The waterfall method tends to support the development of software that is published in distinct releases, while Web sites are often growing continuously and specifications often change with

the wind. A faster waterfall approach that allows rapid minor changes to the site within a larger general phased effort is more appropriate for the Web.

Indeed, it might be argued that RUP is an accelerated form of the Waterfall model, which facilitates rapid minor releases. A number of respondents did indeed indicate that they are using RUP, but these have been classified under the category of "incremental or evolutionary methods and approaches" (13 percent).

No Method Is Used or Development Is Ad Hoc. Six respondents (8 percent) indicated that they have no method or approach or that development is ad hoc. However, four of these six had earlier replied that they have a clear development process, and in three of these cases that process was explicitly documented. These same three also indicated in another question that their organization used documented procedures or guidelines for Web/hypermedia systems design.

This would tend to suggest that, although some respondents would not regard themselves as using constructs as grand or as concrete as a "method," they are nevertheless engaging some form of overarching mechanisms and principles to guide design activity. Of course, this apparent anomaly may be attributable to misunderstanding of the questions, but it is of note that Jonasson (2002) reached a similar conclusion in her study of Web-based multimedia information systems development:

When it comes to methodology use, most respondents claim that there is no specific comprehensive methodology applied in their organization. . . . As the interviews progressed, however, a notion of a structured way of working appeared.

"Structure" in this sense does not necessarily mean "structured analysis/structured design" in the old-fashioned rigid sense of the methodologies era. A "structured way of working" may in fact be quite lean and agile, simply an understood way of doing things within the design team so that work moves as efficiently and smoothly as possible.

Use of Tool-Driven Design Approaches. There is substantial incidence of development approaches that are focused around the use of tools (14 percent), a finding that lends some support to the assertion that developers "dive directly into the implementation phase" (Coda

In the modern context of rapid development, an approach that is focused on the use of tools may actually be pragmatic, not misguided.

et al., 1998). Within the VD group, more so than for the CBSD group, approaches seem to be primarily driven by the particular development tools being used (e.g., Flash, Active Server Pages). Tool-driven approaches have traditionally been frowned on within the software development literature, where thorough up-front analysis and design is emphasized rather than cutting “straight to implementation.” However, in the modern context of rapid development, an approach that is focused on the use of tools may actually be pragmatic, not misguided. After all, tools are means to get the job done. Different tools can promote and encourage different styles of working. It therefore seems reasonable that organizations may choose to invest in a suite of rapid development tools and then to build working methods around those tools. Indeed, this may be easier than selecting a method and then trying to plug development tools into that method. An example of a Web design method that has grown up around a tool (ColdFusion) is Fusebox.

Use of Specialized Web/Hypermedia Development Methods. As further testimony to this latter point, it is notable that *there is very little usage of Web/hypermedia-specific methods*, as was also found in previous studies (Rodriguez-Garcia & Harrison, 2000; Barry & Lang, 2001; Safieddine, 2003). Although there are many such methods set forth in the academic literature (see Appendix A), the findings of this survey reveal that just 4 of 78 respondents have ever used any of them. A possible reason for this is that there is very little published guidance on how to implement these Web/hypermedia-specific methods using industry-standard development tools, nor is it obvious how such methods could be implemented using these tools. Significantly, the most widely used Web/hypermedia-specific method (Fusebox) has been devised by a community of practitioners rather than academics, and books are available that illustrate how it may be implemented using ColdFusion, PHP, J2EE, and Active Server Pages.

Most of the methods listed in Appendix A are documented only in short papers, so there is a dearth of well-illustrated examples to guide their application in practice. These papers are mainly published in academic outlets and are thus unlikely to be read by practitioners. It is only in recent years that specialized third-level courses in Web development and interactive digital multimedia design have been introduced by Irish universities, so even where

these academic Web/hypermedia design methods form part of the curriculum, it will be some time before the results of teaching filter into practice. It is therefore to be expected that there is a general lack of awareness of academic Web/hypermedia design methods. Other than the four respondents who have used a Web/hypermedia-specific method, only another four were otherwise aware of any such method. This lack of awareness is an obvious factor contributing to the low levels of method usage.

Even if levels of awareness were to rise, it has historically been the case that the software development community is slow to change and to an extent may be said to be “stuck to old practices” (Iivari & Maansaari, 1998; Adams & Avison, 2003). It was found in this survey and elsewhere (Barry & Lang, 2001; Rodriguez-Garcia & Harrison, 2000; Safieddine, 2003) both that newer methods and techniques (e.g., OOHDM, WSDM, Unified Modeling Language [UML]) are used very little and that “legacy” methods and techniques (e.g., SDLC, DFDs) are being used widely. Inertia may therefore be an explanatory factor, because developers seem to be reluctant to abandon older methods and techniques even when their usefulness and applicability is questionable.

Other major issues that affect the use of Web/hypermedia design methods are understandability, ease of use, and widespread acceptance and reputation among developers (Barry & Lang, 2001). Henderson-Sellers (1995) makes the point that an important component of an “appropriate” systems development method should be “a workable notation, ideally supported by third-party drawing tools.” Many Web/hypermedia design methods, such as RMM and OOHDM, are quite complex and difficult to grasp, use proprietary formalized diagrammatic notations that call for specialized training, and are not well supported by third-party drawing tools. The only academically produced Web/hypermedia design method that is used to any extent in practice is WSDM (De Troyer, 2001). It is perhaps no coincidence that WSDM happens to be an intuitively sensible and easy-to-learn method that is founded on the principles of user-centered design and uses well-known notations such as use-case diagrams, UML class diagrams, and ERDs.

Finally, to state an obvious explanation for nonuse, it may well be many academically produced Web/hypermedia design methods are simply impractical and not suited to the needs of practice. This point is supported by Safieddine’s (2003) conclusion that:

The areas in which traditional software diagramming techniques are argued to be most lacking are navigation design, user interface design, multimedia modeling, and specification of security.

As respondents build experience and are exposed to more Web modeling languages, for variety of reasons respondents are convinced that they would be better using their own methods or none at all. This research can only suggest that the possible reason could well be the fact that these models do not meet the market's expectations and need.

Use of Modeling Techniques (Based on 165 Responses)

Respondents were presented with a list of modeling techniques and asked to indicate which ones they use. The listed techniques were those that we thought most likely to be employed for data/content modeling and interaction modeling in Web/hypermedia design. In retrospect, it was an omission not to have included some process modeling techniques as well. Granted, respondents were allowed to append other techniques that did not feature in the pre-specified list, but the item response rate for this "specify other" category was too low to be meaningful.

Table 7 shows the extent to which modeling techniques have at some stage been used by respondents. Care should be taken in interpreting the table because of the imprecise nature of informal and semiformal techniques (e.g., the fuzzy boundaries between a storyboard, a flowchart, and a 2D site mapping technique). We should also explain that data flow diagrams did not explicitly feature in the list of techniques. Although no respondents indicated under the "specify other" category that they use data flow diagrams, they are notable by their absence from the findings in Table 7 and it seems likely that a flaw in our questionnaire design has inadvertently introduced a bias here that is somewhat misleading.

The most widely used diagramming techniques are screen prototypes/mockups, flowcharts, 2D site maps, and storyboards, which is consistent with the findings of previous studies (Britton et al., 1997; Rodriguez-Garcia & Harrison, 2000; Barry & Lang, 2001; Taylor et al., 2002). Statecharts were again found to be at the bottom end of the ranking, but the level of usage of entity-relationship diagrams, use-case diagrams, and class diagrams is higher in this survey than in those previous studies.

Numerous authors suggest that traditional software diagramming techniques are not readily transferable to Web/hypermedia design and that some aspects of Web/hypermedia

design are not considered by traditional techniques (Retschitzegger & Schwinger, 2000; Rossi & Schwabe, 2001; Siau & Rossi, 2001). On the other hand, others argue that traditional software design techniques still apply (Pressman, 2000; Constantine & Lockwood, 2002). The areas in which traditional software diagramming techniques are argued to be most lacking are navigation design, user interface design, multimedia modeling, and specification of security. All of these aspects can be modeled to some extent using one or more of the techniques in Table 7 — most obviously flowcharts, storyboards, use-case diagrams, class diagrams, and statecharts — and it is notable that considerable use is indeed being made of all these techniques, although less so of the latter.

Of course, all of the aforementioned aspects can also be modeled using UML and its Web/hypermedia-specific extensions (Bau-meister et al., 1999; Conallen, 2000). However, the findings here indicate that very little use is being made of UML in Web/hypermedia systems design, with the exception of class diagrams and use-case diagrams. There are at least two possible explanations for this.

First, Web/hypermedia systems design teams are multidisciplinary and comprise members from a variety of different backgrounds, many of whom are not trained in the use of software specification techniques such as UML. In a previous study, Carstensen and Vogelsang (2001) "observed that the designers and developers had severe problems in understanding each other. They were not able to read each others' diagrams and specifications." This might explain why participants in this survey seem to prefer informal diagramming techniques, because although these techniques are not as precise as formalized techniques such as UML, they are more generally understandable.

Second, most of the respondents to this survey appear not to be involved in the design of mission-critical systems, given that only 40 percent have procedures in place for testing and debugging (Table 8). One would expect the use of UML and other formalized, verifiable techniques to be higher where systems have critical components.

Just as inertia might be said to be a reason why traditional methods and approaches still dominate, the same explanation may also be offered as to why traditional techniques such as entity-relationship diagrams are more popular than newer ones such as class diagrams and UML. Adams and Avison (2003) put forward the view that:

TABLE 8 Use of Documented Procedures and Guidelines ($n=165$)

Requirements documentation	64%
Project planning and estimation	64%
Interface design/usability	52%
Technical design documentation	51%
System testing and debugging	40%
Coding practices	36%
Use of diagramming techniques	25%

Habits “learned” using previous techniques and problems would bias the application of new techniques and problems. This could explain the dominance of certain techniques used in IS development, such as dataflow diagrams and entity-relationship models. It may also explain why so many “new” techniques are rehashes of older ones. It might also explain why “newer” techniques take so long to get established.

Use of Documented Procedures and Guidelines (Based on 165 Responses)

Most respondents (79 percent) *feel* that plans and working methods should be explicitly documented; 113 of 165 respondents (69 percent) *actually* use documented guidelines or procedures for some purpose (Table 8). Taken together, these findings are suggestive of a broadly favorable disposition among practitioners, in principle at least, toward the use of systematic, streamlined ways of performing certain activities. This usage level is higher than in previous comparable studies by Russo and Graham (1999) and Taylor et al. (2002), where just 32 percent and 52 percent, respectively, of respondents said they were using formalized standards and guidelines, but less than reported by Vora (1998), who found that 90 percent of organizations were using Web design guidelines.

Percentages are based on overall responses, not just those who have documented procedures and guidelines in place (i.e., 165, not 113).

Of the 113 respondents in whose organizations documented guidelines or procedures are used, 73 (65 percent) consider them to be very useful. Given that responses to a previous question revealed that organizational policies and procedures are the least useful sources of knowledge on design methods and techniques, here is an apparent contradiction in the data. This may be explained in two ways. First, these

procedures and guidelines relate not just to design methods and techniques but also to other aspects such as project planning, interface design, and system testing (see Table 8). Second, when asked how prescriptive procedures and guidelines are, 80 percent of respondents whose organizations have them in place indicated that they are moderate and are only stringent in 10 percent of cases. The role of procedures and guidelines therefore seems not to be an attempt to codify design knowledge but rather to serve as a checklist or high-level roadmap.

CONCLUSION

Wynekoop and 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.” Although a host of Web/hypermedia-specific methods have been produced (see Appendix A), with the exceptions of WSDM and Fusebox, we found that few are being used in practice. Instead, hybrid adaptations of time-proven traditional approaches are emerging, tailored to the specific challenges of Web/hypermedia development projects. Hence, the metaphor “new branches, old roots” is appropriate.

This suggests several potentially useful directions for further research:

- To produce guidance on how to adapt, apply, and combine existing techniques from software development, industrial design, graphic design, and other reference disciplines to the domain of Web/hypermedia
- To investigate the nature of hybrid and in-house Web/hypermedia design 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 Web/hypermedia systems design, even though a substantial cohort, including some who actually use such methods, consider them somewhat impractical
- To pursue better integration between design tools and design methods ▲

References

- Adams, C. & Avison, D. (2003) Dangers Inherent in the Use of Techniques: Identifying Framing Influences. *Information Technology & People*, 16(2), 203-234.

- 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 and Software Technology*, 45(4), 217-227.
- Baumeister, H., Koch, N. & Mandel, L. (1999) Towards a UML Extension for Hypermedia Design. In France, R. B. & Rumpe, B. (eds), *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, Vol. 1723, pp. 614-629. Springer.
- Beck, L. L. & Perkins, T. E. (1983) A Survey of Software Engineering Practice: Tools, Methods and Results. *IEEE Transactions on Software Engineering*, 9(5) September, 541-555.
- Britton, C., Jones, S., Myers, M. & Sharif, M. (1997) A Survey of Current Practice in the Development of Multimedia Systems. *Information and Software Technology*, 39(10), 695-705.
- Carstensen, P. H. & Vogelsang, L. (2001) 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, pp. 536-547.
- Ciborra, C. U. (1999) A Theory of Information Systems Based on Improvisation. In Currie, W. L. & Galliers, B. (eds), *Rethinking Management Information Systems*. pp. 136-155. Oxford University Press.
- Cockburn, A. (2000) Selecting a Project's Methodology. *IEEE Software*, 17(4) July-August, 64-71.
- 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.
- Conallen, J. (2000) *Building Web Applications with UML*. Reading, MA: Addison Wesley.
- 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.
- Evans, G. (2003) Agile RUP for Non-Object-Oriented Projects. *The Rational Edge*, September.
- Fitzgerald, B. (1996) Formalised Systems Development Methodologies: A Critical Perspective. *Information Systems Journal*, 6(1), 3-23.
- Fitzgerald, B. (1997) The Use of Systems Development Methodologies in Practice: A Field Study. *Information Systems Journal*, 7(3), 201-212.
- Glass, R. L. (2001) Who's Right in the Web Development Debate? *Cutter IT Journal*, 14(7) July, 6-10.
- Henderson-Sellers, B. (1995) Who Needs an Object Oriented Methodology Anyway? *Journal of Object Oriented Programming*, 8(6), 6-8.
- Howard, G. S., Bodnovich, T., Janicki, T., Liegle, J., Klein, S., Albert, P. & Cannon, D. (1999) The Efficacy of Matching Information Systems Development Methodologies with Application Characteristics: An Empirical Study. *Journal of Systems and Software*, 45(3) March, 177-195.
- Iivari, J. & Maansara, J. (1998) The Usage of Systems Development Methods: Are We Stuck to Old Practices? *Information and Software Technology*, 40, 501-510.
- Jenkins, M. A., Naumann, J. D. & Wetherbe, J. C. (1984) Empirical Investigation of Systems Development Practices and Results. *Information & Management*, 7(2), 73-82.
- Jonasson, I. (2002) Trends in Developing Web-Based Multimedia Information Systems. In Kirikova, M. et al., (eds), *Information Systems Development: Advances in Methodologies, Components and Management*. pp. 79-85. New York: Kluwer Academic.
- Kautz, K. & Nørkjerg, J. (2003) Persistent Problems in Information Systems Development: The Case of the World Wide Web. In Ciborra, C. et al. (eds), *Proceedings of 11th European Conference on Information Systems (ECIS)*, Naples, Italy, June 16-21.
- Lang, M. (2005) A Critical Review of Challenges in Hypermedia Systems Development. In Vasilecas, O. et al., (eds), *Information Systems Development Advances in Theory, Practice and Education. 13th International Conference on Information Systems Development, ISD'2004 Vilnius, Lithuania, September 2004, Proceedings*. Heidelberg: Springer.
- Lang, M. & Fitzgerald, B. (2005) Hypermedia Systems Development Practice: A Survey. *IEEE Software*, 20(2) March-April, 68-75.
- Lockwood, L. & Constantine, L. (1999) Taming Web Development. *Software Development Magazine*, 7 April, 78-80.
- 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.
- 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. 442-461. Hershey, PA: Idea Group Publishing.

- Millennium.* pp. 360-382. Hershey, PA: Idea Group Publishing.
- Paynter, J. & Pearson, M. (1998) *A Case Study of the Web-Based Information Systems Development*. Department of Management Science and Information Systems, University of Auckland, New Zealand. http://www.cecil.edu/reports/www_Case_Multimedia_98.pdf.
- Powell, T. A., Jones, D. L. & Cutts, D. C. (1998) *Web Site Engineering: Beyond Web Page Design*. Upper Saddle River, NJ: Prentice Hall.
- Pressman, R. S. (2000) What a Tangled Web We Weave. *IEEE Software*, 17(1) January–February, 18–21.
- Retschitzegger, W. & Schwaiger, 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.
- Rodriguez-Garcia, D. & Harrison, R. (2000) Practitioners Views on Web Development: An Industrial Survey by Semi-Structured Interviews. In *Proceedings of 13th International Conference on Software and Systems Engineering and their Applications (ICSSEA 2000)*, Paris, France, December 5–8. CNAM.
- Rossi, G. & Schwabe, D. (2001) Object-Oriented Web Applications Modeling. In Rossi, M. & Siau, K. (eds), *Information Modeling in the New Millennium*. pp. 463–483. Hershey, PA: Idea Group Publishing.
- 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: 6th International BCS Information Systems Methodologies Conference*. pp. 24–33. London: Springer.
- Safieddine, F. (2003) *Survey Summary (Ph.D. Research)*. University of East London.
- 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.
- Taylor, M. J., McWilliam, J., Forsyth, H. & Wade, S. (2002) Methodologies and Website Development: A Survey of Practice. *Information and Software Technology*, 44(6), 381–391.
- Vora, P. (1998) Designing for the Web: A Survey. *ACM interactions*, 5(3) May–June, 13–30.
- Wynekoop, J. L. & Russo, N. L. (1995) Systems Development Methodologies: Unanswered Questions. *Journal of Information Technology*, 10(2) June, 65–73.

APPENDIX A: FORMALIZED METHODS AND TECHNIQUES FOR WEB/HYPERMEDIA DESIGN

Some of the following methods and techniques have not been given names by their authors. For all such cases, a selection of keywords from the published source is listed and the author's name is provided. Where a method has been extended, its successors are shown by means of an inheritance arrow →. A full bibliography of Web/hypermedia design methods can be found at http://www.nuigalway.ie/bis/mlang/Web_design_methods.pdf.

- Analysis and design of Web-based information systems (Takahashi & Liang)
- Araneus
- Ariadne/Labyrinth
- AutoWeb/HDM-Lite/Visual HDM
- Component-based methodology for Web application development (Lee & Shirani)
- Conceptual design of WWW-based information systems (Becker et al.)
- Conceptual modeling of large Web sites (Strauch & Winter)
- Dreher Hypertext Development Methodology
- Enhanced Object-Relationship Model (EORM)
- Fusebox/FLIP
- HANDIE
- Hera
- Human Factors methodology for designing Web sites (Vora)
- HyDev
- Hypermedia design methodology for Web applications (Moreno-Muñoz et al.)
- Hypermedia Flexible Process Modeling (HFPM)
- Hypermedia Modeling Technique (HMT)
- Hypertext design method (Morris & Finkelstein)
- Hypertext Design Method (HDM) HDM2 HDM2000/W2000 (Garzotto, Paolini, Schwabe et al.) (see also AutoWeb and WebML)
- Hypertext design methodology (Kemp & Buckner)
- Hypertext development methodology (van Vliet & Wilson)
- Index-driven Hypermedia Design Methodology (IHDM)
- Integrated design methodology for Internet-based information systems (de Lima & Price)
- Intranet Design Methodology (IDM)
- Internet Commerce Development Methodology (ICDM)
- Maintainable, End user friendly, Structured Hypermedia (MESH)

- Methodology for building corporate Web applications (Artz)
- Mockup-driven Fast Prototyping Methodology (MODFM)
- Navigational Development Technique (NDT)
- Object-Oriented Hypermedia Design Methodology (OOHDM)
- Object-oriented Process, Environment and Notation (OPEN)
- Object-Oriented-Hypermedia Method (OO-HMethod)
- Object-Process Methodology (OPM/Web)
- Parallel
- PlumbingXJ
- Relationship Management Methodology (RMM) Extended RMM
- Relationship-Navigation Analysis (R-NA)
- Role-driven Component-Oriented Methodology (RCOM)
- Scenario-based Object-oriented Hypermedia Design Methodology (SODHM)
- Simple Web Method (SWM)
- UML-based Web Engineering (UWE)
- Unified Process-based Hypermedia Systems Development (UPHD)
- User-based design process for Web sites (Abels, White, & Hahn)
- User goal-centered Web engineering approach
- View-based Hypermedia Design Methodology (VHDM)
- Web Information System Development Methodology (WISDM)
- Web Site Design Method (WSDM)
- Web-based Information Systems Development with a Comprehensive Methodology (WISDOM)
- WebComposition / WCML
- WebMLWebML+/Web Information Exchange Diagram (WIED)
- Workflow-Based Hypermedia Development Methodology (WHDM)
- World Wide Web/Structured Hypermedia Design Technique (W3DT/SHDT) eW3DT

APPENDIX B: SURVEY METHODS

A dual-mode (postal and Web) survey was conducted of Web/hypermedia systems designers in Ireland. The sampling frame included (1) organizations engaged in bespoke systems development; (2) those specializing in Web, interactive multimedia, or hypermedia systems development; (3) those from traditional media that have branched into "new media"; and (4) those that have internal IS departments (e.g., financial services firms and banks). An initial list was compiled from a number of classified industry databases, which was then systematically filtered, leaving an eventual sample of 438 organizations.

After two follow-up rounds, 167 usable responses were received, but two of these were discarded because the respondents had experience developing only trivial "brochureware" systems. The usable response rate was therefore 45 percent, based on the size of the true population (i.e., setting aside those who had gone out of business or who responded that they do not engage in Web/hypermedia systems design). The size and primary business of the respondents are reported in Table 5.

Copyright of Information Systems Management is the property of Auerbach Publications Inc. and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.