# Hypermedia systems development: a comparative study of software engineers and graphic designers

Communications of the Association for Information Systems

# HYPERMEDIA SYSTEMS DEVELOPMENT: A COMPARATIVE STUDY OF SOFTWARE ENGINEERS AND GRAPHIC DESIGNERS

**MICHAEL LANG**
*Department of Accountancy & Finance*
*National University of Ireland, Galway*
MICHAEL.LANG@NUIGALWAY.IE

## ABSTRACT

Hypermedia systems development is, in many regards, different from "conventional" systems development, chief amongst these differences being its multidisciplinary nature. Foremost amongst the roles in hypermedia development are software engineering and graphic design. However, there has traditionally been a pronounced tension between software engineers and graphic designers. It is therefore important to gain an understanding of the differences between the two camps with a view to bringing them closer together. This paper reports on the findings of a survey of hypermedia developers conducted in Ireland. One of the objectives of the survey was to compare and contrast the development approaches, methods, and techniques used by software engineers with those used by graphic designers. It was found that software engineers and graphic designers are much closer than might be believed in their attitudes on the value and importance of processes and documented working methods. However, graphic designers primarily base development approaches around the use of specific tools, whereas software engineers are more reliant on traditional and object-oriented software development methods. Regarding diagramming methods, there is some evidence of cross-pollination, as software engineers often use informal techniques such as storyboarding and graphic designers use software engineering techniques such as use case diagrams. However, graphic designers find software engineering techniques to be less useful than vice versa.

**Keywords:** hypermedia, systems development methods, systems development techniques.

## I. INTRODUCTION

As envisioned by Berners-Lee [1996], the Web is "a universe of global network-accessible information…chiefly populated by interlinked pages of text, images, and animations, with occasional sounds, videos, and three-dimensional worlds" Otherwise put, a world-wide communication and information system founded upon rich hypermedia technology. Although the Web in its current form is a primitive low-level hypermedia environment [Nürnberg and Ashman, 1999], it is nevertheless the most common platform for implementing hypermedia systems. Therefore, when studying methods, techniques, and approaches for systems design, interactive Web-based systems should properly be considered within the broader classification of hypermedia systems.

"Hypermedia" is one of those unfortunate terms that is easier to recognize in actuality than it is to define precisely in theory. In his introductory text on the principles of logic, Luce [1958] states that:

> "*Definition is a difficult art,…especially when unfamiliar or ambiguous terms have to be employed…The perfect definition is an ideal … [hence] weak definitions, and even mere descriptions, can be better than nothing*".

In view of such inherent limitations, hypermedia is arbitrarily defined for the purposes of this study as

> "any interactive software system that permits a user to navigate through hyperlinked information by means of various user-selected paths, including such applications as interactive Web sites, electronic catalogues, intranets, courseware/CBT, interactive e-commerce systems, portals, and online information services".

This definition recognizes that hypermedia technologies include but also pre-date the Web, and the traditions of hypermedia development are longer than usually acknowledged.

Notwithstanding these traditions, hypermedia systems development is in many regards different from "conventional" systems development. Chief amongst these differences is its multidisciplinary nature. Development teams typically involve a multiplicity of roles, foremost of which are software engineering and graphic design [Barry and Lang, 2003]. The involvement of graphic designers is essential because hypermedia systems development must consider vital aesthetic and cognitive aspects that are not covered by traditional software engineering approaches [Nanard and Nanard, 1995]. The meeting of graphic design and software engineering is a consequence of traditional communications media becoming dynamic and interactive, and traditional software development becoming visual and graphically-intensive. However, this union has not been an amicable one; rather, more like two unwilling partners awkwardly engaging in an unlikely marriage. Typically the tension between software engineers and graphic designers is pronounced because their values conflict and "*appear to operate in distinctly different worlds*" [Vertelney et al., 1990].

Gallagher & Webb [1997] observed an apparent dichotomy:

-software engineering is structured, takes a logical view, emphasizes functionality, and works outwards from the interior,

-graphic design takes a "hacker" approach, concentrates on the creative design of graphic interfaces, is more user-centered, and works inwards from the exterior.

A popular view amongst software engineers is that graphic design is a purely creative and fuzzy discipline, exemplified by assertions such as

> "*the front end is fluffy … in some sense it doesn't matter what tool you use to produce it … but everything the front end talks to is serious engineering*" [Pressman et al., 1998].

The work of graphic designers is often regarded with condescension, their brief being to "*make things look pretty after they are made to work"* [Vertelney et al., 1990].

Conversely, graphic designers typically regard software engineers as being rigidly logical and pedantically insistent upon such concerns as "*functionalism, modularity and maintainability*" [Gallagher and Webb, 1997]; they are "'*feature-freaks' who could care less how a thing looks as long as the code is elegant*" [Vertelney et al., 1990]. Indeed, software engineers and graphic designers have traditionally followed quite different vocational training paths with little if any overlap in curricula.

Given the paramount roles played by both software engineers and graphic designers in hypermedia systems development, it is important to gain an understanding of the differences between the two camps with a view to bringing them closer together. This paper reports on the findings of a survey that investigates hypermedia development practices. Amongst the objectives of this investigation were to compare and contrast:

- development approaches and methods used by software engineers with those used by graphic designers;
- attitudes towards planning, considered action, and documented working methods;
- attitudes towards the use of diagramming techniques, techniques actually used, and the perceived usefulness of those techniques.

Although a number of studies report on hypermedia development practice, and related areas such as Web and multimedia design [Barry and Lang, 2001, 2003, Baskerville and Pries-Heye, 2001, Britton et al., 1997, Carstensen and Vogelsang, 2001, Eriksen, 2000, Liu et al., 1998, Lowe and Eklund, 2002, Russo and Graham, 1999, Taylor et al., 2002, Vora, 1998], very little empirical research thus far deals with how graphic designers and software engineers compare. To the author's knowledge, only one such substantive study was published previously [Gallagher and Webb, 1997], although others allude to it [Newman and Landay, 2000, Whitley, 1998]. The study reported here is based on a wide-scale survey, whereas the work of Gallagher and Webb was based primarily on interviews and a literature survey.

## II. RESEARCH METHOD

In the last quarter of 2002, a survey of hypermedia development organizations was conducted across all Ireland. Participants were given the option of responding by traditional mail or via the Web. For both modes, appropriate authentication mechanisms were engaged to assure instrumental rigor and validity. The population of interest included organizations engaged in general software development; those specializing in Web, multimedia, or hypermedia systems development; those from traditional media that branched into "new media"; and those with internal IS departments. An initial list of organizations was compiled from a number of industry databases and then filtered systematically, – based on descriptions of activities and portfolios of work as described on Web sites and in secondary data sources, – so as to only include those who developed, or were likely to have developed, hypermedia systems as defined in Section I. The final population consisted of 438 organizations. It was decided to include the entire population in the sample because it was feasible to do so, thus effectively eliminating sampling error.

Prior to its 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]. A number of revisions were implemented iteratively. In addition, the services of professional technical writers were engaged to assist with the wording and visual layout of the questionnaire. No problems with the definition of hypermedia were experienced during pilot testing, and pilot testers were familiar and comfortable with the notion of "hypermedia". Indeed, a few commented that it was entirely appropriate that research into Web design should consider the legacy of previous generations of hypermedia systems such as interactive CD-ROMs, on-line help systems, and even library systems.

Reminders were sent out by post after 4 weeks, and again by e-mail after 6 weeks. The cumulative effect was to double the overall response rate. In total, responses were received from 213 organizations. Of these, 42 indicated no significant experience of hypermedia systems design, and 5 were insufficiently complete. A further 23 questionnaires were returned undelivered or with a note that the organization ceased operations. Thus, the overall response rate was 45.8% (171/373) and the usable response rate based on the true size of the relevant population was 44.5% (166/373)[1]. On average, respondents reported about 5 years experience of developing hypermedia systems.

The cover letter requested that the questionnaire be completed by someone in a design role, - such as software design, information architecture, or graphic design, - the rationale being to

[1] The population of 438 was reduced to 373 (438 - 23 shutdowns - 42 inappropriate) and the number of usable responses from 213 to 166 (213 - 5 incomplete - 42 inappropriate).

capture a random cross-section of respondents across the various disciplines that contribute to hypermedia systems development. Respondents were asked to indicate their professional discipline (open-ended), and in a separate question to grade their knowledge in each of a variety of listed disciplines. On examination of these responses, three separate groups were identified:

- Group 1 ("SE"): Those primarily from a software development background (e.g. software engineering, systems analysis, computer programming) who have substantially less knowledge of other relevant disciplines (55 respondents; 33.1% of overall).
- Group 2 ("GD"): Those primarily from a graphic design background, who have substantially less knowledge of other relevant disciplines (44 respondents; 26.5% of overall)
- Group 3: Those with similar degrees of proficiency in software development and graphic design, as well as miscellaneous, ambiguous, or missing responses. Many of the respondents in this category described themselves as "information architects", "Web developers" or "Web designers" (67 respondents; 40.4% of overall).

That the largest cohort of respondents fall into Group 3 is evidence of the multi-disciplinary nature of hypermedia systems development, and indicates that the gap between software engineering and graphic design is not a void but is in fact mostly populated by individuals who are competent in both disciplines. While only 1 (1.5%) of those in Group 3 claimed to have a high level of knowledge in both graphic design and software engineering, another 44 (67.7%) professed at least an intermediate level of knowledge in both, and a further 12 (18.4%) claimed at least an intermediate level of knowledge in one or the other[2]. However, the concentration in this paper shall be on a comparison of Group 1 and Group 2. Group 3 was excluded from the comparison because, unlike the other two groups, it is not homogenous in its composition and therefore probably does not contain a prevailing culture or common set of values.

Some are of the view that graphic designers are mostly involved in lightweight work of trivial complexity, as opposed to "real" development as done by software engineers. If this were true, a comparison of approaches, methods, and techniques used by both groups would not be very meaningful because the difficulty levels of tasks undertaken would not be equivalent. However, though it may be offensive to the pride of software engineers to utter it, the task of developing interactive software systems became demystified, to an extent, over recent years. By the availability of content management tools, commercial off-the-shelf (COTS) products, open source applications software[3], pre-fabricated components and applets, fifth generation visual programming interfaces, and back-end applications hosted by service providers, graphic designers are now empowered to develop quite sophisticated hypermedia systems without needing to learn programming or software engineering skills. This change is the realization of end-user computing as software development is no longer the elite domain of trained specialists.

To test for equivalence in the complexity of projects worked upon by software engineers and graphic designers, a Mann-Whitney comparison was performed across a number of variables. These comparisons measured the average team size, the length and detail of the requirements specification for the most recently delivered project, the average number of pages/screens in an application, and the number of weeks taken to deliver the most recent project. Overall, the complexity of projects worked upon by both groups seems comparable. The only statistically

[2] Only 65 of the 67 members in Group 3 responded to this question i.e. 2 missing responses. Percentages are computed out of 65.

[3] The term "open source software" conjures up images in many people's minds of operating systems (e.g. Linux, BSD) or server applications (e.g. Apache, MySQL, PHP). I prefer to call these and other such examples "open source *systems* software" as their use requires advanced technical knowledge of how to configure computer systems. To differentiate, I reserve the term "open source *applications* software" to refer to packages (e.g. osCommerce) that may be easily installed and customized by graphic designers and others with reasonable general knowledge of using computers.

significant difference is in the weeks to completion, with graphic design projects taking on average half the time of software engineering projects.

In response to another question which asked about the characteristics of systems developed, it was found that graphic designers were substantially less experienced than software engineers in developing systems that are “database-driven” or “feature many dynamically generated pages”. At first impression, this result might suggest that graphic designers mostly develop static hypermedia systems where designing for maintainability might not be such an important consideration. However, a closer inspection of the data shows that many of the systems they develop have “frequently changing content” and/or “integrate with other back-end systems”.

## III. DISCUSSION OF FINDINGS

### USE OF PROCESSES, METHODS, AND APPROACHES

Concepts such as “process”, “method” and “approach” are difficult to label neatly. The terminology used in this paper is consistent with the definitions set out by Wynekoop & Russo [1995]. However, caution must be exercised in analysing data because the possibility of various interpretations may give rise to measurement error. Variations in interpretation may in part explain why previous research on the use of methods and approaches in Web/hypermedia systems development has been inconsistent. Whereas Britton et al [1997] found that “the ‘big bang’ approach to system development is rare”, and Barry & Lang [2003] tell of an eclectic mix of approaches in use drawn from various background disciplines, Russo & Graham [1999] reported that none of their respondents used a formal system development method.

In the literature several authors [De Troyer, 2001, Lowe and Hall, 1999, Murugesan and Deshpande, 1999, Pauen et al., 1998] speculate that the current state of hypermedia development practice is characterized by “opportunistic”, “ad hoc”, “quick and dirty” approaches. This study suggests that hypermedia systems development is much more disciplined than commonly believed. In reply to a closed multiple-choice question, 83.6% of 165 respondents indicated that their organization uses a hypermedia development process that involves clear tasks and/or phases within it. In half of these organizations, these processes are explicitly documented (Table 1). Only 27 of the 165 (16.4%) organizations from which responses were received do not follow a clear process.

Table 1. Organization's Hypermedia Development Process

| | **SE** n = 54 * | **GD** n = 44 | **Overall** n = 165 |
|---|---|---|---|
| There is no clear process | 16.4% | 18.2% | 16.4% |
| Clear tasks and/or phases, though the process used is not explicitly documented | 45.5% | 40.9% | 41.8% |
| Clear tasks and/or phases, according to an explicitly documented process | 36.4% | 40.9% | 41.8% |

* One respondent from this group did not answer this question i.e. 1 missing response overall.

Because the unit of analysis for this question was the organization as opposed to the individual, it is not entirely appropriate to compare software engineers and graphic designers here. As should be expected in a consistent data set, no statistically significant difference between groups were found, meaning that the organizations within which they work are comparable as regards visibility of processes. What is perhaps of interest is that software engineers and graphic designers are equally able to recognise an implicit process as opposed to a total absence of process. For organizations with no clear process, respondents were asked if they regard the lack of process as a problem. About half of the software engineering group (3 of 7) said that it was, as did two-thirds (4 of 6) of the graphic designers, though these numbers are too small to draw any firm conclusions.

A much more varied 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" (Table 2).

Table 2. Use of Methods and Approaches in Hypermedia Systems Design

| | **SE** *n = 34* *Item Response Rate 61.8%* | **GD** *n = 14* *Item Response Rate 31.8%* |
|---|---|---|
| Hybrid, customised, or proprietary in-house method or approach | 14.7% (5) | 28.5% (4) |
| Traditional "legacy" software development methods and approaches, or variants thereof e.g. *SSADM, Yourdon, JSP, SDLC / Waterfall* | 38.2% (13) | 7.1% (1) |
| Approaches that are focused around the use of tools and development environments, e.g. *PHP, Java, Flash, ASP, J2EE* | 11.8% (4) | 35.7% (5) |
| Object-oriented development methods and approaches e.g. *RUP, OOA&D* | 17.6% (6) | 7.1% (1) |
| HCI / Human Factors Engineering methods e.g. *User Centred Design, Interaction Design, Goal-based Requirements* | 0.0% (0) | 14.2% (2) |
| Rapid or agile development methods and approaches e.g. *RAD, Extreme Programming* | 14.7% (5) | 7.1% (1) |
| Incremental or evolutionary methods and approaches e.g. *Spiral Model, Staged Delivery, Iterative Design* | 5.9% (2) | 7.1% (1) |
| No method used / development approach is "ad hoc" | 5.9% (2) | 7.1% (1) |
| Specialised non-proprietary methods for Web and hypermedia systems development e.g. *Fusebox, WSDM, OOHDM, RMM* | 2.9% (1) | 7.1% (1) |

Figures in brackets after percentages are absolute numbers of responses.

This question apparently gave rise to some confusion, particularly amongst graphic designers, many of whom provided no details other than to comment that they were unclear what the question meant. Accordingly, item response rates for this question were low (SE 61.8%; GD 31.8%) and caution must be exercised in analysing the data as the number of responses is statistically small. It seems that the notion of a "method" or "approach" is alien to many graphic designers, or at least that their understanding of such terms is rather different than that of software engineers. Because many of the responses received were ambiguous, it was difficult to classify them precisely and some are double-counted within overlapping categories. Furthermore, as many of the responses which indicated that an in-house method is used did not provide any details on its orientation, percentages in certain categories may be understated; for example, it is highly likely that many in-house methods might be regarded as rapid/agile or incremental/evolutionary.

For graphic designers, approaches seem to be primarily driven by the particular development tools being used, more so than for software engineers (35.7% versus 11.8%). Tool-driven approaches traditionally were frowned upon within the software engineering literature where thorough up-front analysis and design was emphasised rather than going "straight to implementation". At best, this logical/physical separation is artificial [Walz et al., 1993], and the insistence on up-front analysis as an economical error-trapping mechanism is based upon theories that were devised in the era of third-generation programming languages [Boehm, 1981] when the profile of systems development was quite different from that of today [Fitzgerald, 2000].

Given that it is now possible with rapid visual development tools to generate and refine a design iteratively within a relatively short period without the same expense as of old, it makes strong sense simply to invest in a suite of such tools and devise working methods around them. Indeed, that may be much easier than selecting a method and then trying to plug development tools into that method. As further support for this point, it is notable from Table 2 that little usage of hypermedia-specific methods such as RMM, OOHDM, or WSDM is reported. This finding ought not be surprising, because it is not clear how these methods could be readily implemented using industry-standard hypermedia development tools. Significantly, the most widely used hypermedia-specific method (*Fusebox)* was devised by a community of practitioners rather than academics and books are available which illustrate how *Fusebox* may be implemented using ColdFusion, PHP, and Active Server Pages.

Not surprisingly, the incidence of usage of traditional software development methods and object-oriented methods amongst software engineers is much higher than amongst graphic designers. What is surprising is that no software engineers claimed to use methods or approaches that could be classified under the HCI banner. This finding lends some evidence to the conjecture by Gallagher & Webb [1997] that software engineers are much more concerned with back-end functionality than with front-end usability. However, it is possible that some of the proprietary or hybrid in-house methods used by software engineers might draw upon HCI principles.

## ATTITUDES TOWARDS SYSTEMATIC PLANNING

Even though a small minority admit to not having any process, method or approach (Tables 1 and 2), software engineers and graphic designers alike widely accept the necessity for explicitly documented plans and considered actions (Table 3). In both groups, almost all respondents agreed on an essential need for planning (SE 94.5%; GD 97.7%), and the vast majority agreed that plans and working methods should be clearly documented (SE 74.6%; GD 86.4%).

69.2% of software engineers and 67.5% of graphic designers agreed that ad hoc methods generally result in poor systems, and the suggestion that “documented working methods are pointless” was firmly rejected with 72.7% of software engineers and 81.8% of graphic designers in disagreement. A Mann-Whitney comparison of the two groups showed no significant difference between them regarding attitudes towards plans, considered actions, and the value of documented working methods. Indeed, on the face of it, graphic designers actually seem to be slightly more strongly in favour of such systematic measures than software engineers.

## USE AND PERCEIVED USEFULNESS OF DIAGRAMMING TECHNIQUES FOR CONCEPTUAL MODELLING

Software systems are amongst the most complex of all human inventions. It can be difficult to visualize their conceptual structure because they do not contain any readily identifiable geometric representation [Brooks, 1987]. Hypermedia systems may be even more complex than conventional systems and challenge the limits of existing models and metaphors [Palmquist, 1996]. Essentially, hypermedia attempts to emulate the intricate mechanisms of the human mind by associating blocks of knowledge with one another in a complex multitude of “associative trails” [Bush, 1945]. Such is the potential complexity of these arbitrary links that the links tend towards chaotic “spaghetti” structures, rather like go-to programming which Dijkstra [1968] criticized as being “*just too primitive; it is too much of an invitation to make a mess*”. Issues such as “getting lost in cyberspace”, locating information, visualizing knowledge structures, and managing content quickly become major considerations for hypermedia systems as they scale up. At this point diagramming techniques become useful, as they can help to combat complexity by reducing system descriptions to simplified abstractions.

Table 3. Attitudes Towards Systematic Planning

| | | Firmly disagree | Disagree | Neutral | Agree | Firmly agree |
|---|---|---|---|---|---|---|
| Ad hoc "improvised" hypermedia development approaches generally result in systems of poor quality | p = 0.48 | | | | | |
| *Software Engineering* | n = 52 | 3.8% | 19.2% | 7.7% | 34.6% | 34.6% |
| *Graphic Design* | n = 37 | 2.7% | 21.6% | 8.1% | 45.9% | 21.6% |
| To combat system complexity and time pressures, there is an essential need for planning and considered action | p = 0.46 | | | | | |
| *Software Engineering* | n = 55 | 0.0% | 1.8% | 3.6% | 30.9% | 63.6% |
| *Graphic Design* | n = 44 | 0.0% | 0.0% | 2.3% | 40.9% | 56.8% |
| To ensure efficient and effective collaboration within the development team, plans and working methods should be explicitly documented | p = 0.79 | | | | | |
| *Software Engineering* | n = 55 | 1.8% | 5.5% | 18.2% | 29.1% | 45.5% |
| *Graphic Design* | n = 44 | 2.3% | 2.3% | 9.1% | 45.5% | 40.9% |
| Explicitly documented working methods are futile and pointless | p = 0.92 | | | | | |
| *Software Engineering* | n = 55 | 41.8% | 30.9% | 20.0% | 1.8% | 5.5% |
| *Graphic Design* | n = 44 | 34.1% | 47.7% | 13.6% | 4.5% | 0.0% |

"p" is a measure of the statistical significance of the differences between groups. It has a value between 0 and 1, indicating the probability that differences arising are not merely coincidental.

"n" is the number of valid responses, i.e. the number of respondents in each group who held an opinion.

"Neutral" indicates that the respondent had an opinion, as opposed to "No Opinion" (which was a separate designated response) or a blank missing response. Likewise for Table 4.

## USE AND PERCEIVED USEFULNESS OF DIAGRAMMING TECHNIQUES FOR CONCEPTUAL MODELLING

Software systems are amongst the most complex of all human inventions. It can be difficult to visualize their conceptual structure because they do not contain any readily identifiable geometric representation [Brooks, 1987]. Hypermedia systems may be even more complex than conventional systems and challenge the limits of existing models and metaphors [Palmquist, 1996]. Essentially, hypermedia attempts to emulate the intricate mechanisms of the human mind by associating blocks of knowledge with one another in a complex multitude of "associative trails" [Bush, 1945]. Such is the potential complexity of these arbitrary links that the links tend towards chaotic "spaghetti" structures, rather like go-to programming which Dijkstra [1968] criticized as being "*just too primitive; it is too much of an invitation to make a mess*". Issues such as "getting lost in cyberspace", locating information, visualizing knowledge structures, and managing content quickly become major considerations for hypermedia systems as they scale up. At this point diagramming techniques become useful, as they can help to combat complexity by reducing system descriptions to simplified abstractions.

Respondents were asked to indicate the extent to which they agreed with a number of statements regarding the use of diagramming techniques in hypermedia systems development (Table 4). To avoid speculative responses, a "No Opinion" category was provided. Both groups were very strongly in agreement that diagrams are vital in order to conceptualise hypermedia systems (SE 90.4%; GD 88.6%). This response would indicate that even though many graphic designers use tool-driven development approaches, they are not cutting blindly to implementation but are using some form of diagramming to aid conceptualisation.

Previous research [Barry and Lang, 2003, Britton et al., 1997, McClure, 1998, Newman and Landay, 2000] showed that diagrams used in Web/hypermedia development are mostly paper-based and CASE tools are little used. Respondents were asked whether they agree with the statement that "there is little benefit in using formal diagramming techniques without proper computerized support". The rationale behind this statement is that the effort that is required to use abstract modeling techniques such as class diagrams or statecharts without tools to validate them or generate working prototypes may be disproportionate to the benefit derived. Given that graphic designers would not be expected to use formal diagramming techniques to the same extent as

Table 4. Attitudes Towards Use of Diagramming Techniques

| | | Firmly disagree | Disagree | Neutral | Agree | Firmly agree |
|---|---|---|---|---|---|---|
| Diagrams are vital in order to communicate the conceptual structure of a hypermedia system | p = 0.84 | | | | | |
| *Software Engineering* | n = 52 | 0.0% | 1.9% | 7.7% | 32.7% | 57.7% |
| *Graphic Design* | n = 44 | 0.0% | 6.8% | 4.5% | 34.1% | 54.5% |
| There is little benefit in using formal diagramming techniques without proper computerised support | p = 0.80 | | | | | |
| *Software Engineering* | n = 52 | 5.8% | 42.3% | 15.4% | 25.0% | 11.5% |
| *Graphic Design* | n = 27 | 3.7% | 33.3% | 25.9% | 37.0% | 0.0% |

"p" is a measure of the statistical significance of the differences between groups, based on a Mann-Whitney comparison of two independent samples.

"n" is the number of valid responses, i.e. the number of respondents in each group who held an opinion.

software engineers, it is not surprising that 38.6% (17 of 44) had no opinion on the matter. Of those graphic designers who had an opinion, 25.9% were undecided (i.e. neutral), those agreeing and disagreeing were split evenly, and there were few strongly held feelings one way or the other. Software engineers were much more definite in their views, and although 48.1% (total of disagree and strongly disagree) felt that it was beneficial to use formal diagramming methods even without computer assistance, 36.5% felt that using such methods without proper computer assistance was not beneficial. While it is understandable why graphic designers might choose not to use hypermedia-specific methods such as RMM and OOHDM, it is less clear why such methods have not been adopted by the software engineering community. Perhaps it is because they use proprietary diagramming notations that have negligible support from popular computer-based modeling tools.

Another question listed a number of diagramming techniques, and asked respondents to indicate how useful they considered each technique or else to indicate that they had not used that particular technique. The list included techniques with formalised notations (e.g. entity-relationship diagrams, statecharts) as well as informal techniques (e.g. storyboards) and semi-formal techniques (e.g. flowcharts). Table 5 shows the extent to which techniques were used at some stage by respondents from both groups. Some care must be taken in interpreting the table

because of the imprecise nature of informal and semi-formal techniques (e.g. what's the difference between a "storyboard", a "flowchart", and a "2-D site mapping technique"?). Not surprisingly, software engineers are more experienced than graphic designers in using formalised software diagramming techniques such as entity-relationship diagrams, class diagrams, statecharts, and use case diagrams. However, more graphic designers than one might have expected did, at some point, use these same techniques. It is notable that almost 60% of them used use case diagrams. However, this finding ought not astound given that use case diagrams are an intuitive, easy-to-draw user-centred representation of a system, and hence are likely to be favoured by graphic designers. Conversely, informal techniques such as storyboards and site maps have been used as much by software engineers as by graphic designers.

Table 5. Use of Conceptual Modelling Techniques for Hypermedia Design

| | SE | GD |
|---|---|---|
| Screen prototypes / Mockups | 96.4% | 97.7% |
| Entity-Relationship Diagrams | 96.4% | 43.2% |
| Flowcharts | 94.5% | 88.6% |
| 2-D site mapping techniques | 92.7% | 95.5% |
| Storyboards | 83.6% | 86.4% |
| Object-Oriented Class Diagrams | 76.4% | 43.2% |
| Use Case Diagrams / Scenarios | 74.5% | 56.8% |
| Statecharts / State Diagrams | 61.8% | 34.1% |
| 3-D site mapping techniques | 50.9% | 47.7% |

Although Table 5 shows the extent of technique usage, it does not show a more pertinent metric: the perceived usefulness of techniques. Numerous authors suggest that traditional software engineering techniques are not readily transferable to hypermedia design, and that some aspects of hypermedia design are not considered by traditional techniques [Balasubramanian and Turoff, 1995, Nanard and Nanard, 1995, Retschitzegger and Schwinger, 2000, Rossi and Schwabe, 2001, Siau, 1998]. On the other hand, others argue that traditional software engineering techniques still apply [Constantine and Lockwood, 2002, Pressman, 2000], and that techniques from existing dynamic media such as film production are relevant [Gygi, 1990]. Table 6 presents a summary of each group's perceived usefulness of the various techniques. In the original questionnaire, this question was asked in the form of a 7-point Likert scale, but the results are collapsed here. The original points on the scale are shown in the header row, alongside the recoded labels.

A Mann-Whitney test was performed to compare the two groups, and a few statistically significant differences were revealed. Although screen prototypes / mockups are highly regarded by both groups, graphic designers were more emphatic in their ratings. Entity-relationship diagrams and use case diagrams are perceived as being of considerable use by software engineers but much less so by graphic designers, which might simply be attributed to training and understanding. On the other hand, software engineers do not consider 2-D site mapping techniques to be as useful as do graphic designers, and they have slightly less regard for flowcharts and storyboards.

Because of the multidimensional information and navigation structures that can exist within the abstract world of hypermedia cyberspace, it is not surprising that 3-D hypermedia mapping techniques have been the focus of considerable research [Andrews, 1998, Benford et al., 1999, Das Neves, 1997, Kahn and Lenk, 2001, Mukherjea and Foley, 1995, Munzner and Burchard, 1995, Olsen et al., 1993, Zizi, 1995]. However, there has been quite limited use of 3-D mapping techniques thus far in practice, with only about half the respondents in each group having ever used such techniques (Table 5: SE 50.9%; GD 47.7%). Regarding their usefulness, 53.6% of those software engineers who have experience of using 3-D maps consider them as being at

least of some use, a view shared by just 38.1% of graphic designers. This apparent difference is not statistically significant (p = 0.08).

Likewise, there is divided opinion on the usefulness of statecharts. Amongst graphic designers, just over half (53.3%) of those who have used statecharts consider them as being of little or no use, but the remainder regard them as being of some or significant use. Software engineers seem to be more favorably inclined towards statecharts than graphic designers, for not only do they use them more often but also a larger proportion (58.8%) regard them as being somehow useful.

Table 6. Perceived Usefulness of Diagramming Techniques

| | p | Completely useless 1 | Of little use 2, 3 | Of some use 4, 5 | Of significant use 6, 7 |
|---|---|---|---|---|---|
| Screen prototypes / Mockups | .00 | | | | |
| *Software Engineering* | | - | 3.8% | 24.5% | 71.7% |
| *Graphic Design* | | - | - | 4.7% | 95.3% |
| Entity-Relationship Diagrams | .02 | | | | |
| *Software Engineering* | | 1.9% | 7.5% | 47.2% | 43.4% |
| *Graphic Design* | | - | 21.1% | 63.2% | 15.8% |
| 2-D site mapping techniques | .04 | | | | |
| *Software Engineering* | | - | 13.7% | 52.9% | 33.3% |
| *Graphic Design* | | - | 2.4% | 40.5% | 57.1% |
| Flowcharts | .05 | | | | |
| *Software Engineering* | | 2.0% | 7.8% | 45.1% | 45.1% |
| *Graphic Design* | | - | 5.1% | 33.3% | 61.5% |
| Use Case Diagrams / Scenarios | .06 | | | | |
| *Software Engineering* | | 2.4% | 7.3% | 46.3% | 43.9% |
| *Graphic Design* | | - | 32.0% | 40.0% | 28.0% |
| 3-D site mapping techniques | .08 | | | | |
| *Software Engineering* | | 3.6% | 42.9% | 39.3% | 14.3% |
| *Graphic Design* | | 9.5% | 52.4% | 28.6% | 9.5% |
| Object-Oriented Class Diagrams | .12 | | | | |
| *Software Engineering* | | 4.8% | 14.3% | 54.8% | 26.2% |
| *Graphic Design* | | 5.3% | 42.1% | 31.6% | 21.1% |
| Storyboards | .22 | | | | |
| *Software Engineering* | | 2.2% | 10.9% | 37.0% | 50.0% |
| *Graphic Design* | | - | 10.5% | 26.3% | 63.2% |
| Statecharts / State Diagrams | .28 | | | | |
| *Software Engineering* | | 2.9% | 38.2% | 44.1% | 14.7% |
| *Graphic Design* | | 13.3% | 40.0% | 33.3% | 13.3% |

Alongside perceived usefulness, ease of use is an important factor which impacts the usage level of technology [Davis, 1989]. For both 3-D maps and statecharts, the low levels of usage relative to other diagramming techniques might therefore be explained by difficulties in using them. Given that few well-known computer-based 3-D visualization/drawing tools exist, one is left with the onus of manually drawing 3-D diagrams on paper, which may not be easy if indeed at all possible for some types of such diagrams. Nor are statecharts intuitively easy to draw or read. Indeed, it is ironic that statecharts are intended to model dynamic, interactive, and usable systems, yet as a diagramming technique many would consider statecharts to be abstract, inanimate, and unusable.

## IV. SYNOPSIS AND CONCLUSIONS

This survey found that graphic designers are as committed to the need for plans and documented working methods as software engineers, which may come as a revelation to some commentators. Of course, the granularity and level of those methods and plans is another matter. Previous research [Barry and Lang, 2001] indicates a preference towards light, flexible approaches rather than cumbersome methodologies. This indication is borne out here by the findings that rapid/agile methods, tool-driven approaches, and in-house/customised methods are used frequently.

Graphic designers are more aware of software engineering development methods and diagramming techniques than might be popularly believed, but they tend to prefer their own informal and semi-formal diagramming techniques and rely much more on tool-driven development approaches than software engineers. Likewise, as should be expected, software engineers take their own legacy of traditional software development approaches and continue to apply and adapt these approaches to the new and unique challenges of hypermedia systems development.

Graphic designers and software engineers are both firmly of the view that diagrams are essential to visualize the conceptual design of hypermedia systems. Screen mockups, 2-D site maps, flowcharts, and storyboards are the most used and most useful techniques amongst both groups. Although software engineers continue to make much use of traditional techniques such as entity-relationship diagrams and class diagrams, they are now also using informal techniques such as storyboarding to a major extent. This usage

- may be to facilitate communication with graphic designers and other team members,
- may be the result of the increased representational capacities of storyboards as opposed to traditional techniques, or
- may be the result of the ease of drawing.

Although many hypermedia-specific methods are proposed in the academic literature (e.g. *RMM*, *OOHDM*, *WSDM*, *W3DT*), the findings of this survey are that they are hardly ever used in practice in Ireland. One could surmise that the low usage of academic methods can, in part, be attributed to lack of awareness, or perhaps some degree of inertia amongst practitioners even when they are aware. However, the answer probably lies elsewhere. Barry & Lang [2001] have previously reported that understandability, ease-of-use, and widespread acceptance and reputation amongst developers are major issues in method selection. In all these regards, most of these methods contain serious deficiencies.

Wynekoop & Russo [1995] 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 and impractical. With the emergence of Web and hypermedia systems, further talk ensued about a “pressing need for new methods and tools” [Murugesan et al., 1999]. It is doubtful if wholly new methods are needed, but that is a separate argument beyond the immediate scope of this paper. However, it is clear that hypermedia systems development is a multi-disciplinary domain [Barry and Lang, 2003]. Whatever new methods and techniques are proposed should consider the implications of managing and coordinating software development within a

collaborative multi-disciplinary work environment. Specifically, such methods and techniques should be generally usable by all stakeholders and be accompanied by practical guidance on how to implement them.

*Editor's Note:* This article was received on June 18, 2003 and was published on Septembr 15, 2003. It was with the author one week for 1 revision.

**REFERENCES**

Andrews, K. (1998) "Visualizing Rich, Structured Hypermedia", *IEEE Computer Graphics and Applications,* 18(4), July/August, pp. 40-42.

Balasubramanian, V. and M. Turoff (1995) "A Systematic Approach to User Interface Design for Hypertext Systems". *Proceedings of 28th Hawaii International Conference on System Sciences (HICSS'95)*, Hawaii, USA, January 4-7, 1995. Vol. III, pp. 241-250. Los Alamitos, CA: IEEE Computer Society Press.

Barry, C. and M. Lang (2001) "A Survey of Multimedia and Web Development Techniques and Methodology Usage", *IEEE Multimedia,* 8(3), pp. 52-60.

Barry, C. and M. Lang (2003) "A Comparison of "Traditional" and Multimedia Information Systems Development Practices", *Information & Software Technology,* 45(4), pp. 217-227.

Baskerville, R. and J. Pries-Heye (2001) "Racing the E-Bomb: How the Internet Is Redefining Information Systems Development Methodology", in Russo, N. L., B. Fitzgerald, and J. I. DeGross (eds.) *Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspective. Proceedings of International Federation for Information Processing (IFIP) Working Group 8.2 Conference, Boise, Idaho, USA, 27-29 July 2001*, Boston: Kluwer Academic Publishers, pp. 49-68.

Benford, S. et al. (1999) "Three Dimensional Visualization of the World Wide Web", *ACM Computing Surveys,* 31(4), December, pp. 1-16.

Berners-Lee, T. (1996) "WWW: Past, Present, and Future", *IEEE Computer,* 29(10), pp. 69-77.

Boehm, B. W. (1981) *Software Engineering Economics,* Englewood Cliffs, NJ: Prentice Hall.

Britton, C. et al. (1997) "A Survey of Current Practice in the Development of Multimedia Systems", *Information and Software Technology,* 39, pp. 695-705.

Brooks, F. P. (1987) "No Silver Bullet / Essence and Accidents of Software Engineering", *IEEE Computer,* 20(4), pp. 10-18.

Bush, V. (1945) "As We May Think", *The Atlantic Monthly,* July, pp. 101-108.

Carstensen, P. H. and L. Vogelsang (2001) "Design of Web-Based Information Systems - New Challenges for Systems Development?". *Proceedings 9th European Conference on Information Systems*, Bled, Slovenia, June 27-29, pp. 536-547.

Constantine, L. L. and L. A. D. Lockwood (2002) "Usage-Centered Engineering for Web Applications", *IEEE Software,* 19(2), pp. 42-50.

Das Neves, F. (1997) "The Aleph: A Tool to Spatially Represent User Knowledge About the WWW Docuverse". *Proceedings of ACM Conference on Hypertext*, Southampton, England, April 6-11, pp. 197-207.

Davis, F. D. (1989) "Perceived Usefulness, Perceived Ease of Use, and End User Acceptance of Information Technology", *MIS Quarterly,* (13)3, September, pp. 318-339.

De Troyer, O. (2001) "Audience-Driven Web Design", in Rossi, M. and K. Siau (eds.) *Information Modeling in the New Millennium*, London: Idea Group Publishing, pp. 442-461.

Dillman, D. A. (2000) *Mail and Internet Surveys: The Tailored Design Method,* 2nd edition, New York: John Wiley.

Eriksen, L. B. (2000) “Limitations and Opportunities for System Development Methods in Web Information System Design”, in Baskervielle, R, J. Stage, and J.I. DeGross (eds.) *Proceedings of IFIP 8.2000 International Conference - Organizational and Social Perspective on Information Technology*, *IFIP TC8 WG 8.2 International Conference on the Social and Organizational Perspective on Research and Practice in Information Technology, June 9-11, 2000*, *Aalborg, Denmark*, Boston: Kluwer Academic Publishers. pp. 473-486

Fitzgerald, B. (2000) "Systems Development Methodologies: The Problem of Tenses", *Information Technology & People,* 13(3), pp. 174-185.

Gallagher, S. and B. Webb (1997) “Competing Paradigms in Multimedia Systems Development: Who Shall Be the Aristocracy?”. *Proceedings of Fifth European Conference on Information Systems*, Cork, Ireland, June 19-21.

Gygi, K. (1990) “Recognizing the Symptoms of Hypertext ... And What to Do About It”, in Laurel, B. (ed.) *The Art of Human-Computer Interface Design*, Reading, Massachusetts: Addison-Wesley, pp. 279-287.

Kahn, P. and K. Lenk (2001) *Mapping Web Sites: Digital Media Design,* RotoVision.

Liu, M., C. Jones, and S. Hemstreet (1998) "Interactive Multimedia Design and Production Processes", *Journal of Research on Computing in Education,* 30(3), pp. 254-280.

Lowe, D. and W. Hall (1999) *Hypermedia & the Web: An Engineering Approach,* Chichester: Wiley.

Lowe, D. B. and J. Eklund (2002) "Client Needs and the Design Process in Web Projects", *Journal of Web Engineering,* 1(1), pp. 23-36.

Luce, A. A. (1958) *Logic,* London: Teach Yourself Books / English Universities Press.

McClure, S. (1998) “Web Application Development Developer Perspectives: An IDC White Paper”. Framingham, MA: International Data Corporation.

Mukherjea, S. and J. D. Foley (1995) "Visualizing the World-Wide Web with the Navigational View Builder", *Computer Networks and ISDN Systems,* 27(6), pp. 1075-1087.

Munzner, T. and P. Burchard (1995) “Visualizing the Structure of the World Wide Web in 3d Hyperbolic Space”. *Proceedings of the First Symposium on Virtual reality Modeling Language (VRML '95)*, San Diego, CA, December, pp. 33-38. New York: ACM Press.

Murugesan, S. and Y. Deshpande (1999) “Preface to ICSE'99 Workshop on Web Engineering”. *Proceedings of International Conference on Software Engineering (ICSE'99)*, Los Angeles, CA, pp. 693-694.

Murugesan, S. et al. (1999) “Web Engineering: A New Discipline for Development of Web-Based Systems”. *Proceedings of First ICSE Workshop on Web Engineering*, Los Angeles, CA, May, pp. 1-9. ACM.

Nanard, J. and M. Nanard (1995) "Hypertext Design Environments and the Hypertext Design Process", *Communications of the ACM,* 38(8), pp. 49-56.

Newman, M. W. and J. A. Landay (2000) “Sitemaps, Storyboards, and Specifications: A Sketch of Web Site Design Practice”. *Proceedings of ACM Symposium on Designing Interactive Systems*, Brooklyn, New York, pp. 263-274.

Nürnberg, P. J. and H. Ashman (1999) “What Was the Question? Reconciling Open Hypermedia and World Wide Web Research”. *Proceedings of ACM Conference on Hypertext and Hypermedia*, Darmstadt, Germany, February 21-25 1999, pp. 83-90. New York: ACM Press.

Olsen, K. A. et al. (1993) "Visualization of a Document Collection with Implicit and Explicit Links: The Vibe System." *Scandinavian Journal of Information Systems,* (5), pp. 79-95.

Palmquist, R. A. (1996) “The Search for an Internet Metaphor: A Comparison of Literatures” in Hardon, S. (ed.) *Proceedings of the 59th ASIS Annual Meeting.* (33), Information Today, pp. 198-202.

Pauen, P., J. Voss, and H.-W. Six (1998) “Modeling Hypermedia Applications with Hydev”. *Proceedings of IFIP 13.2 Working Conference / Designing Effective and Usable Multimedia Systems*, Stuttgart, Germany, September 1998, pp. 23-40. Boston: Kluwer Academic Publishers.

Pressman, R. S. (2000) "What a Tangled Web We Weave", *IEEE Software,* 17(1), pp. 18-21.

Pressman, R. S. et al. (1998) "Can Internet-Based Applications Be Engineered?" *IEEE Software,* 15(5), September-October, pp. 104-110.

Retschitzegger, W. and W. Schwinger (2000) “Towards Modeling of Dataweb Applications - a Requirements Perspective”. *Proceedings of Americas Conference on Information Systems (*AMCIS 2000*)*, Long Beach, California, August 10-13*.* Vol. 1, pp. 149-155.

Rossi, G. and D. Schwabe (2001) “Object-Oriented Web Applications Modeling”, in Rossi, M. and K. Siau (eds.) *Information Modeling in the New Millennium*, London: Idea Group Publishing, pp. 463-483.

Russo, N. L. and B. R. Graham (1999) “A First Step in Developing a Web Application Design Methodology: Understanding the Environment” in Wood-Harper, A.T., N. Jayaratna, and J.R.G. Wood (eds.) *Methodologies for Developing and Managing Emerging Technology Based Information Systems: Proceedings of 6th International BCS Information Systems Methodologies Conference*, London, pp. 24-33. Berlin: Springer pp. 24-33

Siau, K. (1998) “Method Engineering for Web Information Systems Development: Challenges and Issues”. *Proceedings of Association for Information Systems Americas Conference*, Baltimore, MD, August 14-16, pp. 1017-1019.

Taylor, M. J. et al. (2002) "Methodologies and Website Development: A Survey of Practice", *Information and Software Technology,* 44(6), pp. 381-391.

Vertelney, L., M. Arent, and H. Lieberman (1990) “Two Disciplines in Search of an Interface / Reflections on a Design Problem”, in *The Art of Human-Computer Interface Design*, Reading, MA: Addison-Wesley, pp. 45-55.

Vora, P. R. (1998) "Designing for the Web: A Survey", *ACM interactions,* 5(3), May-June, pp. 13-30.

Walz, D. B., J. J. Elam, and B. Curtis (1993) "Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration", *Communications of the ACM,* 36(10), pp. 63-77.

Whitley, E. A. (1998) “Method-ism in Practice: Investigating the Relationship

between Method and Understanding in Web Page Design”. *Proceedings of Nineteenth Annual International Conference on Information Systems*, Helsinki, Finland, December 13-16.

Wynekoop, J. L. and N. L. Russo (1995) "Systems Development Methodologies: Unanswered Questions", *Journal of Information Technology,* 10(2), June, pp. 65-73.

Zizi, M. (1995) “Providing Maps to Support the Early Stage of Design of Hypermedia Systems”. *Hypermedia Design / Proceedings of the International Workshop on Hypermedia Design*, Montpellier, France, June 1-2, pp. 93-104. Berlin: Springer Verlag.

## ABOUT THE AUTHOR

**Michael Lang** is Lecturer in Information Systems at National University of Ireland, Galway. His principal research interests are systems analysis and design, information systems development, and Web and multimedia systems. His research in these areas is published in a number of international journals and conferences. This paper forms part of a major on-going project that investigates the nature of hypermedia systems design in practice.

Communications of the Association for Information Systems

ISSN: 1529-3181