On the Acceptability of Conceptual Design Models for Web Applications

Franca Garzotto and Vito Perrone

HOC – Hypermedia Open Center Department of Electronics and Information, Politecnico di Milano (Italy) Piazza Leonardo da Vinci 32, I-20133 Milano, Italy {garzotto,perrone}@elet.polimi.it

Abstract. A possible measure of quality for any model or methodology is the degree of acceptance and usage. This paper discusses the factors that contribute to the industrial acceptability of conceptual models for web application design. We present an empirical study that examined 62 companies or institutions (in America and Europe) involved in large-scale web application development. By investigating the "desiderata" of industrial "practitioners" (developers, designers, or project managers of web applications), we aimed at identifying the requirements that a web design model should satisfy in order to be accepted and used at industry level. The paper describes the design of the study and its main results.

Keywords: web conceptual design, quality, acceptability, user requirements, questionnaire-based study

1 Introduction

"Quality, like beauty, is very much in the eyes of the beholder" [1]

In an arena where large scale, in-house, information intensive web applications dominate the field, industry need comprehensive, well structured development methods and techniques. Conceptual design models have the potential of playing an important role in this scenario. They enable developers to describe a web application at the proper level of abstraction, and support a systematic approach to design. They promote the evolution of web practice from a craft to a structured discipline, and improve the quality and cost effectiveness of the entire development process.

Unfortunately, in spite of the proliferation of conceptual design models for the web produced in the academic world since mid 1990s [5-10], a number of studies [2-4] highlight that, with very few exceptions, practitioners in industry aren't using them. Perhaps it is time to face the fact that, except our students and the partners of our research projects, the rest of the "real" world does not adopt our methods.

Why did we fail? What went wrong? One obvious answer might be that our models simply did not meet the requirements and expectations of industrial users. If "fitness to requirements" is a quality indicator (as suggested by N. Fenton [1] and Dix et al [13]), we must admit that our models do not have the appropriate level of quality from

an industrial perspective (although perhaps they are good and successful from an academic viewpoint.)

Understanding industrial user requirements represents therefore the first step towards improving the quality of our conceptual models for the web. The goal of the empirical study reported in this paper goes along this direction: investigating the industrial user needs and identifying some properties that a web conceptual model should have in order to be *acceptable* in the real world, and potentially used in practice. The rest of this paper describes the design of our study (section 2) and discusses its main findings (section 3).

2 Approach of Our Study

Our work is carried on in the context of the EC (European Commission) funded project UWA- "Ubiquitous Web Applications" - IST 2000-25131. The UWA purpose is the development of models and tools to support the design of multichannel web applications. Within this project, the EC explicitly requires a three steps validation activity: 1) to identify some factors for the industrial acceptability of the UWA models and tools; 2) to compare the UWA "products" against these factors, and 3) to identify the guidelines to improve them. The questionnaire study is developed to implement step 1. We therefore carried on a questionnaire-based study, involving companies and organizations which carry on large-scale web application development. To design the questionnaire, our approach is to hypothesize a set of potentially important requirements for a design model, asking users to judge their relevance. These requirements arise both from our experience in building and using design models in many (over 25) industrial and academic development projects, and from some studies reported in literature [2-4]. We adopt an "holistic" view of conceptual design models, looking at them within "the organizational context in which they have to work" [2]. Our general assumption is that in order to be accepted and used in an industrial environment, a design model alone is not enough. Even if of excellent intrinsic quality, a model should be supported by a number of complementary features, including a proper *methodology*, an accurate *documentation*, and a set of support tools¹.

A methodology defines how to use the model. It identifies the design process that helps designers structure the design activity and carry on the different design tasks in a systematic way. Acceptability factors that we want to verify include the availability of methodologies that are flexible, adaptable to the specific needs of a company, integrated with the whole development process, and able to support, at some degree, the human task of translating design choices into implementation solutions². In addition, we want to explore if a methodology should take into account the managerial aspects of the design process, assisting project management – a task that is crucial to any commercial production.

¹ Among other works, the results of the survey reported in [2] highlights that the industry needs models coupled with methodologies and support to learning and using them.

² The Entity relationship model, for example, addresses this aspect by providing a set of "rules" or guidelines to map ER schemas into relational schemas.

High quality *documentation* about the model and the design process, is crucial for *learnability*, which in turn is a fundamental factor for the usability of any complex method.

Software tools are needed to assist the design activity, relieving designers from all the tedious tasks dealing with producing design specifications and delivering good quality design documentation. Ideally, the tools should also provide some (semi) automated support to translate of design specification into implementation structures.

To verify the above assumptions, our questionnaire includes various types of questions: questions addressing the requirements for a design model per se, questions addressing the requirements for a methodology and a design process, questions addressing the requirements for documentation, and questions about support tools. The questionnaire is organized in three main sections, discussed in the rest of this section: "Requirements for the Model and the Design Process", "Requirements for Documentation", "Requirements for Support Tools"³.

The **first section** ("Requirements for the Model and the Design Process") considers a conceptual design model both "in isolation" and in the context of the design process, and focuses on some general characteristics of both a model and its methodology complementary feature. We first introduce some general, potentially relevant characteristics of a design model and a design methodology, as described in the table of figure 1. Respondents are asked to fill the table by marking with an X the degree of relevance of each characteristic. The table also includes a generic question concerning software tools, to verify, at a very general level, the assumption that a CASE tool is perceived as important for the model acceptability – issue which is investigated in depth in section three. Additional sets of questions explore in detail each specific characteristic of the model and the design methodology, as exemplified in figure 2.

Characteristic	Not relevant at all	Relevant	Strongly relevant	Absolutely necessary
Ease to learn				
Ease to use				
Being a standard				
Documentation support				
Process Customisation				
Support for Iterative and Incremental Design Lifecycle				
Project Management Support				
Fast prototyping				
CASE tools support				

Fig. 1. Investigating general requirements on a design model

³ Each section includes questions, their explanation, and a brief definition of the terminology used (when needed). The questionnaire also includes a section "General Overview of Methodologies Usage" (not discussed in this paper), which investigates the current industrial practice of web design methods and the adoption of the different approaches proposed in literature.

For a design model "alone", table in figure 1 and the complementary detailed questions aim to identify the relevance of the following factor: being *ease to learn* and to use, being a standard, and being *effective for project documentation purposes*⁴.

□ Ease to learn

"Facility in learning the proposed model and notation, composed by primitives, concepts and graphical elements". Our objective is to identify how crucial is learnability (table 1) and which are constraints imposed by industry for spending time and resources in learning a design approach (detailed questions in figure 2). This information, combined with the findings of the second section of the questionnaire "Requirements for Documentation", is important for understanding which type of "training" and didactical documentation is required as a pre-requisite for the acceptance and adoption of a model.

□ Ease to use

"Facility in applying rapidly the concepts and notation in order to produce design specifications for the application under design". The objective of this set of questions is to identify the degree of relevance of easiness of use with respect to other characteristics, and to investigate the attributes that, in the industry expectation, contribute to make a model easy to use. Some aspects we suggest in the detailed questions are: i) the provision of design patterns as high level modeling primitives; ii) flexibility and customizability, i.e., the possibility of using the model in multiple ways, according to the different practices and styles which may be in use in an organization.

□ Being a standard

"The need (or commitment) in the company company to use either an officially standard method (e.g., an IEEE or OMG standards), or a de-facto standard (e.g., UML)". The objective of these questions is to know the degree at which the use of a standard methodology is important, or even mandatory, in the industry field.

D Effectiveness for Documentation Purposes, or "Documentation Support"

"The effectiveness of documenting the design choices using the design model concepts and notations". The objective of this set of questions is to verify the relevance of the *communication power* of a model - how important is to use the model to document the design choices and to communicate them among the various members of the design and development team.

The questions concerning the properties of a design model in the context of the design process address the following characteristics related to a *design methodology*:

D Process customization

These questions investigate the relevance of "being able to adapt and to customize the design process with respect to different situations of use induced by different application fields or different design and development practices used in the company".

Gamma Support for Iterative and Incremental Design Lifecycle

The objective of this set of questions is to verify whether an iterative and incremental process model (which "*defines a set of design steps that can be applied iteratively, in*

⁴ Ease to learn and to use, and being a standard, are generally acknowledged usability principles (usability in turn is a fundamental acceptability factor) [13-14]. Effectiveness for design documentation is a requirement explicitly addressed by many successful software engineering methodologies, such as UML [15].

order to produce incremental versions of the application design specifications, and to help designers improve their design solutions in an progressive, incremental way") is the preferred one for a design methodology or, in alternative, which is the desired one (or the one which is currently used in the company).

□ Fast prototyping

"The support offered by the methodology for producing an application prototype in a rapid way, in order to come to the client with fast results and to obtain early feedback". This set of questions (see also figure 2) aims at investigating the need for deriving early prototypes of the application once various versions of the design are produced, at understanding why fast prototyping is required; and at identifying the desired "type" of prototype.

Project Management Support

"The managerial features that a methodology should support for planning, communication, resources and client management, configuration management, etc". The objective is to verify the need for project management support, and to identify the key aspects of project management that industry people consider important in a methodology in order to improve the control and effectiveness of the design process.



Fig. 2. Detailed questions about model and methodology requirements

The **second section of the questionnaire** focuses on "Requirements for Documentation". It aims at establishing the role of good quality explanatory and self-training documents to make a model and a methodology easier to learn. Other objectives of this section are to identify the most useful types of documentation

required by industry people, their format and structure, and their intended use. We propose different types of documentation (Hand Book, User Guide or Manual, Cookbook, A book on the methodology, On-line hypermedia documentation) and ask a four-levels ranking (*Not Desired at all, Desired, Strongly Desired, Absolutely Necessary*). For each documentation type, we also require respondents to provide an estimation of the size.

In section three – "Requirements for Support Tools" - we focus on the requirements for the tools that support the design activity and their efficient integration with the entire development process. We ask the respondents to evaluate the following characteristics for design tools (using four values: *Not Desired at all, Desired, Strongly Desired, Absolutely Necessary*):

- □ Flexible models management. For example, the flow of design activities is not strictly sequential. A designer may need to switch to navigation design before completing information design, to define some presentation solutions, and to return back to complete navigation specifications, and similar. The objective of this set of questions is to verify whether it is important that the tool allows designers easily switch back and forth among the different design tasks, and what is the expected degree of flexibility.
- □ **Model Versioning.** This set of questions aims at verifying the need for support in the management of different versions of the design specifications of an application which are produced by different authors or at different design stages, and to identify the best way to meet this requirement.
- □ Code derivation. This expression denotes the tool ability to generate source code fragments in a specific implementation language from the design specifications with the tools (e.g., class templates, method templates, and similar). The objective of these questions is to understand several aspects: how crucial is this feature for industrial production? In which cases is it more valuable and in which ones is it less important? Which trade-offs is the designer ready to pay to get some source code, in terms of effort during the design specification phase⁵?
- □ Semi-Automatic generation of prototype. by this we mean the provision of special tool features for creating an application prototype from the design models.
- □ Integration with design methodology. This set of questions explores the expectations concerning the degree of adherence of the tool to the specific design model and methodology. We want to understand whether the interviewees need a tool which is strongly tailored to the model and methodology, or rather prefer a general purpose CASE tools which can be personalized to the specific features of the model. We also explore the degree of degree of model-tailoring respondents require, and which amount of personalization effort they can accept.
- □ Multiple view of the same design artifact. These questions investigate the need for view features that allow designers dynamically restructure a set of design artifacts according to different perspectives. For example, to view the different portions of design specification that addresses the needs of different user categories, or the constraints of different devices; to view the design specification (in terms of information structures, navigation and publishing structures) for a set of content objects, etc.

⁵ Derivation of source code requires in fact a very detailed and formal design specification.

- □ **MS-Windows look and feel.** This part of the questionnaire aims at discovering whether industry people desire a standard MS Windows-like look and feel, or rather prefer different interface paradigms for the design tools interface.
- Semi-automatic derivation of documentation. This part of the questionnaire addresses the requirements on the production of design documentations. In our experience, good quality design documentation is crucial both for managerial reasons (being sometimes the contractual basis for discussing the development follow-up with the customer) and for implementation (to avoid misunderstandings with the implementers). Through these questions, we want to understand whether companies share our point of view on the role of design documentation, and to verify how much they expect that a design tool supports the (possibly partial) generation of well structured design reports from design specifications built using the tool.
- □ **Consistency Check.** These questions verify the relevance of tool features for checking the consistency of the design specification, and for reporting consistency violations (such as a missing cardinality in a relation, a missing attribute in a information structure, a dangling or partially defined link, and similar).

3 Findings Analysis

The questionnaire was sent via e-mail to 62 companies involved in large-scale web application development: 11 organizations in North and South America, and 51 in Europe (from 8 different Countries). We purposefully excluded academic and research institutions from the sample of inspected subjects. The questionnaire was filled in by web project managers, developers, or designers. We had a 44% percent of responses. The findings, based on the statistic analysis of the answers and cross-tabulations, are discussed in the rest of this section.

3.1 Requirements for the Model and the Design Process

Figure 3 summarizes the answers to the questions presented in figure 1.



Fig. 3. Summary results concerning general requirements for a design model and design process (see questions of figure 1)

The most evident results highlighted by the above diagram are that:

- □ 40.74% marks as absolutely necessary to provide Support for Iterative and Incremental Design Lifecycle
- □ 48.15% marks as *strongly relevant* the fact that the methodology should be *Easy to Use*
- □ 44.44% considers as *relevant* that the methodology is *Easy to Learn*
- □ 66.67% says that they do not care about the methodology to be a standard

Other interesting findings emerging from our analysis are:

- Dependence of the Project Management Support is relevant for the 37.04% of the respondents
- □ *Process customization* is relevant for the 48.15% of the respondents, and *absolutely necessary* for the 25.93% of the respondents
- □ *Fast Prototyping* is relevant for the 33.33% of the respondents, while 40.74% considers it *strongly relevant* and 22.22% considers it *absolutely necessary*
- □ For *CASE Tools Support*, the answers are uniformly distributed on the different measurement values, but we can remark that the only 25.93% of the respondents mark as *Not Relevant* this characteristic
- □ Documentation Support is marked as relevant by the 33.33% of our sample, strongly relevant by the 37.04% of the respondents, and absolutely necessary by 29.63%, but the most important, none marks it as Not Relevant

The analysis of the detailed answers (see examples in the previous figure 2) provides an additional set of useful data:

Support for Iterative and Incremental Design Lifecycle and Fast Prototyping

As mentioned above, both characteristics are considered as absolutely necessary and relevant for a significant portion of our sample. There are a significant 85% of the respondents that *prefer an evolutionary prototype* rather than a *throwaway prototype*, which is *preferred* only by the 15%. This high preference for a prototype that evolves until becoming the final application can be justified by the fact that industry people do not want to lose resources in working on a system (the throwaway prototype) that they will have to discard later. Concerning fast prototyping, almost all (92,59%) say that *requirements validation* is the main reason for fast prototyping, while 51.85% indicates design validation, and 48.15% choose *rapid client satisfaction*.

Easiness of learning

The results show that the 70% expects spending between two and four weeks learning a model and a methodology, 19% prefers spending at most one week, and only 11% can spend more than four weeks. The results on the preferred type of training highlights that 48.15% desires courses with theory and practical information, 40.74% mentoring courses (with the expert side-by-side support), and 33.33% online web courses.

Easiness of use

Among the characteristics that make a model easy to use, 51% indicates *customizability* (the ability to adapt its use to different contingent and organizational situations for design.), 66% *flexibility* (the ability to provide different ways by which the designer can use the model) and *more than* 70% the *presence of guidelines and patterns*. The latter result empirically confirms a generally acknowledged principle of software engineering - the utility of design patterns - highlighting that patterns are largely perceived as useful by the industry to improve the usability of a design model and to make the design activity easier and more effective.

Project management support

To the question "Which are the project management activities that are considered as important to be supported by a design methodology?", respondents answer that *Time Planning* (70,37%), *Change Management* (74.07%) and *Configuration Management* (77.78%) are the most voted.

3.2 Requirements for Documentation Support

Figure 4 summarizes the main findings regarding the section on documentation support.



Fig. 4. Requirements on documentation types.

The above diagram highlights that online hypermedia is the most required form of documentation, being marked as *strongly desired* by the 44.44% of the respondents, and as *absolutely necessary* by the 32%. In contrast, *cookbook* receives the highest percentage (22%) of *not-desired*, and has slightly lower values for *strongly desired* and *absolutely necessary* than the other forms of documentation. The results about *book* show that this form of documentation does not have any dominant attribute. The respondents' opinions seem evenly dispersed across all choices.

Regarding the desired size, in terms of amount of pages, of each proposed documentation type, the main results are:

- □ *Handbook*: 5-10 pages: 19%; 10-20 pages: 70%; more than 20 pages: 11%
- □ User Guide Manual: 40-50 pages: 33%; 50-80 pages: 37%; more than 80: 30%
- □ *Cook Book*: 10-20 pages: 35%; 20-40 pages: 42%; more than 40 pages: 42%
- □ *Book*: 70-100 pages: 33%; more than 100 pages: 67%
- □ On-line Hypermedia: 20-40 pages: 59%; more than 40 pages: 41%

3.3 Requirements for Design Tools

The most interesting results of this section is that 70.37% of the respondents answers "*YES*" to the general question on the utility of software tools for supporting the design process. Figure 5 summarizes the main findings regarding the different characteristics desired for support tools.

The most important results to observe are:

□ 51.85% marks consistency check as Absolutely Necessary, while Model Versioning gets the same vote by the 29.63% of the sample, followed by Semiautomatic derivation of documentation (26%). All other characteristics get the same vote by less than 15% of the sample.



Fig. 5. Requirements for a design tool

- Five characteristics are indicated as *Strongly Necessary* with relatively similar percentages of votes (between 40% and 55% of the respondents): *Flexible Model Management* 48.15%; *Code Derivation* 55.56%; *Model Versioning and Integration with Methodology Design Activities* 48.15%; *Semi-automatic Derivation of Documentation* 40.74%
- □ Multiples views of the same Design Artifacts is considered as desired by 40.74 %
- □ Regarding *Semiautomatic Generation of Prototypes*, it is interesting to note that although it is signaled as *strongly desired* and desired by 37.04% and 40.74% of the votes respectively, is also signaled as not desired by 22.22% of the responses
- □ The most important conclusion regarding what people don't want is achieved by the *MS-Windows Look And Feel* feature, which gets a *not desired* by 44.44% of the respondents

4 Lessons Learned and Conclusions

What does "quality" mean for a conceptual design model? Quality is a very broad and generic term, which can be defined along many different perspectives. In this paper, we suggest that a possible measure of quality for a conceptual design model is its degree of *acceptance* in the practitioners' world. Our research investigates some

factors that contribute to the industrial acceptability of web conceptual design models, by examining the requirements of a significant sample of software companies, internet services providers, or organizations which are moving their business towards the web. The results of our research validate our general hypothesis: being a "good" conceptual model for web application design is not the only relevant factor for industrial acceptability. A good set of modeling primitives and notations should be delivered to industry together with a number of complementary features: a proper methodology and design process, an effective documentations about the model and the methodology, a kit of support tools.

For the model per se, and for each complementary feature - the methodology, the documentation, and the tools – we summarize the main lessons learned from our study.

Acceptability features for a conceptual model "per se"

The two important characteristics that industry people have identified as most relevant for a model per se are ease of use and learnability (crucial factors for the usability of any human artefact). Our findings on learnability suggest that the model (and the companion methodology - see below) has to be learned in no more than 4 weeks, and that theory and practical courses combined with mentoring courses (better if including hands-on activities carried on side by side with an expert) are the preferred learning mechanisms. We may conclude that to improve both ease of use and learnability, our models should find a compromise between richness and simplicity, and should try to balance completeness and expressive power of the modelling primitives with intuitiveness and with evidence of their utility. A possible way to achieve this compromise might be to deliver "multi-version" models, made of a "basic kit" and an "advance kit" of modelling concepts and notations. The basic kit can be understood and learnt relatively easily (2-5 days) and can be almost immediately applied for the design of relatively simple applications. The advanced version can address more sophisticated modelling needs, can be learnt after the basic modelling features are fully digested, and used to design complex application features.

Acceptability features for a methodology

A model should be integrated with a proper methodology, which identifies a systematic design process and provides a clear set of guidelines to help designers use the model. The design process should be flexible, incremental, iterative. It should be customisable to different scenarios of use (i.e., to the needs of each specific application and to the individual industrial practice). The design process should be integrated with the whole development process. In particular, it should provide some support for the human translation of conceptual design solutions into implementation solutions, and for fast prototyping production. Prototypes should be evolutionary and should basically help designers to validate user requirements. In addition, the methodology and the process should look at application design and development *within the organizational context:* they should address *project management* issues, to help managers monitor the project lifecycle in terms of time and resources.

Acceptability features for documentation

Good quality documentation is crucial for making a model and a methodology easy to learn and to use in an industrial setting. Our study points out that the preferred documentation support is online hypermedia documentation, followed by user guides and manuals. Richness of examples, case studies, and lesson learned is perceived as an important content to be provided in all forms of documentations.

Acceptability features for support tools

An important highlight of our study is that the availability of CASE tool and fast prototyping tools has a *very high priority* for the world of practitioners. The most desired feature for a CASE tool is a consistency checker (to support a design task where humans are less effective than machines). Additional requirements concern facilities for multiple views of the specification schemas (e.g., at different levels of details, along different design perspectives), support to versioning, possibility of switching among different design schemas. Concerning fast prototyping, there is a need for support to code derivation and for a strong integration of (semi-automatic) prototyping facilities with the representation tools. It is interesting to note that the main reason for fast prototypes is requirements and design validation, which suggests that requirement traceability support is an additional useful feature for CASE tools

Even if the findings of our research reflect the needs and inclinations of a specific industrial sector, they have a general validity. In principle, they may offer interesting highlights on the industrial requirements of *any* design model, also in fields other than the web. In the short term, our future work includes a refinement of our study and an accurate validation of its results. We will enlarge the answer set and adopt more sophisticated evaluation procedures, for checking and correcting errors in the results, and for approximating missing answers. In the mid term, we are using the research findings to improve the features of the web design model (W2000 [12]), documentation, and toolkit, developed within the UWA project, in order to make it more acceptable and potentially usable in the industry practice.

Acknowledgments. Authors are grateful to Mauricio Sansano, who carried on the preliminary version of this study, to the UWA project partners and reviewers for their helpful suggestions, and to the anonymous reviewers for their valuable feedbacks.

References

- 1. Norman E. Fenton: Software Metrics: A Rigorous Approach. London: Chapman & Hall, 1991.
- 2. Barry and Lang: A Survey of Multimedia and Web Development Techniques and Methodology Usage. IEEE Multimedia, April–June, 2001
- 3. C. Britton et al.: A Survey of Current Practice in the Development of Multimedia Systems. Information and Software Technology, vol. 39, no. 10, 1997, pp. 695–705.
- 4. B. Fitzgerald: An Investigation of the Use of Systems Development Methodologies in Practice. Proc. Fourth European Conf. Information Systems, Lisbon, Portugal, 1996,
- F. Garzotto, P. Paolini, D. Schwabe: HDM A Model-Based Approach to Hypertext Application Design. ACM Transactions on Information Systems, Vol. 11, No. 1, January 1995.
- 6. S. Ceri, P. Fraternali, A. Bongio: Web Modeling Language (WebML): a modeling language for designing Web sites. Proc. of the 9th World Wide Web Conference (WWW9), Amsterdam, 2000.
- 7. T. Isakowitz, E. Stohr, P. Balasubramanian: RMM: A Methodology for Structured Hypermedia Design. CACM (1995), 38(8), pp. 34–44
- 8. D. Schwabe, G. Rossi: An Object Oriented Approach to Web-Based Application Design. Theory and Practice of Object Systems, 4 (4), J. Wiley, 1998.

- 9. Rolf Hennicker and Nora Koch: A UML-based Methodology for Hypermedia Design. In volume 1939 of Lecture Notes in Computer Science, York, England, October 2000. Springer Verlag.
- J. Conallen: "Modeling Web Application Architectures with UML. Communications of the ACM, 42:10, pp. 63–70.
- 11. UWA ("Ubiquitous Web Applications") project IST 2000–25131 –, technical annex 1 "Description of Work". Official Web site: www.uwa-project.org
- 12. L. Baresi, F. Garzotto, P. Paolini, V. Perrone: UWA Deliverable D7, "Hypermedia and Operation Design". www.uwa-project.org
- 13. Alan Dix et al. Uman-Computer Interaction. Prentice Hall 1998.
- 14. Jakob Nielsen: Designing Web Usability: The Practice of Simplicity. New Riders Publishing, Indianapolis, 2000
- 15. G. Booch, I. Jacobson, and J. Rumbaugh: The Unified Modeling Language User Guide. The Addison-Wesley Object Technology Series, 1998.