An Agile Hypertext Design Methodology

Gary B. Wills, Noura Abbas, Rakhi Chandrasekharan, Richard M. Crowder, Lester Gilbert,
Yvonne Howard, David E. Millard, Sylvia C. Wong and Robert J Walters
School of Electronics and Computer Science, University of Southampton
Southampton, SO17 1BJ, UK, +44 (0) 23 8059 2831
{gbw, na06r,rmc, lg3, ymh, dem, sw2, rjw1}@ecs.soton.ac.uk

ABSTRACT

Customers are driving down lead times for software, especially for Web applications, to only a few months. While a number of hypertext design models exist, they do not address the issue of the requirements and analysis process that normally feeds the design process. In this paper we present an agile approach to developing hypertext applications, which focuses on the requirements and analysis stages, something that is largely ignored in current methodologies.

Categories and Subject Descriptors

D.2.2 Design Tools and Techniques.

General Terms

Documentation, Design.

Keywords

Web Design Methods, Hypermedia Design, Semantic Web, Web Service Design.

1. INTRODUCTION

Customers are driving down lead times for software, especially for Web applications, to only a few months. In general, to mitigate the short lead time and yet maintain quality organizations are favouring an iterative and incremental approach to software engineering.

Organizations are also amassing more and more information, often in heterogeneous databases, so designing hypertext systems within such short lead times necessitates an unambiguous design and an effective but reduced set of design documentation; this combination will ensure good communication between customer and team, leading to a high quality product. Hypertext aims to mimic the way people think by capturing the association between resources as links. Large industrial hypertext systems bring their own set of problems as first identified by Malcolm *et al* [15]. Some of this has been addressed in earlier work by Heath *et al*. [9], where it was shown how structural links can be generated from the structure of documents. However, for the associative linking the solution proposed used human authored links and this is not practical for large industrial hypermedia applications.

There is also a move away from the traditional monolithic or tightly coupled applications into a more loosely coupled Service Oriented Architectures (SOA) [21]. SOA implement the functionality of the system using Web services. SOAs are an attempt to modularise large complex systems in such a way that they are composed of independent software components that offer services to one another through well-defined interfaces. This supports the notion that any of the components could be 'swapped' for a better one, when it becomes available.

Ontological hypertext and SOA add additional complexity to Hypermedia Design [27]. A more holistic design process is needed, that ties together the underlying knowledge, behaviour and presentation. Ontological hypertext, overcomes this problem by generating the associations between resources based on the semantics of their content [4]. The ontologies can be derived from the document structure, the tasks being undertaken and the roles of the people using the application.

This paper will first assess current design models and examine their shortcomings regarding large scale hypertext applications. The paper will then present the model we use to develop industrial hypertext applications with short lead times and concludes with a discussion of the work.

2. RELATED WORK

In this section we present the main design methodologies on which this work is based.

RMM (Relationship Management Methodology) [10] an is entity-relationship based method, suitable for structured hypermedia applications with volatile data.

HDM (Hypermedia Design Model) is an early E/R-based design model proposed by Garzotto *et al*. [7] to define the structure and interactions in large scale and read-only hypermedia systems. The model is suitable for domains with a high level of organisation, modularity, consistency, and describing the information objects in terms of entities.

OOHDM (Object Oriented Hypermedia Design Model) is an OO-based design model that allows the specification of hypermedia applications as navigational views over the conceptual model [18]. Its design process consists of four main dimensions and has recently been extended to cover formally requirements gathering [8] and personalisation modelling [17]. Navigation units or nodes are mapped to conceptual classes, and the design and generation of OOHDM-based read-only web sites is supported by a CASE tool called OOHDM-Web [19]

Conference'04, Month 1–2, 2004, City, State, Country.

EORM (Enhanced Object-Relationship Model) [11] is an OO-based methodology whose major characteristic is the representation of relationships between objects (links) as separate objects.

SOHDM (Scenario-based Object-oriented Hypermedia Design Methodology) [13] is another OO-based approach focusing on process-oriented hypermedia systems to support organisational processes.

WSDM (The Web Site Design Method) [6] is a user-centred approach in which the application model is based on the user model, identifying user classes and their preferences and views.

OntoWebber is an ontology-based approach to building read-only web sites, focusing on integrating heterogeneous data sources to build data-intensive web portals [5, 23]. Nodes or navigation units are called cards (mapped to ontology classes via the content model) and the overall design is represented by an XML-based meta-schema using RDF and DAML+OIL [23]

WebML (Web Modelling Language) is a high level, model-driven, and E/R-based (but compatible with UML class diagrams) design approach allowing a conceptual specification and automatic implementation of data-intensive web sites [3]. WebML extensions [2] allow interactive content management with entry units to update the site content, and WebML has a CASE tool called WebRatio.

So these design methods are largely concerned with presenting/publishing data and not necessarily designed to manage the content. These methodologies and models take a simple layered approach, separating the design issues so as to allow independence for:

- Mapping the domain, in terms of its structure, content, work flow, *etc*.
- Analysing the associations and relation in that domain
- Presenting the information to appropriate users

Several limitations persist with regard to content modelling and management and resulting linking capabilities within document. Wills *et al.* [20] suggested the use of ontological hypertext in the design process as a means of moving between these layers and also making explicit the relationships between the semantics within the text. As a result, designers can extend their influence into the texts to influence the production of semantic content. The Semantic Web [1] augments the Web with explicit statements of document semantics, allowing the Web to be used for more than a human-browsable repository of information. The meaning of the published documents, knowledge about their authors and the reasons for their publication are all used to infer contextually appropriate associations, *i.e.*, knowledge.

2.1 Service Oriented Architectures

Service-Oriented Architectures (SOAs) enable large complex systems to be mutualised, that is, composed of independent software components that operate through well-defined interfaces. A service approach is ideally suited to more loosely coupled systems, where individual parts may be developed by different people or organizations. Wilson *et al.* [21] discuss in detail the advantages of using a SOA; the ability to couple dynamically services, interoperability of services due to clearly define standards, and as a result the ability to avoid technology 'lock-in'. Due to the nature of the loose coupling in a SOA, applications can be developed and deployed incrementally. In addition, new features can be added easily after the system is deployed. This modularity and extensibility makes an SOA especially suitable in situations with evolving requirements and standards.

2.2 Shortcomings of Existing Models

Overall, most of current web design models provide users with model-driven approaches for the systematic design of high-level, read-only, well-organized, and easy to maintain web applications in different domains. Their coverage of the application life cycle focuses on the design stage with different levels of support provided at different phases within the design process.

None of the current hypertext design methods addresses in detail the requirements and analysis of a system prior to design. We believe that a more holistic approach will ensure that the knowledge model, behaviour specifications and presentation formats are all aligned. For example, the use of participatory or co-design is not explicitly mentioned in any of the models. Yet these methods can provide the alignment that we need, and end users will buy into the implementation if they are involved in the design.

3. AN AGILE DESIGN METHOD

In this section we first explain the components of the method and then explain how they are combined using an agile approach. The overall process is shown in Figure 1.

Personas and Scenarios

Personas and scenarios are a lightweight method of capturing and recording the requirements of the system from an end user's view point [25]. A persona describes one of the end users in some detail, their background, job function, situation in the organization. In some cases we have used personas to describe variants of a particular role, for instance, the ideal employee, the laid back employee, the anxious employee. Scenarios are textual descriptions of how the persona interacts with the system interacts with other personas when using a system. The scenarios are independent of any technology and they may represent either current practice or improved practice (as-is or to-be).

The process relies heavily on end user involvement, that generally leads to user buy-in. Persona and scenario based modelling is commonly used in software engineering, but we have found that the end users can find the process a little confusing. However, by walking them through previous examples, giving them templates, and through the use of co-design workshops they can help produce detailed personas and scenarios.

Resources and Ontologies

Identifying and locating the multimedia resources is a significant component of the hypertext authoring process [14]. The structure and use of the resource also inform the development of the ontologies. Both the identification of resources and ontology building need the knowledge of end users and stakeholder. During this process the scenarios can be embellished with details of how the resource may be used and information they supply.

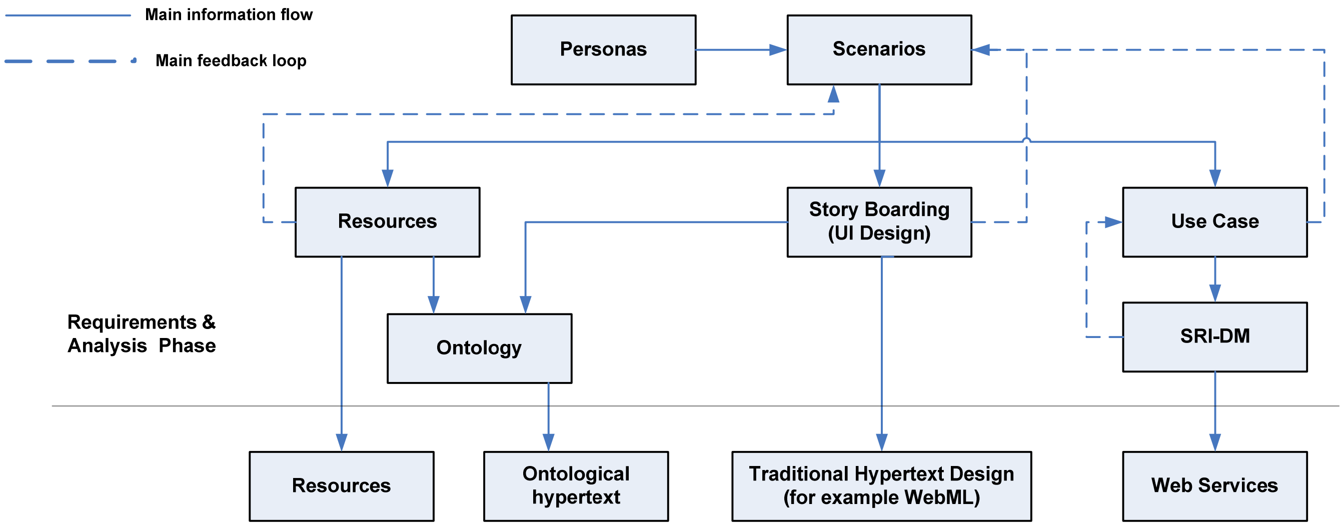

Figure 1. The Agile Hypertext Design Method.

Story boarding

Using the scenarios, a number of story boards can be created to represent the user interface design (UI). This is a standard technique used in HCI development and is effective when used in a participatory (or co-design) process. This brings together the end users and the HCI experts, to designing the UI. During this process, the scenarios and use cases can be clarified and modified if required.

UML Use Cases

These are constructed from the scenarios. The use cases are developed mainly by software engineers, with input from the users. These are standard UML use cases, consisting of a use case diagram, with success scenarios for each case. We have chosen Use Case diagrams because they are abstract and implementation independent. A brief narrative description is held alongside the diagram as a whole, as well as for each individual use case. This description helps to disambiguate the use cases, explains the roles of the different actors associated with that use case and focuses at a high level on what each use case involves. From an agile point of view they are also effective because they are relatively informal, yet help to define and capture a problem space in enough detail and can be understood by the whole team, including the end users.

Web Service Design

Web service design uses a method called The Service Responsibility and Interaction Design Method (SRI-DM) [16]. SRI-DRM separates abstract representations of Services from their implementation.

Service Profiles are abstract descriptions of a service that may be fulfilled by several different *Service Implementations* that may each expose different concrete interfaces. Service Profiles are thus modeled in an abstract way that does not prescribe a data model or dictate explicit methods. To do this we use Service Responsibility and Collaboration cards (SRCs) based on Class Responsibilities/ Collaborations, a modeling technique first described by Beck and Cunningham for eXtreme Programming (XP) [26]. The SRCs do not show how services will be combined in a wider scenario, but do model possible collaborations with other services that might occur for this service to fulfill its own specific responsibilities

At the scenario level, services represented by SRCs must interact with each other to fulfil a wider purpose. These interactions are complex and include transactions, sequences and state. If the scenario modelling is to maintain the high level of abstraction necessary for agile development then it would be inappropriate to declare a detailed data model, or to specify the logic of the communicating services. So we use UML 2.0 Sequence Diagrams to represent the interactions, showing which services should communicate and in which order, and containing enough description to show how the individual services are responsible for moving and processing data, specifying the detail of the data model or the decision making logic.

3.1 How is it Agile?

Agile methods are a number of software development methods which were proposed in the mid 1990s as a reaction to the limitations of traditional software development methodologies. Although these methods vary in practice, they share common principles such as [12]:

- Deliver working software frequently within a short timescale
- Close communication within the team and with the customer
- Simplicity
- Programming over documenting

In this method, we have gone for limited documentation that still ensures effective communication within the team and to the customer. Figure 1 shows what may be perceived as a linear model. However, through the feedback loop, the developers are continuously improving the scenarios as the requirements are refined. In addition the use of Web service provides the rapid and flexible response to change, in that the complexity of the functionality can be delivered incrementally and at different iterations.

4. CONCLUSIONS

There is a move away from the tightly coupled systems into a more Service Oriented Architecture approach. In addition there is a downward pressure on leads times for quality software generally. While there exists a number of models and methodologies coving the design of the hypertext systems, the area of requirement analysis is least investigated

In this paper we have proposed an agile approach to model the requirements and analysis within current design models and methodologies. The agile approach gives us lightweight method of capturing and recording the requirements.

This paper is based on our experience over a number of years of producing hypertext/Web applications with short lead times in large industrial and related hypermedia projects, including the mPLAT [24], FREMA [16] and Semantic Web IPAS [22] projects.

5. REFERENCES

[1] Berners-Lee, T., Hendler, J., Lassila, O. (2001) The Semantic Web, Scientific American, May 2001 34-43

[2] Bongio, S. Ceri, P. Fraternali, A. Maurino (2000). Modelling data entry and operations in WebML. WebDB: 87-92

[3] Ceri S, Fraternali P, Bongio (2000) A: Web Modelling Language (WebML): a modelling language for designing Web sites. In Proceedings WWW9 Conference, Amsterdam, May .

[4] Carr, L., Miles-Board, T., Woukeu, A., Wills, G. and Hall, W. (2004) The Case for Explicit Knowledge in Documents. In Proceedings of ACM Symposium on Document Engineering.

[5] Carr. L., Kampa. S., and Miles-Board. T., (2001). MetaPortal Final report: Building Ontological Hypermedia with the Ontoportal Framework. Technical Report, IAM, ECS, University of Southampton, March 2001.

[6] De Troyer O. and Leune C. (1997) WSDM: A user-centered design method for Web sites. In Proceedings of the 7th International World Wide Web Conference, 1997.

[7] Garzotto F., Paolini P. (1993) HDM- A Model-Based Approach to Hypertext Application Design. In ACM Transactions on Information Systems, Vol. 11, No1 January p1-26.

[8] Guell, N., Schwabe D., Vilain, P. (2000): “Modeling Interactions and Navigation in Web Applications”, Lecture Notes in Computer Science 1921, Proceedings of the World Wide Web and Conceptual Modeling ’00 Workshop, ER’99 Conference, Springer, Salt Lake City, 2000. ISBN 3-540- 41073-2.

[9] Heath I., Wills G., Crowder R., Hall W., Ballantyne J..(2000) A New Authoring Methodology for Large-Scale Hypermedia Applications. IEEE Multimedia Tools and Applications 12(2/3): 129-144, November

[10] Isakowitz T, Stohr EA., Balasubramanian P. (1995) RMM: A design Methodology for Structured Hypermedia Design. In Communications of the ACM 38 (8) August pp 34-44.

[11] Lange DB. (1996) An Object-Oriented design Method for Hypermedia Information Systems. In Journal of Organizational Computing & Electronic Commerce, 6 (3).

[12] Larman, C. (2004). Agile and Iterative Development: A manager’s guide. Pearson Education.

[13] Lee H., Lee C. and Yoo C. (1998) A scenario-based object-oriented methodology for developing hypermedia information systems. In Proceedings of 31st Annual Conference on Systems Science, Eds. Sprague R., 1998.

[14] Lowe D, Hall W, Hypermedia Engineering, the Web and Beyond, Wiley 1999.

[15] Malcolm KC, Poltrock SE, Schuler D.(1991) Industrial Strength Hypermedia: Requirements for a Large Engineering Enterprise. Proceedings Hypertext '91

[16] Millard, D. E., Howard, Y., Jam, E. R., Chennupati, S., Davis, H. C., Gilbert, L. and Wills, G. B. (2006) FREMA method for describing Web Services in a Service Oriented Architecture. Technical Report ECSTR-IAM06-002, ECS, University of Southampton.

[17] Rossi G, Schwabe D, Guimaraes R. (2001) Designing Personalized Web Applications. In proceedings WWW10, Hong-Kong, May

[18] Schwabe D, Rossi G, Barbosa SDJ. (1996) Systematic Hypermedia Design with OOHDM, Proc. Of the Hypertext’ 96, Washington, March.

[19] Schwabe D, . de Almeida Pontes R, Moura I. (1999) OOHDM-Web : An Environment for implementation of Hypermedia Applications in the WWW. SigWEB Newsletter, Vol. 8, #2, June de 1999

[20] Wills G, Woukeu A, Carr L, and Hall H. (2003) The need for semantic in web design. Semantic Web workshop, Fourteenth Conference on Hypertext and Hypermedia Nottingham, UK, August 26, 2003

[21] Wilson S., Blinco K., and Rehak D. Service-Oriented Frameworks: Modelling the infrastructure for the next generation of e-Learning Systems. JISC, Bristol, UK 2004.

[22] Wong, S. C., Crowder, R. M., Wills, G. B. and Shadbolt, N. R. (2006) Knowledge Engineering - From Front-line Support to Preliminary Design. In *Proceedings of **ACM** Symposium on Document Engineering (DocEng)* , Amsterdam, The Netherlands.

[23] Yuhui J, Sichun X, Stefan Decker. (2002) OntoWebber: A Novel Approach for Managing Data on the Web. In the 8th International Conference on Extending Database Technology (EDBT 2002). Prague, Czech Republic, March 24-28, 2002.

[24] Davis HC , Gilbert L, Howard YM, Millard DE Wills GB (2007) m-Plate project http://www.mplat.ecs.soton.ac.uk/

[25] Cooper, A. and R. Reimann (2003). About Face 2.0: The Essentials of Interaction Design, John Wiley & Sons.

[26] Beck, K. (2000) Extreme Programming Explained: Embrace Change. Addison-Wesley.

[27] De Bra P., Aroyo L., Chepegin V. (2004) The Next Big Thing: Adaptive Web-Based Systems. Journal of Digital Information, Vol 5 No 1 Article No. 247.