Issues of Usability in Hypermedia Systems: A Usability Estimation Questionnaire for Hypermedia Designers

Nikos Tsirakis
Research-Academic
Computer Technology Institute
Dept. of Computer
Engineering & Informatics
University of Patras 26500
Rion, Greece
tsirakis@ceid.upatras.gr

Nikos Karousos
Research-Academic
Computer Technology Institute
Dept. of Computer
Engineering & Informatics
University of Patras 26500
Rion, Greece
karousos@optionsnet.gr

George Gkotsis
Research-Academic
Computer Technology Institute
Dept. of Computer
Engineering & Informatics
University of Patras 26500
Rion, Greece
gkotsis@ceid.upatras.gr

ABSTRACT
Open Hypermedia have come up with several systems, which can provide services to software developers. This paper argues that usability is a critical attribute that should be taken into consideration during the development process of an Open Hypermedia System (OHS). An analysis of the usability issue from a developer's perspective is carried out. Moreover, targeting at a usability estimation technique, a questionnaire for OHSs designers is being proposed. Finally, some evaluation results of numerous existing systems are being presented and discussed.

Keywords
usability, hypermedia, hypermedia development

1. INTRODUCTION
Nowadays hypermedia systems are aiming at global acceptance by making use of heterogeneous services, allowing to the applications to interoperate through different environments. In order to achieve the above goal, developers should feel free to make use of hypermedia systems productively.

OHSs support both knowledge workers (end-users) and hypermedia unaware developers (intermediate-users) by providing them hypermedia functionality. End users usually handle hypermedia systems through a graphic interface of a hypermedia client application, while developers make efforts to use or integrate hypermedia services into their applications. Although the usability factor from the end-user perspective is being tracked down, developers' requirements for easily used hypermedia services have not been addressed to sufficiently.

Targeting at the increase of OHSs usage, developer's support during the use of hypermedia services can be enforced by both taking into consideration developers' needs and designing usable hypermedia systems. Hence, in this paper we focus on the usability factor in hypermedia systems through a developer's perspective and we propose a usability estimation procedure for hypermedia designers. By applying this procedure we try to discover how easily some existing hypermedia services can be used by the developers and what are the designing issues that can improve the usability factor.

Moreover, a study of usability factor from the developer's perspective and a proposal of a usability estimation questionnaire for OHSs designers follow. In this context an interpretation of the fundamental usability attributes into a set of questions for hypermedia designers (based on the developers' needs) is presented. Finally, some interesting results on the maturity level of usability which have been verified through feedback provided by some OHS designers are highlighted.

Metainformatics Symposium, November 9–11, 2005, Esbjerg, Denmark.

2. USABILITY FACTOR IN OHSS
2.1 Definition
ISO 9241-11 [7] describes usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. A usable system is designed to consider all the aforementioned features.

Since usability is a term too abstract to be studied directly, it is usually divided into the following terms identified by Nielsen [13]:

1. *Learnability*: It's a measure of how much training is required before a specified level of proficiency is reached. (A1)
2. *Efficiency*: The attribute of a system that allows to the developer to maximize productivity. (A2)
3. *Memorability*: This term shows how easily a developer recalls how to reuse a system after a period of time and measures how robust the learning and performance are. (A3)
4. *Errors*: It's a measure of the accuracy of the work carried out to complete tasks. (A4)

5. *Satisfaction*: It's the feeling of the developer after the use of a system. (A5)

Usability, during this research, is analyzed from the developer's point of view. End-users and developers are two meanings that some times coincide. In our case we deal with developers who can be considered users of hypermedia systems and use hypermedia services or components. Alternatively, those developers are also called B-Level knowledge workers[1].

In order to achieve better results, we argue that usability needs to be removed from realms of user ploy, and elevate to the status of development aid. In other words, it can become part of the development process of a system. This process, called usability engineering, interferes effectively with several engineering processes, providing specific methodologies and tools.

2.2 Designing Usable OHSs

Although, there is work to be done in how developers can be supported, the usability issue was mostly targeting towards the end-users of hypermedia systems. Many efforts, such as [6], have focused on the provision of user-friendly interfaces that allow the presentation and organization of information in an effective manner. Furthermore, OHSs have satisfied the demands of a service oriented system, namely openness, interoperability, etc [1]. However, the developer is accounted a user familiar with the disciplines embodied within the OHS community. This results in the fact that these systems are designed in a way that the use of a hypermedia service is not founded on a developer-oriented approach.

Nowadays, designers perceive usability engineering as a critical part of engineering. Usability engineering methods can help to support usability measures at low cost [13]. Both user testing with real users and questionnaires are the most common usability evaluation methods. Thus, many aspects of usability can be derived by simply asking the users.

However, the fact that many usability problems must be solved in design-time stimulated the current research towards a questionnaire for designers. This questionnaire arises from our experience on developer's requirements [9] and is applied to designers as a pre-estimation tool for helping them to make usable systems.

It could be argued that the questionnaire should be given to developers, who are actually using these systems and have been dealing with actual usability issues, than to designers. However, considering the research and empirical results of the needs that developers have, we introduce this questionnaire as a first step towards an examination of the systems. Hence, it can be deduced that using this questionnaire at the initial phase of design we can achieve direct usability impacts.

[1]According to Douglas Engelbart [5], there are three types of work that can be performed in an organization. The A-Level is the work of the organization itself. The B-Level is work that develops tools to improve the ability of people performing A-Level work, while C-Level is work that develops tools to augment the ability of people performing Blevel work.

2.3 A Usability Estimation Questionnaire for OHSs Designers

With respect to the end-user needs[2], we analyze five usability attributes from the developer's perspective and proceed towards an interpretation of each attribute to a number of questions that concern usability engineering during OHSs development. Thus, the questionnaire should focus on the development tasks and try to highlight the usability attributes. It should be noted that the questionnaire is not aiming at a general evaluation of the hypermedia system.

Efficiency is an attribute that compounds both feasibility and flexibility. Developers make use of an OHS should feel free to adopt different design principles that would not prevent productivity. *Learnability* is considered to be the ability of the developer to adopt proposed techniques, with low time and effort. The above includes the ability to make use of widely adopted technologies and methods and the prevision for simplicity of OHSs. *Memorability* is a feature that differs from learnability in a way that it also combines the ability to become familiar with the architecture and the basic OHS abstractions. Moreover, the quality of the interface through which developer is able to make use of the OHS is heavily expressed through the *error* messages and details he is fed with. Finally, *satisfaction* is considered to be a factor that is influenced from all previous usability attributes plus the financial overhead for making use of a hypermedia system.

The proposed questions which originate from our experience of developing an OHS environment are addressed to the designers of the systems. The choices of the questions has not been based on a predetermined procedure. Some of the most important points that have influenced the questions' choise are presented below:

- Our experience in the developers' requirements about hypermedia services, which are easy to discover and easy to use/integrate into their applications [9].
- Observations of the task of making open hypermedia services available to the broad Web community [8].
- Existing usability questionnaires for quick usability evaluation [4].

Each one of the questions is related to one or more attributes introduced in the previous section. This relationship is presented in the following questions at table 1. Furthermore, a weight factor is introduced for each one of the questions that correlate to 5 usability attributes. For each attribute the sum of the weights is 100. Both questions posed and their weight factors remain open to further investigation. One of the primary goals of our agenda is to re-evaluate our questionnaire, taking into consideration the feedback provided.

[2]Analyzing usability from an end-user perspective is out of the scope of this paper.

Table 1: OHS Usability Evaluation Questionnaire

No	Question	A1	A2	A3	A4	A5
1	Does the architecture of your hypermedia system provide abstraction layers? Is it possible for a 3rd party application to use a service of your system?	20	8	0	0	0
2	Is there a native support for documentation?	30	0	15	11	5
3	Has the system introspection capabilities/ is there a self-describing support for the provided services?	40	4	0	0	0
4	Is there an API in order to make use of your system services? Are there any technical or functional specifications?	10	12	0	0	0
5	Does the hypermedia system allow the developer to augment/extend the provided semantics?	0	14	8	0	0
6	With respect to the domain behavior semantics, how well does the hypermedia system carry out the required tasks load?	0	12	0	0	10
7	Is the developer allowed to use various technologies on the client side?	0	15	0	0	0
8	Can your hypermedia system scale and to which parameter (users, number of hosts etc)?	0	15	0	0	0
9	Is the hypermedia system available under different platforms? Is there a provision for interoperation between the hypermedia system and other 3rd party systems?	0	15	0	0	0
10	Is the understanding of the system specifications and functionality achieved within a fair time frame by the developer?	0	0	77	0	0
11	Does your hypermedia system provide software tools for supporting the developer's work (e.g. IDE, Examples, Tutorials etc)?	0	0	0	89	10
12	Is there a financial overhead in acquiring and maintaining the hypermedia system?	0	0	0	0	45
13	Is the system able to provide its services in a real environment for commercial usage?	0	5	0	0	30
		100	100	100	100	100

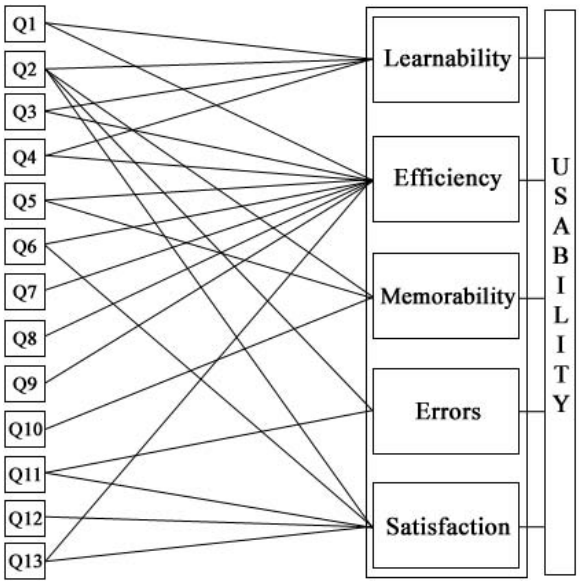

Figure 1: The structure of the proposed questionnaire.

All above questions were answered in free text. A critical part of the estimation process is the examination of the answers that leads to an arithmetic evaluation of the level of the answers participation to the questions attributes.

3. CASE STUDY

3.1 Systems

The questionnaire of table 1 was posed to the designers of a set of hypermedia systems. A brief presentation of these systems which includes the names of asked designers, follows.

- *Linky* (Millard, D.) [10]: Auld Linky is a structure server that is specifically designed to be used as an OHS with contextual support. It is a stand alone process that manages an XML "linkbase" of association structures expressed in the Fundamental Open Hypermedia Model (FOHM) [11] and provides pattern matching services via HTTP, modifying the structures served according to the declared context of the querying client. It structures associations between objects. Objects can be information resources (such as webpages, documents, multimedia annotations) or they can be physical objects (such as artifacts, locations), using a suitable naming scheme. Users querying Linky and after it provides a view of all the loaded link-bases, allowing unions and intersections of structure that cannot be handled with multiple linkbases. Linky's form of contextual Open Hypermedia can be used to implement many of the techniques common in Adaptive Hypermedia Systems. In fact Auld Linky can be seen as an Adaptive Engine for generic hypermedia structures, adapting both navigational links and content.

- *Babylon* (Karousos, N., Pandis, I.) [8]: The Babylon system targets to the provision of multiple categorization services using abstractions met in taxonomies. The goal is to provide developers with an easy and fast way to include item categorization into their applications. The main entity of the Babylon's data model is the object entity called 'item' in which several characteristics are assigned. Each item can be inserted in a single category. Each category is identified by a group of characteristics and may contain a set of items or other categories, formulating a tree-structure. The system supports the creation of category shortcuts in other categories of a tree, giving the notion of relevance. Finally, the meaning of association is also defined; a specific relationship between different tree categories that are being used for automatic tree-to-tree item transferring and sharing. Clients invoke Babylon service through HTTP GET and POST messages.

- *IUHM* (Nanard, M.) [12]: The Information Unit Hypermedia Model. (IUHM) is used for describing all relationships between arbitrary system entities, including services, data and metadata. It consists of a generic, computable hypertext structure with typed links, known as the Information Unit (IU), and is the minimal structural scheme to which all encapsulated entities comply. IUHM accepts heterogeneity and provides a common encapsulation mechanism, the IU, together with a common execution scheme. The reflexivity of the IUHM model provides a powerful means for composing services, which, together with the generic execution mechanism of the functional core, provides openness for the addition of an arbitrary new service and new usage policy.

- *WildDocs* (Atzenbeck, C.) [2]: WildDocs is a spatial-based knowledge management system. It is focuses to use knowledge management based on spatial structures. It differs from other spatial structure applications: WildDocs supports rich structure elements, such as rotation or bindings, and it offers realistic interaction methods, such as smooth zooming. The implementation follows on a component-based approach, based on a structural computing environment; this guarantees that it is easy extensible. However, WildDocs currently remains a monolithic system. It cannot operate as an autonomous service and its usage is possible only through its native interface by the end-users. Consequently, it is difficult for WildDocs to fully participate in the proposed evaluation.

- *Construct* (Wiil, U.) [17, 16]: The Construct development environment is targeted at the construction of different types of hypermedia services. The primary goal of the environment is to ease the construction of Component Based - Open Hypermedia Systems (CB-OHSs) by providing development tools that assist the system developers in the generation of the set of services that make up a hypermedia system. The environment has built-in solutions to many of the difficult issues involved in building CB-OHSs such as inter-process communication, naming and location, and storage management. Both predefined and generated services are encapsulated inside service components that can operate in a massively distributed environment such as the Internet. The development tools can generate a skeleton service from a graphical

UML diagram or from a high-level service specification written in IDL (see Figure 1). The system developer only needs to fill in the semantic parts of the generated skeleton (the service method bodies). Thus, the level of expertise needed to construct a state-of-the-art CB-OHS component with the Construct development environment is considerably lower than the level of expertise needed to build a similar component from scratch. The Construct development environment is coded in Java.

- *Callimachus* (Tzagarakis, M. et al) [15]: Callimachus is a CB-OHS that follows the structural computing principles. It targets to the provision of a variety of knowledge management services in the context multiple hypermedia domains (navigational, spatial, etc.). Its primary architectural elements are client applications, structure servers and infrastructure. Client applications can be either native or third-party applications, such the MS Office Suite and Emacs, or even web servers and web applications. Client applications request services from structure servers using a well defined protocol. The on-the-wire messages sent between clients and servers are encoded using XML All client-side aspects of the protocol come in the form of a library that implements an API (this library is an essential software module at client side). Different structure servers require different protocols to communicate with client applications, while this communication is stateless. The construction of the client-side API takes place during the development of the structure server. The Callimachus structure servers have the form of TCP/IP daemons that listen at a specific port for incoming requests.

The variety of the above systems architectures was a positive factor while evaluating the presented prototype. That was because the given answers covered a global scope and lead us to observations and conclusions that can affect design issues in different Open Hypermedia architectures.

3.2 The answers

All the received answers were rated in a percentage measurement and are presented in table 2.

From the given answers many conclusions about the hypermedia systems can be derived. Firstly, Linky - according to the answers of some questions (1 and 2) - was not designed to be an enterprise system and lacks the sort of security and scalability. In addition, it does not have a software oriented text editor bundled with it. According to questions 3 and 8, it does not have introspection capabilities and finally is not designed as a scalable solution so it would not scale globally. For these reasons the rate there, is 0. On the contrary, in the rest of questions with exception to 13, Linky has a good percentage.

Babylon contributes to the sum of questions but lucks in question 2 because there is not native support for documentation, in 3 as it is possible for 3rd party application to use Babylon as it is. Also in Babylon some tasks may cause to latency while responding to complicated requests so the rate in question 6 is 30 of 100. Finally, Babylon provides only some examples, and help documents so its contribution to question 11 is poor.

Table 2: Evaluation Results of Selected OHSs

	Linky	Babylon	IUHM	WildDocs	Construct	Callimachus
Q1	0	70	70	-	100	100
Q2	0	25	0	-	0	0
Q3	0	30	70	-	15	100
Q4	80	40	0	-	100	100
Q5	100	50	100	-	100	100
Q6	60	30	80	-	50	50
Q7	80	65	50	-	100	100
Q8	0	85	50	-	80	50
Q9	100	35	0	-	100	100
Q10	80	60	70	-	80	80
Q11	100	20	0	-	80	0
Q12	70	60	0	-	75	100
Q13	20	80	0	-	0	0

IUHM, on the one hand (questions 2, 4, 9, 11, 12 and 13), lacks because there is no documentation about internal structure. Also it supports only internally API in order to make use of system services and technical or functional specifications. IUHM does not provide software tools for supporting the developer's work and there is not available for end users a testbed for experimental research. Also it does not provide its services in a real environment for commercial usage. On the other hand, IUHM allows the developer to augment/extend the provided semantics.

WildDocs currently, is a monolithic system and in its first prototypic implementation was basically destined to be a "playground" to test certain implementation issues. Consequently, WildDocs could not answer to the given questions.

Construct seems to satisfy many requirements as it has three abstraction layers and an API for each service. Apart from that, according to the answers in questions 2, 3 and 13 there are some drawbacks due to inexistence of documentation in the system itself. Also it has not introspection capabilities even a self-describing support for the provided services and does not provide its services in a real environment for commercial usage.

Finally, Callimachus has no native support for documentation, does not provide software tools for supporting the developer's work and can provide its services in a real environment for commercial usage only partially. Contrarily, the answers for the rest of questions were reached a high level.

3.3 Evaluation procedure and results

The next step after the evaluation questionnaire and the rated answers is the establishment of the final results that relate Hypermedia Systems and the five usability attributes. In order to do that, we followed a specific procedure. Each of the questions is related to one or more attributes and affect them according to a weight factor. Totally, every attribute has a weight of 100 as it has been pointed out in Table

1. We also have the rates for each of the answers given by the hypermedia designers with maximum rate 100. In order to find the final rate of a usability attribute for a specific Hypermedia System the partial products of the rate of the answer and the weight of the attribute for all the questions will have to be added. Afterwards, the total sum will be divided by 100. A mathematical type that supports this operation for every Hypermedia System is the following:

$$Rate_{attribute} = \sum_{k=1}^{13} \frac{(Rate_{answer\ k} * Weight_{questionattribute\ k})}{100}$$

After collecting all the above results we use a scale of five categories (very low, low, average, good and very good) in order to give a comprehensible view of usability in OHSs. It is worth mentioning that the scale does not use equal values for every category and it is based on the empirical usage of evaluation practices. The scale uses as maximum rate 100. In this way "very low" is between 0 and 29, "low" between 30 and 49, "average" between 50 and 59, "good" between 60 and 84 and "very good" between 85 and 100. Finally we produce table 3 that presents the usability evaluation results of the selected OHSs.

Table 3 can result in several observations. A first impression from the usability pre-estimation is that learnability, errors and to a considerable amount satisfaction don't meet developers needs in a satisfactory level. As far as learnability is concerned, developers have difficulty in understanding how the systems work. In particular little have been done in order to provide native support for documentation (Q2). The above fact constitutes a serious obstacle to the making of hypermedia systems open to developers. Furthermore, the majority of Open Hypermedia Systems, that have been selected, has not any introspection capabilities or self-describing support for the provided services (Q3). Concluding, learnability needs more accurate attention in the future.

Although developers have difficulty in understanding how the systems work the memorability attribute remains in a good level for all the selected systems. This means that its easy for someone to reuse a system after a period of time.

Moreover, as shown in table 3, Open Hypermedia Systems seem to be able to satisfy the needs of hypermedia developers (efficiency), but it is considered that the errors is an attribute that is unregarded from most of the systems as they don't provide software tools for supporting the developers work (Q11). As a conclusion, errors should be studied and invested further in order to provide developers with error handling together with error explanation.

In addition Satisfaction is on a modest level and only some OHSs provide its services in a real environment for commercial usage. Finally it would be expected to rise, if for example, some of the above systems were initially designed for external use.

4. IMPROVING THE USABILITY FACTOR DURING THE DESIGN AND THE DEVELOPMENT OF OPEN HYPERMEDIA SYSTEMS.

The systems that have been used in the above case study are in fact prototypes or just light versions and do not predicate commercial use. The results, in a way, cater for many of the usability attributes but finally the concern for usability is not satisfactory. At this point we have to ask ourselves, if the Open Hypermedia Community must take into account the usability requirements during the prototype design.

There is no doubt that the cost of fulfill of these requirements into the design and development phase of hypermedia systems is important. In addition, the main goal of researchers is the construction of both models and prototypes of systems and not commercial applications. Thus, the value of such task sometimes is less important for a number of designers.

On the other hand, the objective of these systems and their services being used by end-users, developers and external applications calls for supporting usability criteria. This means that even in the prototyping level some of the basic criteria must be satisfied. Moreover, as the requirements affect design issues, it is preferable that the criteria should be taken into consideration during the development. This is because in case one of the usability attributes is not satisfied the cost of the resulting alteration will rise and the redesign of the systems will probably be inevitable.

In this context, the evaluation results to the need for the OHSs to be aware of the usability factor towards the improvement of the provision of hypermedia functionality to 3^{rd} party developers and applications. More specifically, there are three critical levels in which some important guidelines for supporting usability are presented:

- *Providing hypermedia functionality through easy integrated services*: Moving OHSs from monolithic architectural modes to Service Oriented Architecture (SOA) or Component Based - Open Hypermedia System (CB-OHS) can facilitate to the decrement of integration cost. Furthermore, the appliance of standardized SOA, like Web Services that support self description capabilities and platform independency will contribute to easiness of the hypermedia functionality discovery and usage.
  SOA in hypermedia systems will also enable the service composition and the creation of customized services, helping developers to reuse already tested services or to develop their own custom services based on a particular use case scenario. Finally this movement can be an important try to address scalability requirements due to the usage of standardized service oriented methodologies that are already aware of the particular issue.
- *Development towards usable APIs*: Hypermedia systems functionality should be provided through clear APIs that can fulfill the usage requirements. There are several architectural guidelines that address the need

Table 3: OHS Usability Evaluation Questionnaire

	Linky	*Babylon*	*IUHM*	*Construct*	*Callimachus*
Learnability	very low	low	low	low	good
Efficiency	average	average	low	good	good
Memorability	good	average	good	good	good
Errors	very good	very low	very low	good	very low
Satisfaction	good	average	very low	good	low

of supporting usability while designing such systems. For example, one of them proposes the decoupling between the structural layer and the interface layer of a hypermedia service in order to make a flexible service. In that case, the designer will have the ability to make alterations in the API layer according to the user's requirement with no influence to the structural layer. A more innovated approach is the appliance of usability-supporting architectural patterns in software systems in which whole systems are redesigned in a basis of design patterns as it is noticed in [3, 14].
Apart from API creation, the Usability Engineering Life Cycle could also be used in such systems by augmenting the OHS creation process with prototype implementation and user testing which will be able to redesign the system and recreate the prototype through callbacks if needed. Testing workgroups with no awareness of the whole system together with workgroups of hypermedia aware users will make valuable results in the systems evaluation.

- *Providing helpful tools and documents for enabling service usage*: The lack of well documented hypermedia systems is an important aspect that has to be addressed. Information of how a system works, what is the API of the system and how it can be invoked and interoperated by custom applications should be provided in an accessible way. Developers who try to integrate their applications with OHSs should have the ability to feedback the system designers with comments and valuable notifications in order to contribute to newest versions of more usable hypermedia services. In addition, examples of using a hypermedia system can upgrade the provided usability level due to the decrement of integration time. Towards that, automatic creation of the client side communication API will also have important results.

Despite the fact that making hypermedia systems as commercial products is not one of the primary goals of the hypermedia research community, the aforementioned issues can guide to stabilization of OHSs and enhance developers confidence in the usage of hypermedia systems.

5. FUTURE WORK

The usability pre-estimation questionnaire was designed based on the empirical definition of both the questions and the weights. This means that a continuous feedback from designers of OHSs will contribute to a constant redefinition of the weights which will yield their stabilization. Apart from the weights, the number and the content of questions remain open as well. In order to enable the questionnaire stabilization, the provision of the questionnaire as a web-service to the designers will be our first step. This service will be provided to designers through a web site or as a pure SOAP web service and will support actions such as the query answering, the commentary of the questionnaire and the proposition of new usability pre-estimation ideas.

On the other hand, the only secure way of evaluating system usability are both the end-users' testing and evaluation of the system. Thus, the next step will be a try for a real time evaluation task of both Babylon and Calimachus systems. Results of the evaluation will be cross-checked against the corresponding pre-estimation results that have been pointed out in Table 3 and will be used in order to evaluate the questionnaire weights and questions.

Furthermore, a continuous questionnaire refinement will increase the maturity level of the questionnaire and will show the way to the definition of a set of predefined answers that could assist the completion of the questionnaire. This will transform the evaluation of the answers to an automatic process.

The final target is to try to redesign the Babylon server according to the design patterns for usability support. In this task we will also focus on the creation of both the appropriate documentation and tools that can help the Babylon usage of hypermedia unaware developers.

6. CONCLUSIONS

As the demand for the increase of the OHS use rises, the need for providing high quality services to developers grows. Usability has a great impact on the quality of services. System designers are encouraged to introduce the usability factor in every aspect of hypermedia engineering. Starting from designing prototype editions up to the level of building ready to use hypermedia systems and services, the aspect of usability should always be taken into consideration. This task could produce new system specifications in each phase of the development procedure.

In order for OHS designers to pre-estimate the usability factor, a usability pre-estimation questionnaire is proposed. By giving them the ability to follow a self-evaluation procedure in the early stages of system development we help the incorporation of usability attributes into it with a low cost. Consequently, usable hypermedia systems will raise the quality of provided services to a high level. As a result, the acceptance of such systems by both knowledge workers and developers will be enhanced.

7. ACKNOWLEDGMENTS

The authors would like to thank the designers for answering the questionnaire.

8. REFERENCES

[1] K. M. Anderson. Software engineering requirements for structural computing. In *Proceedings of the 1^{rst} Workshop on Structural Computing*, pages 22–26, University Esbjerg Denmark, 1999.

[2] C. Atzenbeck and P. J. Nürnberg. Wilddocs - emerging metainformation support. In *Proceedings of the 5th I-KNOW Conference*, Graz, Austria, 2005.

[3] L. Bass, B. E. John, N. Juristo, and M.-I. Sanchez-Segura. Usability-supporting architectural patterns. In *ICSE '04: Proceedings of the 26th International Conference on Software Engineering*, pages 716–717, Washington, DC, USA, 2004. IEEE Computer Society.

[4] J. Brooke. *SUS: A 'quick and dirty' usability scale*. UK: Taylor & Francis, London, 1996.

[5] Engelbart, D. Keynote speech, 4th Int'l Workshop on Open Hypermedia Systems, 1998. http://www.bootstrap.org.

[6] F. Garzotto, L. Mainetti, and P. Paolini. *Hypermedia Application Design: A Structured Approach*. Springer-Verlag New York, Inc., 1996.

[7] ISO. *EN ISO 9241-11. Ergonomic requirements for office work with visual display terminals (VDT's). Part 11*. 1997.

[8] N. Karousos, I. Pandis, S. Reich, and M. Tzagarakis. Offering open hypermedia services to the www: a step-by-step approach for developers. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 482–489, New York, NY, USA, 2003. ACM Press.

[9] N. Karousos, M. Tzagarakis, and N. Koumbarou. Selecting services for web applications: The open hypermedia case. In *International Workshop on Web Engineering in conjunction with ACM Hypertext 2004*, Santa Cruz, 2004.

[10] D. T. Michaelides, D. E. Millard, M. J. Weal, and D. D. Roure. Auld leaky: A contextual open hypermedia link server. In *Revised Papers from the International Workshops OHS-7, SC-3, and AH-3 on Hypermedia: Openness, Structural Awareness, and Adaptivity*, pages 59–70, London, UK, 2002. Springer-Verlag.

[11] D. E. Millard, L. Moreau, H. C. Davis, and S. Reich. FOHM: a fundamental open hypertext model for investigating interoperability between hypertext domains. In *UK Conference on Hypertext*, pages 93–102, 2000.

[12] M. Nanard, J. Nanard, and P. King. Iuhm: a hypermedia-based model for integrating open services, data and metadata. In *HYPERTEXT '03: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pages 128–137, New York, NY, USA, 2003. ACM Press.

[13] J. Nielsen. *Usability Engineering*. Academic Press, USA, 1993.

[14] K. Sousa, E. Furtado, and H. Mendonça. Upi: a software development process aiming at usability, productivity and integration. In *CLIHC '05: Proceedings of the 2005 Latin American conference on Human-computer interaction*, pages 76–87, New York, NY, USA, 2005. ACM Press.

[15] M. Tzagarakis, D. Avramidis, M. Kyriakopoulou, m. c. schraefel, M. Vaitis, and D. Christodoulakis. Structuring primitives in the callimachus component-based open hypermedia system. *J. Netw. Comput. Appl.*, 26(1):139–162, 2003.

[16] U. K. Wiil. Using the construct development environment to generate a file-based hypermedia storage service. In *OHS-6/SC-2*, pages 147–159, 2000.

[17] U. K. Wiil, P. J. Nürnberg, D. L. Hicks, and S. Reich. A development environment for building component-based open hypermedia systems. In *HYPERTEXT '00: Proceedings of the eleventh ACM on Hypertext and hypermedia*, pages 266–267, New York, NY, USA, 2000. ACM Press.