Preprint: final version of paper available from

BLANDFORD, A., KEITH, S., CONNELL, I. & EDWARDS, H. (2004) Analytical usability evaluation for Digital Libraries: a case study. In *Proc. ACM/IEEE Joint Conference on Digital Libraries*. 27-36.

Analytical Usability Evaluation for Digital Libraries: a Case Study

Ann Blandford
University College London
Remax House, 31-32 Alfred Place
London WC1E 7DP
+44 02 7679 5288
a.blandford@ucl.ac.uk

Suzette Keith
Middlesex University
Trent Park Campus, Bramley Road
London N14 4YZ
+44 20 8411 5098
S.Keith@mdx.ac.uk

Iain Connell & Helen Edwards
University College London
Remax House, 31-32 Alfred Place
London WC1E 7DP

ABSTRACT
There are two main kinds of approach to considering usability of any system: empirical and analytical. Empirical techniques involve testing systems with users, whereas analytical techniques involve usability personnel assessing systems using established theories and methods. We report here on a set of studies in which four different techniques were applied to various digital libraries, focusing on the strengths, limitations and scope of each approach. Two of the techniques, Heuristic Evaluation and Cognitive Walkthrough, were applied in text-book fashion, because there was no obvious way to contextualize them to the Digital Libraries (DL) domain. For the third, Claims Analysis, it was possible to develop a set of re-usable scenarios and personas that relate the approach specifically to DL development. The fourth technique, CASSM, relates explicitly to the DL domain by combining empirical data with an analytical approach. We have found that Heuristic Evaluation and Cognitive Walkthrough only address superficial aspects of interface design (but are good for that), whereas Claims Analysis and CASSM can help identify deeper conceptual difficulties (but demand greater skill of the analyst). However, none fit seamlessly with existing digital library development practices, highlighting an important area for further work to support improved usability.

Categories and Subject Descriptors
D.2.2 [**Design Tools and Techniques**]: User Interfaces; H.1.2 [**User / Machine Systems**]: Human factors; H.3.7 [**Digital Libraries**]: User issues; H.5.2 [**User Interfaces**]: Evaluation / Methodology, Theory and methods, User-centered design

General Terms
Design, Human Factors.

Keywords
Digital Libraries, usability evaluation, Heuristic Evaluation, Cognitive Walkthrough, CASSM, Claims Analysis, scenarios, personas, Scenario-Based Design.

1. INTRODUCTION
Digital libraries hold great promise as structured repositories of quality-checked information that can be manipulated and accessed in powerful ways. However, they also pose great challenges from many different directions: technical, organizational, legal, and others. In particular, as digital libraries become more widely available it is becoming clear that they also pose usability difficulties [2][8][9].

There is no simple analysis of why digital libraries can be so difficult to use. Blandford and Buchanan [7] highlight a number of issues, from the individual to the social, and from the superficial (e.g. conflicts in the terminology used by different stakeholders) to deeper issues concerning fundamental difficulties users sometimes have in even knowing what they are looking for.

On the other side, there is no unique view of the development process for digital libraries. Bates [4] discusses the complexities of DL development in terms of a "cascade" of interactions. We discuss the fragmentation of development further in section 5 (below), as part of our account of applying Claims Analysis with developers.

One key challenge is to understand users' difficulties in working with information and particularly with digital libraries (DLs), and to equip developers with ways of thinking about users and their needs that help guide DL development and evaluation. In this paper, we present and discuss various techniques we have investigated for addressing this need.

2. BACKGROUND

Two important lines of research contribute to our understanding of how people work with digital libraries. The first is research on the activity of finding information; the second is empirical studies specifically studying digital library use and usability.

Finding information generally takes places within some broader information task; for example, Kuhlthau and Tama [20] relate users' information needs to the broader tasks of lawyers, while Attfield, Blandford and Dowell [1] relate information seeking to the broader context of writing, relating information needs to the creative process of designing a piece of text. Understanding at this level tends to be domain-specific. A more general overview of users' activities as they relate to the design of digital libraries can be gained by considering the next level down: that of information seeking processes.

Several models of the information seeking process exist (e.g. [3][15][17][21][28]). These differ largely in focus and terminology rather than substance, so we briefly describe Kulhthau's [21] Information Search Process (ISP) model as an example; we refer back to this model at various points in our discussion of evaluation techniques for DLs. Kuhlthau's ISP model identifies six stages of the information search process through which an information seeker moves from uncertainty to a constructed understanding. The information search process begins with vague thoughts and associated feelings of uncertainty, followed by identification of possible information sources. The feelings of uncertainty dissipate as the information seeker weighs prospective topics against criteria of personal interest. As the person explores general information on the chosen topic, feelings of uncertainty and confusion increase, and it is at this time that an inability to express precisely what information is required results in awkward communication between the user and the information system. The most critical part of the process is the point where the information seeker forms a focus for their task; this acts as a turning point. Following formulation thoughts become clearer, uncertainty gives way to confidence, and clarity and confidence increase as the user gathers information. Finally, a sense of relief is experienced as the search is completed. This model is necessarily simplified by abstracting away from detail; it should also be noted that search does not generally proceed in a neat, linear fashion from initiation to conclusion, but involves iteration and back-tracking, and sometimes even abandonment.

This model is presented in a way that is independent of the places (such as traditional or digital libraries, or the Web) in which search takes place, but provides a good overview of the key stages of information seeking. Focusing further, it is necessary to consider how users interact with particular information systems, such as digital libraries. In this case, details such as how users formulate queries and react to search results become important. Stelmaszewska and Blandford [26] found that some of the greatest user difficulties when working with DLs arise in formulating effective queries and understanding search results. As noted above, early on in a new search, users may not even know quite what it is they are looking for: problem formulation emerges through the dialogue between a partially specified topic (for example, an essay title) and a set of information sources; at this stage, the user may have difficulty assessing the relevance of documents to their information task. In principle, a browse-based interaction can be more effective than search at this stage, depending on the thematic organization of the information sources. As the task becomes better understood, the bounds of the information problem become clearer, and the issue becomes more one of understanding how to formulate a query for this particular search interface (e.g. using Boolean logic or descriptor fields). When browse or search results are returned, the user has to be able to assess the relevance of each document rapidly and reliably. Each of these steps is demanding, and can pose such difficulties that users give up or leave with unsatisfactory results [8].

The discussion so far has not considered particular DLs. In practice, most work on usability of DLs has involved classical empirical studies of single libraries or features within libraries – generally conducted or commissioned by the systems' developers, to inform future developments (e.g. [19][23]). While this work has advanced the general understanding of usability issues for DLs, it seemed to us that empirical work could usefully be complemented by analytical techniques:

- While empirical studies tell you what people do, they often fail to highlight what it is about the design of the system that is provoking particular user behaviors. Well chosen analytical techniques can yield insights not available through empirical data. As illustrated in the analyses below, design includes everything from interface layout and the choice of labels to information structure and interaction possibilities.
- Some DL development teams have easy access to their target user population, so that empirical trials are relatively easy to organize; others do not, so that every empirical study is costly, in terms of time and effort. For example, in section 4, we discuss the Athens authentication feature, which we believe to only be available in the UK and a few other locations; although the example makes use of a widely available DL, a user's experience is heavily influenced by a feature that is local to the UK. Thus, testing a globally available DL demands working with users in different continents to identify both cultural differences in the ways users perceive DLs (e.g [14]) and also localization differences.
- Analytical evaluation makes the analyst think deeply about the design and about users, which can yield insights and long-term learning that inform future design decisions.

We are only aware of one previous study that has applied and compared analytical techniques in library systems evaluation; that study, by Warren [29], informally compared the application of Heuristic Evaluation, design patterns and 'Computers as Theatre' concepts to the design of an OPAC system; however, the focus of that work was on understanding the usability issues pertaining to the OPAC system, rather than understanding the scope of the evaluation techniques being applied.

In the following sections, we present and discuss a selection of text-book usability evaluation techniques, focusing on how we have adapted and used them in the context of digital libraries, and what we have found out about them. The four techniques (Heuristic Evaluation, Cognitive Walkthrough, Claims Analysis and Concept-based Analysis of Surface and Structural Misfits (CASSM)) are discussed in approximate order of increasing complexity. The first two have only been tested in-house by us; the remaining two have been tested with collaborators (developers of digital libraries).

Since the focus of this report is on the evaluation methods, rather than any particular DL, we have used different examples to illustrate the methods. Most of the methods have been applied to the same proprietary DL, being developed by one of our collaborators, but not to a widely accessible library; we have chosen to use widely accessible examples here.

3. HEURISTIC EVALUATION

Heuristic Evaluation [22] is probably the most widely used usability evaluation technique, because it is perceived to yield reasonable benefits for low cost.

3.1 Overview of the technique

Heuristic Evaluation (HE) is a checklist-based approach to assessing the usability of an interactive system. In the original version of this technique, the analyst (or team of analysts) works through every page or screen of a system, asking ten questions about that system. For example, one of the prompts is "Match between system and the real world". In other words, the system should use words, phrases and concepts familiar to the user, rather than system-oriented terms. It should also follow real-world conventions, making information appear in a natural and logical order.

Nielsen [22] suggests that between three and five analysts should assess the system; their notes can then be compared to generate a shared perspective on usability issues for the system. Cockton *et al* [11] have argued that a smaller number of analysts can perform better, provided that they have a structured way to report problems that encourages reflection on their assessments; in particular, fewer analysts generate fewer false positives – i.e. predicted problems that are not found to be actual problems in empirical studies. They also recommend that, rather than trying to consider every possible system state (e.g. web page), analysis is more effective if analysts consider a set of user tasks and all the pages a user will experience in performing those tasks in sequence.

3.2 Example of application to a DL

For the purposes of illustration, we present just one heuristic as applied to a single page of the SwetsWise DL (www.swetswise.com) (Figure 1).

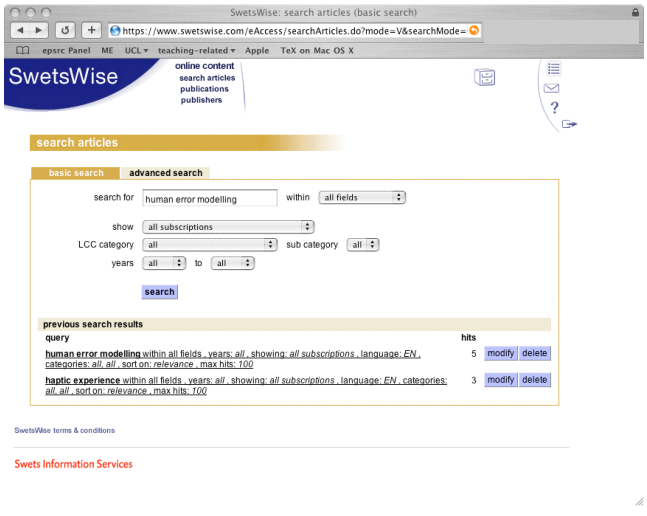

Figure 1: Search results in SwetsWise

Consider the heuristic "Match between system and the real world". The user is asked to select search categories from various pull-down menus: "all fields", "show", "LCC category", "sub category", "years". Most of these terms can be understood either immediately or by the user who views the options available under each category. However, the term "LCC category" is likely to be mysterious to most users, particularly those outside the USA (LCC is short for "Library of Congress Classification"). Considering the icons top-right: the filing cabinet icon draws nicely on the real-world analogy of a place to store documents, whereas the topmost icon (dots and lines), which takes the user to a news page, is unlikely to be interpreted correctly by many users on first viewing. Even the terms "modify" and "delete" may be slightly confusing as the former refers to the query formulation while the latter refers to the results set, but their juxtaposition implies to the user that they will act on the same object.

3.3 Findings and discussion

Like others who have used Heuristic Evaluation [16], we found that there can be huge differences between the findings of different analysts. The heuristics provide a broad checklist of issues to consider, but most DLs offer so many possible routes through, and operate on such large document repositories, that it is almost impossible to be sure that one has assessed every page of the DL or every possible transition through the system. Thus, Nielsen's recommended strategy of considering every page proved infeasible, while the alternative strategy of considering particular user tasks forced us to work with well-defined tasks – a difficulty we discuss further in relation to Cognitive Walkthrough (section 4). Also, as others have noted [29] and our own empirical studies [8] have shown, most often it is not surface features of individual pages that make libraries difficult to use: it is difficulties in formulating effective queries, assessing results, noticing and responding to transitions between libraries, managing multiple windows, etc. – phenomena that Heuristic Evaluation does not address.

When using Heuristic Evaluation, we found there was also a great tendency to focus on local issues – concerning one particular page or the transitions between just a few pages – so that we lost the big picture. It could help the analyst identify surface difficulties – poor wording, poor grouping of information, etc. – but not deeper conceptual difficulties with a library. This finding is consistent with that of Warren [29], who notes that HE focuses attention on micro features of an interface rather than the global picture.

Finally, the list of heuristics cannot be applied effectively without expertise – not just in human factors but particularly in understanding users of digital libraries, their typical tasks and the kinds of terminologies they use. For example, considering the heuristic "Match between system and the real world", as used in the example above, how can the analyst know what terms user thinks in terms of? Particularly if the analyst is knowledgeable about libraries and librarians' terminology, it is difficult to remember how non-specialist users use terminology. This issue is discussed more below in terms of CASSM.

In summary, Heuristic Evaluation proved useful in identifying surface usability difficulties, but demands expertise, including an understanding of the users and uses of DLs, to use well.

4. COGNITIVE WALKTHROUGH

Looking for an approach that might give insights into a narrower range of usability difficulties, but that was better structured, we investigated Cognitive Walkthrough (CW) [30], an approach that takes an explicitly user-centred perspective on design. Like HE, CW has been taken up, at least to a limited extent, in commercial software development [25].

4.1 Overview of the technique

Wharton *et al* [30] recommend that CW be conducted by a team of analysts. The team needs to agree:

1. Who will the users be?

   Assumptions about prior experience and knowledge should be articulated

2. What tasks are to be analysed?

   The team should agree a set of user tasks to work on, carefully selected to be representative of the broad range of task the system supports, and to test features of the system as thoroughly as possible.

3. What is the correct action sequence for each task?

4. How is the interface defined?

For the next stage of analysis, team members work separately, working through the tasks as agreed and, for each step of each task, considering the following four questions:

1. Will the user try to achieve the right effect?

   Given what the user is trying to achieve overall, will they identify the correct way to achieve it with this system?

2. Will the user notice the correct action is available?

   Is the action visible at the interface, or somehow discoverable?

3. Will the user associate the action with the effect?

   Is it obvious the action addresses the goal?

4. If the correct action is performed, with the user see that progress is being made towards the goal?

   Is the feedback helpful?

Once each team member has worked systematically through the tasks, the team reconvene to compare notes and summarize findings.

4.2 Example of application to a DL

We present a brief illustrative example of the application of CW to a well-defined search in the Ingenta DL (www.ingenta.com). See Figure 2.

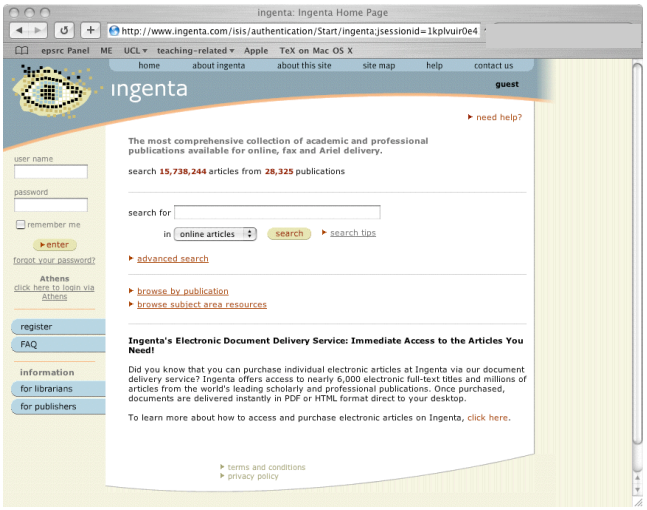

Figure 2: home page for Ingenta

We first have to define user assumptions. In this case, reasonable assumptions are that the users are academics and researchers who make use of journals as part of their work, that they are competent web users, but do not necessarily have sophisticated searching skills. They will generally have good knowledge of their subject, but not necessarily be familiar with what is in this particular library. For this example, we assume that the users are based in the UK, and have access to the Athens user authentication system (www.athensams.net).

Normally, a set of tasks would be defined; here, we consider just one: to find journal articles on usability and haptic interactions. However, in order to use the DL we need to gain entry by logging on via an Athens account which handles the access permissions. Thus the task sequence, starting from the Ingenta home page is to:

- log in to the Athens account,
- enter a search expression such as "haptic interaction usability" into the search box and press "search",
- view the results,
- select full text for any interesting article, then
- select "pdf" to download the article.

Since Ingenta is a fully functioning system, the interface is defined by the series of screens the user sees (such as that shown in Figure 2). Note that if CW was being applied to an early design, a storyboard of screen sketches could be used instead of real screenshots.

For each step in the interaction, we need to answer the four questions. In this paper, for illustration purposes, we only work through the very first step of the interaction – logging in via Athens. If the user fails to complete this step successfully, they will be unable to download full-text documents.

Figure 3: The next screen seen by the user (Athens)

Will the user try to achieve the right effect? In this case, will the user know that the first thing they should do is to log in? If the user is a regular DL user, they may be aware that before they can access documents they need to log in; since every user has to register explicitly for Athens, they will be aware that they have an Athens user name and password, and will therefore expect to use this. On the other hand, a user who does not use libraries regularly may not anticipate needing to log in, and therefore will not try to "achieve the right effect" at this point in the interaction. More critically, the new user may not even have an Athens account, and may therefore need to visit the administering library to set one up.

Will the user notice the correct action is available? The "login via Athens" option is clearly displayed on the left-hand side of the screen. Due to the color scheme adopted on this page, the user's attention is more directly drawn towards the search box in the centre, so the user may not immediately notice the Athens option, but it is, nevertheless, clearly displayed.

Will the user associate the action with the effect? The action is clearly labeled "login via Athens", so if the user has the goal of logging in via an Athens account, it should be very obvious to them that pressing this link is the correct way to do it.

If the correct action is performed, with the user see that progress is being made towards the goal? To assess this, we need to see the next screen the user would see in this situation. This is shown in Figure 3. This screen is clearly labeled "Athens Authentication Point", with clearly labeled data entry fields for username and password, so yes: the user will see that progress is being made towards the goal.

A similar procedure is followed for each step of the interaction.

4.3 Findings and discussion

CW helps to identify issues – such as whether the user will know they need to log in, whether appropriate terminology is used, whether options are visible and what the quality of feedback is – that relate to local features of the interaction. Compared to HE, it provides a clear structure for how to go about the analysis once user profiles and tasks have been defined. While it is more limited in scope than HE (for example, not considering how to recover from errors), the more explicit structure makes findings slightly more reproducible across analysts.

However, the rigid structure of CW limits its scope. For example, CW was not well suited for considering error sequences. To illustrate this, consider the task discussed above: it is possible for the user to perform the search and request an article before logging in – something users may well do, as there is little point in logging in if it turns out that there are no interesting articles to download. This task sequence needs to be analysed as a separate task. In this case, when requesting download of the article, the user will see a screen as shown in Figure 4. The only options visible at this point are to pay for the article, to sign in via Ingenta Select, or to enter user name and password (shown at bottom left of screen). The user is likely to use one or other of the visible sign-in options, but neither of these will work – the user is simply told their details are not recognized by the system. If the user scrolls down the page, they will find the Athens login option (currently hidden off-screen to the bottom left). CW does not explicitly support even this level of error analysis.

Figure 4: Ingenta: user not logged in.

As well as these limitations, the greatest challenge with CW turned out to be in specifying tasks. Most real users' interactions with DLs are opportunistic, and tightly coupled: users do not plan the details of their interactions ahead of time; it is not realistically possible to specify any but the simplest search tasks (such as that illustrated above) in such a fashion. In addition, it was difficult to specify user knowledge at the level of detail needed to anticipate how a particular user would react at any given point in the interaction.

In summary, like HE, CW addresses surface features of usability well, but does not identify deeper issues such as how users formulate good queries and evaluate results – more fundamentally, how users interact with the information in the library. To try to get at these deeper issues, we turned to Claims Analysis.

5. CLAIMS ANALYSIS

Claims Analysis (CA) [10][27] is a form of "psychological design rationale" – that is, a semi-structured approach to considering design from a user perspective. Claims are statements about the positive and negative effects of a design on the user within a particular context of use (a 'scenario'). Claims Analysis is less structured than Cognitive Walkthrough. Compared to Heuristic Evaluation, it is less

structured in terms of the 10 principles, but more structured in the way the context is specified.

5.1 Overview of the technique

Claims Analysis is ideally applied during design. The process of generating claims starts with generating user scenarios. These are similar to the tasks of Cognitive Walkthrough, but may be at a higher level of abstraction and contain more contextual information about a user, as illustrated in the example in section 5.2 below.

For each scenario, the analyst (or design team), work systematically through the main features of the design, invoking either relevant theory (e.g. on information seeking) or empirical results (e.g. of users working with an earlier version of the system) to generate claims about what effects the feature will have on the user. These will include both positive and negative claims, reflecting the upsides and downsides of the proposed design.

Carroll and Rosson [10] propose nineteen questions that can be asked about each feature to guide the generation of claims; Sutcliffe and Carroll [27] streamline this, proposing the Cognitive Walkthrough questions (as outlined above) as guides. We simplified the questions slightly further, structuring claims generation in terms of user goals, user actions and system feedback.

Once negative claims have been identified about a particular feature, ways to change the design that reduce the negatives while retaining the positives can be considered.

5.2 Example of application to a DL

For illustrative purposes, we present a sample scenario and claims base around the user task described in section 4.2. As noted above, scenarios are typically more abstract than CW task descriptions, so a corresponding scenario might be:

> "A researcher is conducting a literature search in preparation for starting a new project on usability issues for haptic interfaces. The organization she works for subscribes to the Ingenta digital library, and user authentication is managed via Athens. The user has only worked with digital libraries a few times before, and does not have sophisticated information seeking skills. She decides to search the Ingenta library to find articles relevant to her new project. Initially, her searching is exploratory (as described in the information seeking models outlined in section 2), but it gradually becomes more focused as she gains familiarity with the contents and structure of the library."

Assuming the user has arrived at the home page of Ingenta (Figure 2), we see that this page has many features: at the very top, there is a link bar that is accessible from every Ingenta page; there is advertising information about how many articles and publications are available through Ingenta; the search facility is prominent on the page; logging in and Athens authentication are available on the left; etc. For each of these features, we can consider user goals, actions and feedback.

Consider the search box. It is very clear to the user that this is a place to type in search terms (positive claim about the user's goal); however, the user may have difficulty formulating a good query (negative claim about the user's goal). Once the user has selected the box, typing is easy (positive claim about the user's action); however, it is not obvious to the user that they have to explicitly select the box before typing: if the user types without selecting the box the text is lost (negative claim about the user's action). As the user types, the words appear so that the user can check their query formulation (positive claim about feedback). And so on. There is not necessarily a single positive and negative claim about each feature; there may be multiple claims or none.

5.3 Findings and discussion

We have developed and applied CA in collaboration with two DL development teams, as reported more fully elsewhere [5][18]. Although we have found it more difficult to learn than HE or CW, we have also found that it supports the analyst in thinking about usability issues for DLs more deeply. Rather than focusing particularly on well defined tasks and details such as interface layout and qualities of feedback, we found that CA provokes thinking about why things are the way they are, and how they could be different. Like HE, different analysts tend to identify different points in the claims, but the structure of questions and the process of considering features guides the analysis at a level that is appropriate to the current stage of design taking account of high level user activities as well as the detailed interaction examined in CW. Conducting CA without working with developers raises many of the same difficulties as working with HE: that it is difficult to ensure coverage of all library features; in addition, claims are intended to reflect the developers' intentions in designing particular features – intentions that can be impossible to fathom without access to developers.

One of the real strengths of CA, in our experience, is the use of scenarios. We found it helpful to augment these with personas [13] – descriptions of different individuals that might use the library. While these individuals are fictitious, they can be based on either real individuals or generalizations drawn from (for example) the information seeking literature. These personas helped to overcome one difficulty for developers: imagining what real users are likely to do and what their prior experiences might be. There is a great tendency to think of users as falling into one of two categories: either they are highly sophisticated individuals who know as much about the library as the developers do and are therefore capable of finding materials within it with little difficulty, or they are exploratory users who are investigating what is possible with the library and are prepared to try anything to see what happens. In practice, while both these categories of user exist, the vast majority of users lie between these extremes; in particular, most users are not information seeking experts, they may not be particularly familiar with the features offered by this particular library, but they will have a particular information need, even if, as discussed above (section 2), that need is not yet well formulated.

A substantive gulf of understanding separates human factors specialists and the DL developers they work with. In addition, there can be a fragmentation of responsibility for the user interaction design: those who develop underlying technical infrastructure, those who extend core systems with novel but useful features and those who develop collections for end users to access are often working independently of each other, but all have some responsibility for the interface that the users ultimately work with. *Systems* developers focus attention on engineering high quality systems; if they do not, given the

unavoidable complexity of DLs, systems will be riddled with bugs and inconsistencies that will make them impossible to use. However, technical challenges can be so demanding that it is really difficult for developers to simultaneously be thinking about the user's perspective. In parallel, *collection* developers focus their attention on the management of the collection, including its structure, organization and access rights. Although usually closer to the end users than the system developers, it is difficult for them to distinguish critical differences in user skills and understanding of both system and content. Meanwhile, the *human factors* specialists naturally think in terms of the user's experience, and they find it difficult to fully grasp the technical constraints that determine what is and is not possible in terms of design of the system, and organizational imperatives of the collection. Scenarios and personas, which were needed before it was possible to generate claims, proved invaluable in bridging this gulf.

Claims allowed us to draw on theories of information seeking behavior, as outlined in section 2, as well as empirical studies of users working with DLs, to contextualize the usability evaluation to the design of DLs in particular, in a way that HE and CW did not.

6. CASSM

Taking a different perspective on bridging the gulf between human factors and software engineering, we applied CASSM (Concept-based Analysis of Surface and Structural Misfits) to two DLs – one in collaboration with the developers, the other stand-alone. CASSM is a locally developed evaluation technique [12] that is intended to complement existing methods, most of which consider design in terms of procedures (tasks, scenarios, etc.). In the first study, we focused on a music DL (within the New Zealand DL, www.nzdl.org), eliciting information from technically aware musicians and comparing that to the representation within the DL; the findings are summarized by Connell *et al*[12]. In the second, people were interviewed about their conceptualization of a library and the findings were compared against an analysis of the ACM DL (www.acm.org/dl/) (Figure 5). As Duncker [14] notes, digital libraries are not the same as traditional ones – the term 'library' is more a metaphor than an accurate description. However, the study allowed us to probe this metaphor.

6.1 Overview of the technique

CASSM considers design in terms of concepts: the concepts the user is working with, those implemented within the system, and those represented at the interface. The analysis focuses on the quality of fit between the user and system concepts. Concepts are considered in terms of *entities* and *attributes*; the analyst determines how easy it is for a user to create or delete an entity, or to set or change the value of an attribute. In addition, the analyst may identify *relationships* between concepts; in the analyses reported here, this step was omitted.

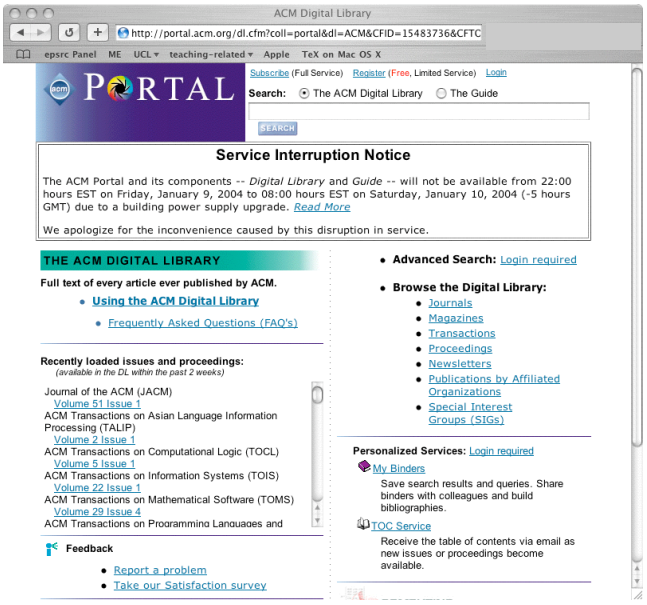

Figure 5: home page of the ACM DL

User concepts are identified by eliciting knowledge from real users; this may take the form of interviews, observational study with think-aloud, or any other approach in which users talk in their own words about what they are doing and how they think about their interactions with the system (for whatever tasks they perform with it).

System and interface concepts are identified through system documentation, through inspection of an implemented system (if one exists), through consulting specifications or interviewing developers – depending on what the current stage of development is.

Thus, the human factors perspective is closely mirrored by the focus on user concepts, while the developer's perspective is mirrored in system and interface concepts.

6.2 Example of application to a DL

For this example, we draw on the second of the proof of concept studies outlined above. We found that users' core concepts included:

- *Library as place* – to study, to meet people, somewhere quiet.
- *Books* – may be chosen, and have a status such as issued or returned.
- *Sections* – physical areas of the library that store material that is thematically linked.
- *Librarian* – a person to seek guidance and information from.
- *Date* – of issue, when book due back, etc.

Clearly, many of these concepts do not apply to DLs – so what concepts do users have to work with in a DL such as the ACM DL?

- *Users* – have status such as not registered, registered or subscribing (personally or via an institution).
- *Articles* – may be from journals, newsletters, transactions, etc.

- *Search facilities* – including search, advanced search, and browse.
- *Feedback* – about problems or satisfaction.
- *Binders and the ToC Service* – features that can be useful and powerful, but that the user does not need to know about to be able to retrieve documents from the library.

A preliminary misfit analysis is shown in Figure 6. Since this is difficult to decipher, we present the analysis in more detail below.

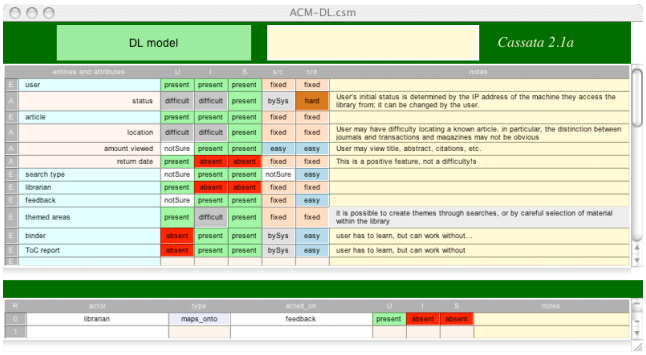

Figure 6: Preliminary CASSM analysis of ACM DL

The main misfits identified are as follows:

- The nearest equivalent there is to the librarian in the DL is the feedback facility. However, the difference between a librarian and the system feedback facility is substantial, and may cause difficulties for new users.
- Articles do not need to be returned; this is probably a pleasant surprise for novice users!
- However, the different levels of 'subscription' are not clearly presented to new users, who may have difficulty understanding what they are entitled to and the significance of institutional membership (if this is available to the user).
- DLs provide powerful search capabilities that are not available in traditional libraries; users need to learn to use these features effectively.
- The distinctions between journals, transactions, proceedings and magazines may be non-obvious to new users. In the music library, this difficulty presented itself in an analogous form: that the user had to learn about the structure of collections within the library.
- There is no immediate digital equivalent of themed sections, although a user can create such areas by searching on themes.
- Unfamiliar concepts such as 'binders' and the 'TOC service' are available, and need to be learnt if users are to use the library in powerful ways (but can be ignored safely in the meantime). However, such features are only available to individual subscribers to the library.

6.3 Findings and discussion

CASSM does not deal with usability issues at the level of detail of (for example) Heuristic Evaluation. It takes more of a broad-brush approach in considering key user and system concepts.

Even with just two analyses, we have found that it is possible to re-use much of the user side of analysis in assessing the two DLs: there was much commonality in the findings from the two studies, even though those from the music DL went beyond text-based analysis to investigate musicians' conceptualization of the music information retrieval domain. We would expect this to extend to any further DLs analyzed. In other words: CASSM locates itself in the DL domain by drawing on empirical data from users, which can be assessed against system-oriented descriptions of particular systems.

Compared to the three analysis techniques described above, CASSM lacks detail. It has also proved more difficult for novices to learn than either HE or CW. However, it has provided insights that complement those of more traditional evaluation techniques.

7. DISCUSSION AND CONCLUSIONS

We have investigated four usability-oriented design and evaluation techniques as applied to digital libraries. We have applied a form of action research, in which we have aimed to understand usability issues for DLs and the application of user-oriented techniques by selecting, applying and reflecting on techniques drawn from the human–computer interaction literature and our own work, to extant DLs and DLs under development.

Our aim has not been to develop a complete analysis of any particular library, but to investigate the scope and limitations of various techniques, with the long-term aim of developing a collection of techniques that can usefully be applied to support reasoning about usability in the DL context.

In this work, we have found that to assess the usability of DLs, knowledge of DLs and their users is essential. This may sound obvious, but is, in fact, not explicitly acknowledged by techniques such as HE & CW, both of which were originally developed and tested in contexts that are familiar to most HCI practitioners (such as word processing and web navigation). Thus, HE and CW have been found to address superficial aspects of DLs, but not the more complex issues of information seeking and use as discussed in section 2. For those surface aspects, HE covers broader scope than CW, but offers less support to the analyst in identifying potential user difficulties.

In contrast, CA and CASSM both probe deeper issues. CA demands that the analyst draw on a theoretical and empirical basis to construct scenarios and claims, while CASSM includes explicit gathering of the user perspective. Of these two techniques, CA is both the more demanding of the analyst, and the one that delivers a wider range of insights.

CASSM covers different territory from the other three techniques investigated, by focusing attention on user and system concepts rather than procedures or – in particular – scenarios. One point that CASSM has shown up about many libraries is that the concepts of a 'collection' and (variable) access rights are alien to most users. These are neither new nor surprising insights, and yet many extant libraries continue to include these features and to fail to explain them adequately to users, highlighting the ongoing gulf between users and system development.

One feature that is (broadly) common to CW and CA is the need for scenarios. In work with DL developers, we have found the use of scenarios particularly powerful. It appears that further work on developing a library of DL scenarios could be fruitful to guide development. Initially, scenarios could be

based on findings from the information seeking literature; further work is needed to develop scenarios covering other aspects of DL use, such as how people organize, and subsequently work with, information resources (currently implemented in features such as ACM's "binders"), how they share information with colleagues and collaborate over information, and how library creators or collection builders (the people who so often mediate between the developers of core technologies and end users) gather together information and present it to end users. These are all important aspects of DL creation and use that are currently under-researched.

Of the techniques discussed above, HE appears to be the most widely used by practitioners. This is consistent with a strong user preference expressed at the 2002 workshop on usability of DLs [6], where it was found that most participants felt that checklists would help them in assessing DL solutions. While this is hardly a scientific survey, it is at least indicative that checklists are perceived as providing good support for those given the responsibility for assessing potential DL solutions. Sandusky [24], for example, proposes a list of features of a DL to check; these include both user-oriented features and others (e.g. how payment is organized). Further research is needed on the effectiveness of checklists: while they may appear easy to use, there has been little research on what background knowledge analysts need to have in order to use them effectively, and in DLs – as with other domains – there is no such thing as a free lunch (i.e. no insight without effort).

In completing this work, we have found two gulfs that need to be bridged in creating DLs that are both technically sound and also responsive to user requirements. The first is between user-focused and technology-focused perspectives: the user focus places emphasis on the user experience, but at the expense of considering technical details, while the technology focus addresses technical issues at the expense of user concerns. As noted above, scenarios of use provide one promising route to bridging this gulf; approaches such as CASSM, which explicitly bring together system and user perspectives, also show promise.

The second gulf is between problem- and solution-focus in design. Most HCI evaluation techniques have traditionally focused on understanding problems, as a prerequisite to developing solutions to those problems. In contrast, developers tend to be driven by solutions, needing only to understand enough about a problem to be able to generate a promising solution to it. Of the techniques considered here, only CA makes explicit its role within ongoing design, as a tool to support reflection on design decisions and hence on possible design modifications. There is a clear need for closer integration between evaluation and re-design – a need that is not peculiar just to DL development.

In summary, we can identify several important conclusions that emerge from this work:

- *Bridging gulfs*: there is a need to bridge gulfs between user and system perspectives on the design of DLs. Scenarios and CASSM both appear to be promising approaches in this regard.
- *Depth of analysis*: HE and CW yield useful insights into surface features of DL design. However, they do not probe deeper usability issues. For those, good understanding of DL users' perspectives is needed. This deeper understanding is only available through techniques such as CA and CASSM, which explicitly draw on empirical findings about users. CASSM does not address details, but takes a broader-brush approach by considering core concepts for the user and system. CA can be used to address issues at different levels of description, but is correspondingly more time-consuming to apply.
- *Territory*: HE, CW and CA all focus on procedures, whether expressed through tasks or scenarios. CASSM focuses on user concepts, and thus identifies different kinds of user issues from the other three approaches tested.

Unsurprisingly, none of the techniques tested has been found to deliver the perfect solution. Nevertheless, each has contributed some part to the overall understanding of how DLs can be considered from a user-oriented perspective. Further work, both empirical and analytical, is needed to develop evaluation techniques that are grounded in an understanding of users and their information work. The work reported here is a step in this direction.

8. ACKNOWLEDGMENTS

This work has been funded by EPSRC grants GR/N37858 and GR/R39108. This work would not have been possible without our collaborating partners, and we thank them for their contributions. Thanks also to Thomas Green and anonymous reviewers, who made useful comments on an earlier version of this paper.

9. REFERENCES

[1] Attfield, S., Blandford, A. & Dowell, J. Information seeking in the context of writing: a design psychology interpretation of the 'problematic situation'. *Journal of Documentation*. 59.4, 2003, 430 - 453.

[2] Bates, M. J., Wilde, D. N. & Siegfried, S. Research Practices of Humanities Scholars in an Online Environment: The Getty Online Searching Project Report No. 3. *Library and Information Science Research* 17, 1995, 5-40.

[3] Bates, M. The Design of Browsing and Berrypicking techniques for the Online Search Interface. *Online Review*, 13,5, 1989, 407-424.

[4] Bates, M. The cascade of interactions in the digital library interface. *Information Processing and Management* 38, 2002, 381-400.

[5] Blandford, A, Keith, S., Lodder, T. & Fields, B. *Claims Analysis 'in the wild': a case study on Digital Libraries*. Submitted for journal publication. Available from www.uclic.ucl.ac.uk/annb/DLUsability/DLtools.html

[6] Blandford, A. & Buchanan, G. Report on Workshop on Usability of Digital Libraries at JCDL'02. In *SIGIR Bulletin*, 36.2 (Fall 2002). Available from http://www.acm.org/sigir/forum/F2002-TOC.html

[7] Blandford, A. & Buchanan, G. Usability of digital libraries: a source of creative tensions with technical developments. In *IEEE-CS Technical Committee on Digital Libraries' on-line newsletter*, 1.1 (Summer 2003). http://www.ieee-tcdl.org/Bulletin/summer2003/

[8] Blandford, A. E., Stelmaszewska, H. & Bryan-Kinns, N. Use of multiple digital libraries: a case study. In *Proc. JCDL 2001*. ACM Press, 2001, 179–188.

[9] Borgman, C. L. *From Gutenberg to the global information infrastructure: Access to information in the networked World*. Cambridge, MA: The MIT Press, 2000.

[10] Carroll, J. M. & Rosson, M. B. Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Transactions on Information Systems*, 10.2, 1992, 181-21.

[11] Cockton, G., Woolrych, A., Hall, L. & Hindmarch, M. Changing Analysts' Tunes: the Surprising Impact of a New Instrument for Usability Inspection Method Assessment. In: *People and Computers XVII: Proceedings of HCI'03*, Springer, 2003, 145-161.

[12] Connell, I., Green, T. & Blandford, A. Ontological Sketch Models: Highlighting User-System Misfits. In E. O'Neill, P. Palanque & P. Johnson (Eds.) *People and Computers XVII, Proc. HCI'03*. Springer, 2003, 163-178.

[13] Cooper, A. *The Inmates are Running the Asylum*. Sams Publishing, Indianapolis, 1999.

[14] Duncker, E. Cross-cultural usability of the library metaphor. In *Proceedings of JCDL'02*, ACM Press, 2002, 223-230.

[15] Ellis, D & Haugan, M. Modelling the information seeking patterns of engineers and research scientists in an industrial environment. *Journal of Documentation* 53.4, 1997, 384-403.

[16] Hertzum, M., and Jacobsen, N.E.. The Evaluator Effect: A Chilling Fact about Usability Evaluation Methods. *International Journal of Human-Computer Interaction*, 13.4, 2001, 421-443.

[17] Ingwersen, P. Cognitive perspectives of information retrieval interaction: elements of a cognitive IR theory. *Journal of Documentation* 52.1, 1996, 3-50.

[18] Keith, S., Blandford, A., Fields, R. & Theng, Y. An investigation into the application of Claims Analysis to evaluate usability of a digital library interface In A. Blandford & G. Buchanan (Eds.) *Proc. Workshop on Usability of Digital Libraries at JCDL'02*. 2002. Available from www.uclic.ucl.ac.uk/annb/DLUsability/JCDL02.html

[19] Komlodi, A. & Marchionini, G. Key frame preview techniques for video browsing. In *Proc. ACM Digital Libraries 1998*, 1998, 118-125.

[20] Kuhlthau, C. C. & Tama, S. L. Information search process of lawyers, a call for 'just for me' information services. *Journal of Documentation*, 57.1, 2001, 25-43.

[21] Kuhlthau, C. Longitudinal case studies of the information search process of users in libraries. *Library and information science research* 10.3, 1988, 257-304.

[22] Nielsen, J. Heuristic Evaluation. In J. Nielsen & R. Mack (Eds.), *Usability Inspection Methods* New York: John Wiley, 1994, 25-62.

[23] Papadakis, I., Andreou, I. & Chrissikopoulos, V. Interactive Search Results. In M. Agosti & C. Thanos (Eds.) *Proc. ECDL 2002*. Springer LNCS 2458, 2002, 448-462.

[24] Sandusky, R. J. Digital Library Attributes: Framing usability research. In A. Blandford & G. Buchanan (Eds.) *Proc. Workshop on Usability of Digital Libraries at JCDL'02*. 35-38, 2002. Available from www.uclic.ucl.ac.uk/annb/DLUsability/JCDL02.html.

[25] Spencer, R. The Streamlined Cognitive Walkthrough Method, Working Around Social Constraints Encountered in a Software Development Company, *Proc. CHI'2000*, 2000, 353-359.

[26] Stelmaszewska, H. & Blandford, A. Patterns of interactions: user behaviour in response to search results. In A. Blandford & G. Buchanan (Eds.) *Proc. JCDL Workshop on Usability*. 2002. Available from www.uclic.ucl.ac.uk/annb/DLUsability/JCDL02.html.

[27] Sutcliffe, A. & Carroll, J. M. Designing claims for reuse in interactive systems design *Int J Human-computer studies* 50, 1999, 213-241.

[28] Vakkari, P. A theory of the task-based information retrieval process: a summary and generalisation of a longitudinal study. *Journal of Documentation*. 57.1, 2001, 46-60.

[29] Warren, P. R. Why they still cannot use their library catalogues. *Proceedings of Informing Science Conference* June 19-22 2001 Krakow, Poland Cracow University of Economics. 2001. Available from http://ecommerce.lebow.drexel.edu/eli/.

[30] Wharton, C., Rieman, J., Lewis, C. & Polson, P. The cognitive walkthrough method: A practitioner's guide. In J. Nielsen & R. Mack (Eds.), *Usability inspection methods* New York: John Wiley, 1994, 105-140.