

Preprint: final version of paper available as: BLANDFORD, A., KEITH, S. & FIELDS, B. (2006) Claims Analysis 'in the wild': a case study on digital library development. *International Journal of Human-Computer Interaction*. 21.2. 197-218. DOI [http://dx.doi.org/10.1207/s15327590ijhc2102\\_5](http://dx.doi.org/10.1207/s15327590ijhc2102_5).

## **Claims Analysis 'in the wild': a case study on Digital Library development**

Ann Blandford<sup>1</sup>, Suzette Keith<sup>2</sup> & Bob Fields<sup>2</sup>

<sup>1</sup> UCL Interaction Centre, University College London, Remax House, 31-32 Alfred Place, London WC1E 7DP, UK. [a.blandford@ucl.ac.uk](mailto:a.blandford@ucl.ac.uk)

<sup>2</sup> Interaction Design Centre, Middlesex University, Trent Park, Bramley Road, London, N14 4YZ, UK. {S.Keith, B.Fields}@mdx.ac.uk

**Corresponding Author:** Ann Blandford (contact details shown above)

**Title short form:** *Claims Analysis 'in the wild'*

### **Abstract**

One of the long-standing challenges in HCI has been the integration of usability evaluation methods within design practice. In the work reported here, we investigated the question of how to include user concerns within an unstructured, system-focused development process. The project investigated the use of Claims Analysis as a method for assessing the effects of design decisions on users' experience. Claims Analysis was found to be more difficult than expected to learn, to communicate to systems developers and to apply effectively in practice. The work has highlighted a tension between user-centred and function-oriented design approaches, and differences in values and perspectives between the human factors specialists and traditional developers involved in the study.

**Keywords:** Digital Libraries; Claims Analysis; Scenario Based Design; usability evaluation; software development processes.

# 1 Introduction

One of the long-standing challenges in HCI has been the integration of usability evaluation methods within design practice. Approaches such as contextual design (Beyer and Holtzblatt, 1998), participatory design (Muller and Kuhn, 1993) and scenario based design (Rosson and Carroll, 2002) have been developed to incorporate user perspectives within the design of new systems. However, such approaches demand a complete commitment to user-centred design: a commitment which not all development teams are willing or able to make (Heinbokel *et al*, 1996). The work reported here recognises this and instead addressed the question: can we tailor usability evaluation techniques to be incorporated effectively and useably within system-oriented design practice? In particular, Claims Analysis (CA: Carroll & Rosson, 1992; Carroll, 1996; Carroll, 2000; Carroll & Rosson, 2003) was adopted and adapted as a technique to bridge between evaluation of an existing system and redesign of that system.

A case study approach was taken, working with the developers of a corporate Digital Library (DL): that of BT, a large UK telecommunications company. The focus of this paper is on the experience of applying Claims Analysis to the evolutionary design of that Digital Library.

Digital libraries are structured repositories of electronic documents, in various media (text, sound, graphics, animation, etc.). They are widely perceived as being difficult to use (e.g. Borgman, 2000; Blandford, Stelmaszewska and Bryan-Kinns, 2001). For example: the way information is structured in them can be difficult for users to assimilate quickly, and often does not conform to the ways users think about their topics of interest; they typically support powerful ways of searching that are unfamiliar to most users; and they often present results of a search in ways that are difficult to understand. However, their structure makes them accessible to powerful searching techniques, and the fact that they are managed gives users assurance about the quality of the DL contents. In addition, many digital libraries include features that can enhance the user experience, when compared to using a simple internet search or traditional library. Examples of such enhancements include personalised document management systems and notification systems that allow the user to share a document discovery with others.

The study reported here is located in the real world of system development (as advocated by John (1998)). This account draws on various experiences over a three-year project that started with the broad aim of developing user-oriented tools that were applicable in DL design practice. The work has been exploratory and developmental rather than following a more traditional, science research paradigm. For presentational purposes we describe the background and our own work under two initial themes: development and evaluation of digital libraries (section 2), then Claims Analysis (section 3). The two case studies that form the core of this paper are presented in section 4, followed by discussion (section 5) and conclusions (section 6).

## **2 DL development practices**

DL environments typically include components from many different sources: Bates (2002) describes the user's experience of interacting with a DL in terms of a 'cascade' of different interacting components. For example, the end user's experience of a DL is likely to be determined by the design of a web browser (through which the DL is accessed), the technical infrastructure (e.g. the networking and interoperability protocols), the document management software and the collection building system. Additional features may have been provided by yet other development teams. In this complex development environment, in which basic functionality and interoperability of components present substantial technical challenges, user concerns can become secondary.

### ***2.1 Usability in DL development practice***

Historically, much work on usability of DLs has been *post hoc* – at least in the sense that a fully functioning prototype system exists to be tested (e.g. Papadakis *et al*, 2002; Komlodi & Marchionini, 1998). More recently, there has been an interest in check-list based approaches that make explicit some of the desirable features of DLs (e.g. Kyrillidou & Giersch, 2005); however, these are reported as being used to evaluate existing DLs, and we are not aware of any reports of such approaches being applied within ongoing DL development. There are accounts of user-centred design of DLs (e.g. Champeny *et al*, 2004), but the process is described as iterative or participatory, without any clear account of the use of tools or structured methods to support evaluation or design reasoning. For

example, Champeny *et al* describe their design process as involving “iterative, formative evaluation, a process in which system design is studied in parallel with user needs and requirements”.

## **2.2 *A starting point: understanding the development context***

The aim of the work reported here was not to introduce a design process that would have been alien to the developers with whom we were working, but to equip them with usability evaluation tools that they could use with only small adaptations to their established design practices.

The first few months of the project were devoted to familiarisation – with the library, development practices and users – and with investigating possible evaluation techniques.

### **2.2.1 The library**

The BT digital library is a corporate library providing user access to all the electronic resources to which BT subscribes. These include electronic books (e.g. the O’Reilly series of computing books was reported to be popular with staff) and abstracting services such as INSPEC (IEE, 2006), as well as full-text journals. It also aims to provide ‘value added’ services to support the research community. These include a set of ‘information spaces’ that allow users to keep track of new information in their specialist areas, and collaborative tools enabling users to share information they find, both of which are described in section 4 (below).

### **2.2.2 The development context**

The development is coordinated by three key members of library staff, who have the technical capability to deal with some aspects of the development (notably interface implementation), and who also have the usual kinds of contact with library users visiting the physical library, and with information suppliers. New systems development is interleaved with their other librarian duties. Development of underlying technical infrastructure is devolved to specialists. The DL had undergone three major upgrades, in terms of features offered and user interface, in the previous six years. Development was evolutionary and did not follow any standard methodology; the BT librarians described their development process as very informal – that they tended to have an idea, implement a prototype and then discuss the design-as-instantiated with colleagues to get reactions. In this paper,

we refer to the library staff as developers when discussing their development role, and as librarians when referring to their librarianship skills; both terms refer to the same three individuals.

Their knowledge of user difficulties and requirements was derived primarily from users calling them or coming into the library with queries and requests. They also accessed system logs to ascertain levels of use and which facilities or resources were being most extensively used. Around the time of each new release, they would conduct user trials, focusing on finding bugs and design flaws in the system; they tended to give users in these trials detailed instructions on what to do and what features to test.

### **2.2.3 The users**

Users of the library are all employees (or interns) at BT. Initial interviews with a selection of users (nominated by the library staff as representing the broad cross-section of library users) indicated a need for novel information in relation to new projects. These users were experts within their own domain but novices with respect to the requirements of a specific project. As intermittent users, they had little opportunity to build up their knowledge of the library and, due to changes and upgrades to the interface, remained permanent novices. They used few of the features – using simple keyword and phrase-based searching and perhaps registering for one of the browsable information monitoring topics.

### **2.2.4 The evaluation techniques tested**

In the early stages of the project, various user-oriented design and evaluation techniques were investigated, with two particular concerns in mind:

- They should be adaptable to fit within existing DL design practice: to make any impact at all, we needed techniques that were acceptable to DL developers, and that could co-evolve with their design practices.
- Techniques should be ones that could support reasoning about the kinds of issues that make DLs difficult to use, such as how to formulate queries, how to probe the DL to identify appropriate search terms and how to deal with uncertainty in searching.

The techniques studied in this phase were user testing, Cognitive Walkthrough (Wharton *et al*, 1994) and Heuristic Evaluation (Nielsen, 1994). These techniques were chosen as ones that were believed to be relatively easy to learn and apply. In addition, user testing provided a forum for meeting a sample of users, and finding out more about their knowledge and use of the digital library. While all three of these techniques gave useful insights about superficial aspects of design, such as the suitability of labels and the quality of feedback, we found that they did not give leverage on deeper usability issues concerning information seeking (Blandford *et al*, 2004). About nine months in to the project, we decided to adopt Claims Analysis as our primary method for structuring evaluation to inform redesign.

### **3 Claims Analysis: adoption and adaptation**

As reflected in the theme of this Special Issue, there are few published techniques that bridge between evaluation and design practice. In particular, we needed an approach that supported reasoning about users' cognitive processes and would mesh well with existing development practice and culture as outlined in the previous section. The technique that appeared to best satisfy these criteria was Claims Analysis, because CA is flexible enough to permit rich argumentation about a variety of cognitive issues, while being able to function in a relatively stand-alone way, making it possible to accommodate it with an existing development context. Although Claims Analysis has, in recent publications (e.g. Rosson and Carroll, 2002), been included as a small step within a full scenario-based design process, in earlier papers (e.g. Carroll and Rosson, 1992) it was presented more as a stand-alone technique (albeit requiring the analyst to develop scenarios of use). Claims Analysis explicitly aims to draw on cognitive and other relevant theories to support reasoning about the effects of the design on the user. It also fits within ongoing design practice: the analysts' reflection on design strengths and weaknesses could directly inform design changes.

The first stage was to learn CA ourselves. This involved substantial reading and practice, followed by time spent encapsulating it in a form that supported communication to DL developers. A tutorial on CA for DLs was developed and delivered at two international DL conferences (Fields, Keith &

Blandford, 2003). In parallel, we applied the approach with the BT developers, on various features of their library as it underwent development. Key Claims Analysis sessions are described in section 4.

### ***3.1 Claims Analysis: introduction***

Claims Analysis has been described in many sources by Carroll and co-workers. The most comprehensive account is provided by Carroll and Rosson (1992); a succinct summary is provided by Sutcliffe and Carroll (1999).

Carroll and Rosson (1992) describe Claims Analysis as 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). Generating these claims requires that scenarios of use have been formulated; then claims capture strengths of the design (that should be retained in any re-design) and possible problems that merit further investigation or indicate areas to pay attention to in re-design.

To illustrate claims with a simple example, consider the following scenario: “Alice is a new member of the research team working on usability of digital libraries. She is a graduate software engineer who has recently joined the company. She is unfamiliar with the information resources available and the details of digital libraries, and want to find out what is available in her subject area. She accesses the company digital library via a link from the company home page and observes a page on which the most prominent item is a search box. She types “digital libraries” into the search box and presses the “go” button.” A positive claim about the search box is that its prominent position will encourage Alice to look for information by entering a keyword or phrase. A negative claim is that Alice may not know what syntax to use or how to best formulate her query.

### ***3.2 CA in design***

CA is embedded within a particular view of design evolution which Carroll and Rosson term the ‘task-artefact cycle’. Under this view, existing user tasks may be used to define requirements on new artefacts to support those tasks, but those new artefacts create new interaction possibilities that change users’ tasks, so that we get an evolving interplay between design and use.

Carroll (1998) states that he does not view CA as primarily an evaluation method; rather, claims “are intended to help developers think about what changes to specify by helping them become more aware of the tradeoffs among consequences of possible design changes for users and their activities” (Carroll, 1998, p.242). Nevertheless, Carroll, Koenemann-Belliveau, Rosson and Singley (1993) discuss the use of CA as an evaluative technique that supports an understanding of critical incidents (i.e. surprising incidents in an interaction – often ones where something has gone unexpectedly wrong).

Claims can support design decision making: Sutcliffe, Fickas, Sohlberg and Ehlhardt (2003) report on the use of claims to structure a contextualised interpretation of design principles to support reflection on multiple design alternatives for an assistive user interface to an electronic mail / communication program. They describe claims as a “set of trade-offs that have to be resolved” – in other words, as a tool to support reflection and re-design. Sutcliffe and Carroll (1999) focus more explicitly on structuring claims for re-use. A similar approach is taken by de Bruijn and Spence (2005), who use claims as a way of encapsulating insights about cognition and design alternatives for presenting information.

### ***3.3 Use of CA outside the original development team***

As far as we can ascertain, few formal studies of CA have been conducted by anyone outside the original developer group. John and Marks (1997) include CA in a set of six techniques adopted by a group of students to evaluate an interface to a novice programming environment. Each technique is allocated a numerical score indicating how many problems (categorised under various headings) were identified using it, and how many of those problems were subsequently dealt with by the system developer. The authors conclude that “CA (circa 1991) did not lead to any design changes. We speculate that its ‘claim ... but’ format, presenting both advantages and disadvantages of a design makes it difficult to see clearly a better way to implement the system.” (John & Marks, 1997, p.199).

Theng *et al* (2004) report on their adaptation of CA to a particular learning context by generating a set of 19 questions, adapted from questions from the CA literature, which could be answered by

representative users (in their case, teenagers) acting as evaluators. Iverson (2003) also reports on applying CA to reason about the design of a digital library, but gives little detail of either the experience or the findings from doing so.

None of the extant accounts of using CA explicitly consider the challenges and benefits of engaging (non-user-centred) developers in generating and reasoning with claims, as has been done in the work reported here.

### ***3.4 How to apply CA: generating Claims***

The clearest guidance on applying CA is provided by Carroll and Rosson (1992), who present a list of nineteen questions to ask as a basis for generating claims. These questions are organised according to the seven stages of Norman's (1986) action cycle, which describes user–system interaction from a user perspective as encompassing user goal formation, intention formation, action specification, action execution, perception of system state, interpretation of that state and evaluation of that state (with respect to user goals). See Figure 1. For example, under the 'action execution' step, Carroll and Rosson propose three questions: how does the artefact make it easy or difficult to perform a task? What slips are most likely, or most costly? And how does the artefact indicate progress in task performance? Sutcliffe and Carroll (1999) propose something similar, but at a higher level of abstraction, suggesting (p.236) that an appropriate approach is “to apply an extended form of cognitive walkthrough to inductive claims analysis”. Carroll (1996) extends the original nineteen questions with a further eighteen that help identify organisational claims; these are structured according to the organisational goals and how an organisation (or team) engages in planning, execution, interpretation and evaluation of activities. Most other accounts of CA appear to assume that it is obvious to the reader how to generate claims, providing rich examples of the outcomes of CA, but little guidance on generating them.

Figure 1 about here

### **3.5 How to apply CA: developing scenarios**

Claims are generated in the context of scenarios: descriptions of “the activities a user might engage in while pursuing a particular concern” (Carroll and Rosson, 1992, p.185). Sutcliffe and Carroll (1999) describe three different kinds of scenario that can be used with CA: *problem initiation* (describing the situation prior to re-design); *usage* (describing a sequence of user–system interaction, based on empirical evidence); and *projected usage* (describing anticipated interaction with a re-designed artefact). Carroll and Rosson (1992) discuss the selection of scenarios in terms of typical and critical use scenarios: focusing on the tasks people perform most frequently and those that are most likely to cause problems. The main source of scenarios discussed by Carroll and Rosson is empirical studies of how users work; in principle, it is also possible to create them by reflecting on how users are intended to work (notably when creating new interaction possibilities).

### **3.6 Our codification of Claims Analysis**

As far as possible, our use of CA followed that in the published literature as described above (sections 3.4 and 3.5). Within the first year of the project, three refinements were made:

- To help DL developers “put themselves in the users’ shoes”, we embellished scenarios with personas, as described by Cooper (1999) and later by Rosson & Carroll (2002), who refer to them as ‘stakeholders’ (that text was not available in the early stages of our study).
- Extensive use was made of the information seeking literature to provide background accounts of established information seeking behaviour (see section 3.7). This could be used both for providing the ‘bigger picture’ for developing scenarios and personas, and for generating claims.
- We found that the ‘craft skill’ approach to generating claims that was implicit in most accounts of scenario-based design and CA provided too little structure. Conversely, the 19 + 18 questions proposed by Carroll and Rosson (1992) and Carroll (1996) made the claims generation process too onerous to be acceptable to our development teams. A ‘middle way’, in which we focused on three stages of Norman’s action cycle, was adopted. In other words,

we considered goal formation, action execution and evaluation of feedback for each stage of a scenario, and sought to generate both positive and negative claims for each of these phases of interaction. The net effect is similar to the approach suggested by Sutcliffe and Carroll (1999) of basing claims generation around the Cognitive Walkthrough questions.

Figure 2 about here

In summary, the approach developed for applying Claims Analysis, as encapsulated in our tutorial on the subject and applied in our interactions with developers, was as follows (see also Figure 2):

1. Describe the design or design feature (at a level of detail appropriate to the current stage of development).
2. Propose personas and scenarios to evaluate that design feature. Sources for these include HCI and information seeking theory, and also local knowledge of users and their tasks and queries (see Figure 3).
3. Develop claims by considering the interaction in detail, considering at each step of the interaction what goals the user would form, what actions they would execute and what feedback they would receive. This involved working systematically through an interaction corresponding to a scenario and generating claims. Example questions for generating claims are shown in Figure 4.
4. Assess the positive and negative features of the claims generated, considering their importance and the ease of changing the design to retain positive features while eliminating negative ones.

Figures 3 & 4 about here

Since information seeking theory is central to understanding how users interact with digital libraries, we briefly summarise the theories of information seeking that were most influential in our thinking before presenting the details of the study.

### **3.7 Information seeking theory**

Two ideas from the information seeking literature were particularly influential in our thinking. The first is that information needs are not always well defined, and that the user of a digital library may need to interact with the library over an extended period of time to find out what is available and to refine the information requirement to a point where it can be clearly expressed through a search query. Digital libraries need to support such poorly formulated information requirements. The process of moving from uncertainty to satisfaction of information needs is succinctly summarised by Kuhlthau's (1998) Information Search Process (ISP) model. Kuhlthau identifies six stages of the information search process through which an information seeker moves on the path from initial uncertainty, through exploration, to understanding. Early on in a new search, users may not even know what 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, and the user may have difficulty assessing the relevance of documents to their information task. A browse-based interaction can be more effective than search at this stage, depending on the thematic organisation 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. Then, when browse or search results are returned, the user has to be able to assess the relevance of each document rapidly and reliably. This ISP provided a structure for generating scenarios based around the different stages of information seeking, and hence of generating claims about features of the library under consideration.

The second key idea is encapsulated in Bates' (1989) work on "berrypicking". In this view, the information seeker starts with a need (which may be poorly formulated) and as they interact with information sources they pick up both resources ("berries") and also new ideas of ways to further the search, and so the search evolves over time. Again, this model provided a way of thinking about scenarios, claims and design solutions.

## 4 Engaging with developers

Having familiarised ourselves with the library, its users and the development culture, and investigated other evaluation methods (section 2.2), and studied and codified Claims Analysis (section 3.6), it was possible to involve the developers themselves in Claims Analysis. There were two formal CA sessions: an exploratory one that focused on a novel feature called Jasper, then a more structured one on the information spaces and other design features; finally there was a debrief session to establish the developers' perceptions of CA as an approach. Between these phases, there were informal discussions about users' needs and practices, and also a study of the librarians' own information seeking skills (reported by Fields *et al*, 2004), which maintained an ongoing dialogue between developers and HF specialists on understanding users with different levels of expertise, and also the needs of the developers. The overall process is summarised in Figure 5.

Figure 5 about here

The two formal CA sessions involved participant observation: one HF specialist participated as a member of the design team, bringing her knowledge of Claims Analysis into the design context and engaging the rest of the development team in user-centred discussion based around scenarios and claims. In the Jasper session, which lasted about 2 hours, data collection was informal, being based on observation notes. The focus of these notes was on the design rationale and claims and scenarios that were developed. The second session, which was a whole day and focused attention on the information spaces, was audio-recorded; for this session, data was transcribed and analysed using an 'emergent themes' approach (Wong and Blandford, 2002) to establish how the developers engaged with scenarios and claims. Data collection was as unobtrusive as possible, to minimise disruption to the ongoing activity.

### 4.1 *Jasper: Assessment of a novel feature*

Jasper was a web site notification feature, developed by another group within BT, that enabled the user to send an interesting URL to nominated colleagues, including a note about why the user thought it was interesting. It also took a snapshot of the web site that it stored in case the website vanished.

Jasper is an example of a novel design feature for which there was no prior user data to indicate how it had been used, or theory to predict future use.

The question was presented as *where* to locate it within the library, and how to present it to users. In particular, the developers believed that the data entry form should be redesigned, and they wanted to focus on that. However, it was not possible to say much about the form without creating a scenario accounting for how a user would have reached this point. The scenario that was developed involved a user surfing the web and finding a page that was relevant to one of their colleagues, and deciding to use Jasper to draw their attention to the page with an explanatory note of why the page looked interesting. Creating such a scenario highlighted the fact that the user would have to enter the digital library (while keeping track of the interesting URL) to access the Jasper feature, which in turn presupposes that they are aware of, and value, the Jasper feature. This led to a claim that if the user was not aware of the Jasper feature and the way it functions then they could not form a goal to use it.

Further discussion about the details of the interaction scenario highlighted the fact that the user was given no feedback when they had sent the form. This was articulated as a claim that the lack of feedback would leave the user unsure whether or not their action had had the intended effect. The latter was a solvable problem, so one of the BT developers immediately went to the whiteboard to map out how he could create a preview / feedback page that addressed the identified difficulty. He was not entirely happy with this solution as it added a step to the process of working with Jasper (claim: the effect of adding a feedback page is to create an additional action for the user to perform), making it less efficient, but the process of generating claims based on the user's knowledge and goals had persuaded him that it was a sensible addition to the system. In this instance, the stories were not written down, but provided a way to structure discourse and design reasoning.

This first CA session involved a form of apprenticeship for both parties: each learning from the other through their mutual engagement with the design activity.

## 4.2 *Information spaces: Assessment of an established feature*

The subsequent CA session at BT was more structured, including an explicit knowledge transfer element. It involved three phases: presentation of a shortened version of the tutorial; development of personas and scenarios; and CA. The CA considered the whole library in various degrees of detail, but focused particularly on the information spaces (shown in the centre of the screen in Figure 6). An information space is a browsable collection of documents on a specified topic, to which new documents are regularly added. Each is specialised so that it will not grow unmanageably large. The information spaces were originally designed to enable marketing specialists to keep abreast of developments in their areas of specialism. They were known to work well at a functional level, but web logs showed that they were not widely used, and the developers wished to know how their design could be improved.

Figure 6 about here

Three personas and scenarios (each combining an individual and a searching context) were constructed by the BT developers. The personas covered users with different backgrounds; in creating them, the developers drew heavily on people they knew, but also drew on theory (notably Bates' (1989) "berrypicking") to deliberately construct less skilled users because they were aware that it is the less skilled users that have most problems. The scenarios associated with these personas covered use of several features of the library, including the keyword browser and information spaces. They named their three personas after the Goodies (a UK TV show from the 1970s).

- Tim was an expert in his subject, and had done a preliminary search on credit card fraud. He had requested help on following links from a relevant journal (a berry picking approach which was supported by existing system tools, but which was known to cause users difficulties).
- Graeme was an expert in his field but new to the DL. Working against a tight deadline, he had no time to waste. The scenario described a basic search and then simple strategies, including

system support for selecting appropriate resources, changing key terms and using the keyword search feature.

- Bill was doing an MBA, and therefore using the resource for professional development, pushing the boundaries of the resources beyond the core business content. He had a preference for books, and therefore wanted to search the book catalogue, but was having difficulty using phrase search.

These scenarios, generated by the developers, share some notable properties. They all reflect real cases where users had come to the library seeking help; they were all based on existing library features that were known to be difficult to use; all the personas lacked a good understanding of the search process, library features, the structure of the database, and the choice of resources; however, two of the personas represented people who were experts in their own fields. Because the personas were based on known people, the developers did not perceive any value in expanding the level of detail included, as advocated by Cooper (1999): to have done so would have been to invest time in activities that were not perceived as productive.

The developers reported favouring problem scenarios where the information seeking was less than instantly successful on the basis that such scenarios would provide more useful insights into problems and solutions.

The scenarios, described from a user perspective, were then considered alongside the system description to assess whether users with the specified skill sets (as defined within the personas) would know what to do and would understand the results they were getting back. For example, when analysing the information spaces, it became apparent that they are much more appropriately regarded as a monitoring feature for expert users than a good place for a novice user to start, unless the user happens to be lucky and have a requirement that neatly matches an existing topic area. This is illustrated in the following discussion between the Human Factors specialist (HF) and one of the developers (D1):

HF: So what you said was that you're worried the information spaces may distract Graeme.

D1: Yes, on this page, on the home page there are all of these topics: they may be taking away from the emphasis on the search box. [See Figure 6.]

HF – Mmm, and it might not match what he's looking for so it's not a positive distraction.

D1 – In his case, it doesn't match what he's looking for.

This extract illustrates an informal negative claim (about the possible distraction caused by the information spaces). The same claim was counterbalanced by a positive one about the visual salience of the search box later in the discussion:

HF: [...] the search box is fairly strong on the home page, but you might worry that you're getting a distraction effect, from the information spaces which don't really match what he's looking for.

Having worked with information spaces for several years, this was the point where both the developers and the human factors specialist realised explicitly why they did not provide good support for novices.

Later in the discussion, considering how the user would review a set of search results for relevance prior to selecting a document, the HF specialist articulated further claims, clearly linked to the phase of forming a new goal. This is dependent on the relevance of the results presented:

HF: in terms of process, of understanding what to do, yes it's telling you what your result is about, but you're in a situation of forming a new goal and deciding what to do, and how to do it, and then deciding which most closely matches your results – it's fine if something does closely match the results, that you can feel confident about.

Some claims were not about the design explicitly but about the user experience within a particular scenario of use:

HF: on the good side he found some articles quickly and on the bad side he found some links that were not working, and didn't follow them up

Overall, while the developers engaged with personas and scenarios, the generation of claims was left almost entirely to the HF specialist, and many of them were only ever informally articulated.

However, the developers did start to consider explicitly what users would know in the context of a scenario, and what goals they would form. For example, discussing Tim and the information spaces, one of which is labelled ‘Business and Management’:

D1: What are you doing: ‘credit card fraud’. So, he’d go, he’d think well it must be about ‘Business and Management’. If he knew that, that was all he needed to click. I’m not sure there’s anything there though.

Here, we see one of the developers predicting reasonable user behaviour, but then recognising that that reasonable action is unlikely to achieve the intended effect – another aspect of the information spaces not always achieving their intended purpose. Later, the developers extended the discussion into problem solutions:

D1: Um, well if you don’t find anything, there’s a link to us for help.

D2: we should make that more prominent for a start!

D1: “I suggest you check your spelling and use the phone number for help”.

The HF specialist expanded the scenario to investigate new design solutions; the following example is referring to a feature called the keyword browser, a search tool which analyses the index terms used to categorise the documents. In expert hands it is a powerful search tool, but it is not immediately intuitive to a casual user like Tim:

HF: Yeah, it’s a good feature and its powerful, but it’s connecting it in- if we stick with Tim and his storyline, what would Tim want from it? What Tim would want from it would be some words that are better than the words that he’s got. So if we think about it by the definition of what supports Tim, then what you want Tim to do is form a goal that says ‘I don’t think the words I’ve got are very good, can I go and find some better words from this stuff which is vaguely in the right area but not really that good for me.’ In that case the keyword browser term is suggesting he can go and find some better terms

Although this session was explicitly set up to focus on CA, as illustrated in the examples above, the developers did not engage well with the causal aspects of claims. They worked readily with the scenarios, and switched seamlessly between scenarios, problem diagnosis and solution generation.

Through their choice of scenarios, the developers focused on problems that were in principle solvable; they:

- were not interested in features of their design that were assessed to be working well;
- did not perceive any value in identifying problems that could not be fixed; and
- were not interested in pursuing the theoretical cause of a problem unless that contributed directly to fixing it.

This session enabled the developers to get a richer sense of the difficulties less experienced users face and a deeper understanding of the user experience than they previously had. Personas (based on real users) and scenarios (based on real incidents) that were examined at the level of detail determined by a consideration of goals, actions and feedback were central to this richer understanding; claims *per se* provided more of an ideal target to aim for than a genuinely useful tool.

## 5 Discussion

One of the real challenges for HCI researchers is developing design and evaluation techniques that are theoretically grounded, but that are also taken up and worked with by others. This demands two things: the provision of ways to make the technique accessible to people who were not involved in the original development of the approach, and an understanding of how the technique can fit within real design practice (the theme of this special issue). We consider each of these issues in turn.

### 5.1 *Learning and applying CA*

As illustrated by the timeline in Table 1, learning and applying CA took much longer than one might expect. This was partly because it proved necessary to both develop a deep understanding of the approach ourselves and find effective ways to communicate it to developers (initially done informally through apprenticeship; later via the tutorial) and partly because DLs proved to be a particularly challenging object of study for the approach (see section 5.1.2). The best learning resource we found was Carroll and Rosson (1992); Sutcliffe and Carroll (1999) provided a useful alternative perspective, although the focus of that paper is more on re-use than initial generation of claims. Most of the other

publications that discuss CA at all do so peripherally, leaving the reader with the impression that generating claims is so easy that it does not even need to be explained explicitly. The presentation of examples divorced from the cultural context in which they were generated makes it difficult to generalise from them. Because we found the shortage of procedural descriptions of how to apply CA a disadvantage, we generated our own procedural description, as outlined in section 3.6.

Applying CA presented two particular challenges: developing good scenarios, and identifying the effects on the user (i.e. generating claims). We consider each of these in turn.

### **5.1.1 Scenarios and personas**

In a function-oriented design process, defining scenarios at an appropriate level of detail proved difficult. As discussed in section 4.1, it demanded a substantial shift in thinking to move from “how shall we link Jasper into the library?” to “how will people use Jasper?”. Because Jasper was creating new interaction possibilities rather than upgrading existing ones, the only option was to create projected usage scenarios. Initially, it was difficult to establish common ground between the developers’ traditional technical orientation and the human factors specialists’ user-centred orientation.

Defining scenarios for an existing feature (information spaces: section 4.2) was less difficult, because the developers have regular contact with their users and based scenarios on them and their experiences. The personas (Tim, Graeme and Bill) helped them distance themselves and their expertise from that of their users. Thus, although empirical studies were used (of both novice and expert searchers), together with established information seeking theory, the main source of inspiration to the developers was their own experience. This arguably gave them a greater sense of ownership of and confidence in the scenarios.

In both cases, the developers’ strong preference was to focus on scenarios that predicted user difficulties *that could be fixed with obvious design changes*. They were not interested in either scenarios that highlighted no user difficulties or ones that highlighted difficulties that they could not address.

### 5.1.2 Generating and evaluating claims

As described in section 4.2, developers did not engage well with generating claims. Even for the HF specialist, it was often difficult to work out what the positive and negative consequences on a user would be, and to assign a theoretically grounded cause. If, as generally happens in search-based interfaces, the user is simply presented with a search box, what is the effect on the user? It clearly depends on who that user is: if we assume that the user recognises a search box for what it is, there are still questions about how the user will formulate a query, resulting in non-determinism in what they might do and how they will perceive and interpret the results they get back. Even with the help of information seeking theory and basic cognitive theory, we found ourselves struggling with this kind of scenario. Maybe we got the level of description of the scenario wrong: we could answer this kind of question if our scenarios were extremely detailed (to the point of specifying the formulation and results of each search); however, if we tried to generalise over cases, it was difficult to be confident about effects, because there was so much non-determinism in the interaction.

In practice, it proved to be much easier to think in terms of causes of user difficulties. The developers had enough contact with users, as well as access to system logs, to be able to identify places within their existing designs where users often had difficulties or did inappropriate things or failed to use particular features that could help them. Drawing on psychological or information seeking theory, it was possible to argue that a user could not perform an action because they had not formed an appropriate goal or they lacked a knowledge of the domain or they lacked information retrieval skills, or whatever. But that was an explanation of why the user was having a problem; it was not saying what the effect of the design was on the user.

It was possible to extend that to make basic claims of an effect that this particular part of the interface does not convey to the user what to do. This may be considered a negative effect, although it is perhaps more accurately an *absence of effect*: the user receives neither positive guidance nor misinformation about what they might do next. A lot of the problems that the users experienced were situations where the tools were available to them but they did not know how to use them, so the tool

was not affecting the user: the tool was not providing the support the user needed because it never occurred to the user to use it.

Developers found it much easier to identify positive effects of their own designs. Developers are highly motivated to deliver usable, effective design solutions, and are much more aware of their positive motivations than of any difficulties users might experience. However, they described positive features with more hope than confidence: hope that they would be useful and usable, without either theory or empirical studies to back up that hope. Their normal reviewing practice was devoted to finding bugs or identifying where things did not work as expected. The personas, scenarios and three-stage process of generating claims about goals, actions and feedback helped to provide theoretical context for identifying claims; however, developers found it hard to identify weaknesses of their favoured solutions, when they had already devoted effort to ensuring the technical feasibility of those solutions.

## ***5.2 Integrating CA with traditional design practice***

As noted in section 1, Claims Analysis demands the use of scenarios to support reasoning, and hence fits most comfortably within a totally scenario-based design approach, as described by Rosson and Carroll (2002). In this project, our aim was different: to tailor user-centred evaluation techniques to the particular domain of digital library design, taking a participatory development approach – more “usability by stealth” than user-centred design. This exposed some fundamental differences in values and perspectives between the DL developers and HF specialist.

The development team had a functional approach to thinking about design. They are not unique in this: in our work with the developers of the New Zealand Digital Library ([www.nzdl.org](http://www.nzdl.org)), as yet unpublished, we found a similar approach; Hartson *et al* (2004) speculate that the developers of the NCSTRL digital library took a similar approach, noting that “an important category of common problems that we observed involves an apparently functionally-oriented design approach rather than an approach based on user task threads”. Because of this functional perspective, the developers had difficulty perceiving how, from the user’s perspective, all these functions might to be joined up into a

continuous interaction experience as described in theories of information seeking. The developers started several discussions by presenting the human factors specialist with a particular issue – for example, where should this page be linked in to the rest of the system, or how should it be laid out? – without considering the broader context of how the user would have got to that point in the interaction, or why they were there. Claims analysis could not proceed from this point without probing the context and creating scenarios, however informal.

There was a cultural gulf to be bridged: the human factors specialist had difficulty comprehending the interaction from a technical perspective, while the developer teams had equal difficulty in comprehending it from a user perspective. Over time, scenarios, personas and claims were used to create common ground, but the process of building this ground was slow and difficult. And the process of generating scenarios was reversed from the text-book account, in that we typically worked from functions that developers were concerned about to generate scenarios that involved users working with those functions.

As well as thinking about design in terms of the functionality a system supports, another important characteristic of the developer perspective was that it was solution-focused. Developers were, frankly, not interested in *post hoc* reflection on the qualities of a design; consequently, they only really engaged with CA when it could help identify solvable problems, and analysis was often interleaved with solution generation and discussion. They valued the insights from theory, but only when those insights could guide problem-fixing.

## 6 Conclusions

In electing to work with CA, our initial assumption was that it should be possible to create a library of scenarios and claims that could then be readily re-used by DL developers (Sutcliffe and Carroll, 1999). As illustrated through the preceding account, this assumption was flawed. We have identified several reasons for this:

- The cultural gulf between the development team and ourselves was substantial. They were function- and solution-focused, whereas we were scenario- and user-focused. Scenarios and

claims proved useful for creating common ground between these perspectives, but the gulf remains.

- Each scenario only relates to a small subset of the DL functionality, and developers' interest shifts from one function to another, limiting the potential for re-use.
- DL development is, in our experience, very ill-structured, typically involving interoperating components developed by distributed teams who may have no direct contact with each other. At the least, this creates distributed responsibility for delivering a user experience that is an emergent product of a complex web of design decisions.
- Some of the functions (notably Jasper) were novel designs that created new interaction possibilities, so that there was neither relevant theory nor empirical data on which to base scenarios and claims. This also made it more difficult for the developers to get a sense of ownership of any scenarios developed.

Nevertheless, the use of personas, scenarios and claims helped deliver important design insights (e.g. about how to position Jasper within the library, the importance of feedback, and how to better target the information spaces), and to bridge the gulf between conflicting perspectives. Using claims as a communication tool proved to be more effective than writing them down: the process of discussing the scenarios and who the users are and what they are trying to do generated productive dialogue between the developers and human factors specialist. Thus, the power is in the dialogue and potential for creating shared understanding. Sutcliffe and Carroll (1999, p.214) assert that claims can “deliver HCI knowledge to designers”. This was more difficult than anticipated, and we do not believe that the developers we worked with either could or would use claims unsupported. Indeed, in the debrief, the BT development team explicitly stated that scenarios are very useful, but generating claims is “too academic”.

Our experience is necessarily coloured by the development context in which we were working, and by the skills and personality of the HF specialist who engaged in the participant observations. It appears to be impossible to eliminate the “evaluator effect” (Hertzum & Jacobsen, 2001), and it may be that

different individuals coming together in a slightly different design context would have had a substantively different experience. Nevertheless, this study provides a data point in the broader enterprise of understanding the challenges of integrating a usability perspective within non-user-centred design practice. DL design and deployment is often ill-structured, involving semi-detached but interdependent strands of development. The whole process is much more organic and opportunistic than is assumed in descriptions of design processes, including scenario-based design (Rosson and Carroll, 2002). This complexity makes it difficult to apply scenario-based design in anything like a textbook fashion. More critically still, the culture of the development team was highly function-focused, and the developers do not want problems: they want solutions. This illustrates the importance of HCI researchers knowing our users: design and evaluation techniques need to be usable by, and acceptable to, developers if those developers are to be empowered to deliver systems that are, in turn, usable by their intended audience.

Did CA ‘work’ in this context? Certainly not in the smooth way projected in publications by Carroll, Rosson, Sutcliffe and their co-workers. It proved to be difficult to learn, communicate effectively to developers, and apply within their ongoing design process. Nevertheless, the cultural shift in being forced to consider personas and scenarios as a precursor to generating claims provided a strong foundation for creating common ground between divergent perspectives on design.

## **7 Acknowledgements**

We are grateful to the development team at BT who worked with us on this project and created such fruitful interactions, and to tutorial attendees who provided useful feedback on the approach. The work was funded by EPSRC Grant GR/N37858.

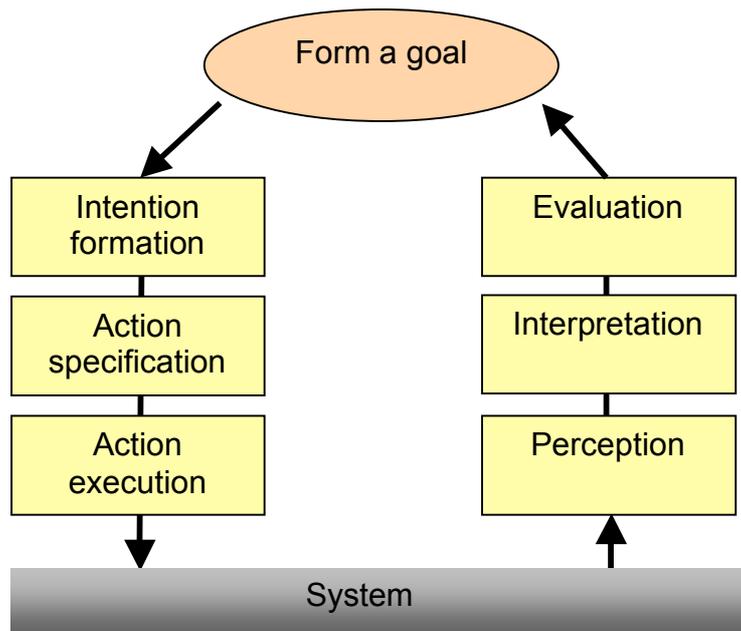
## **8 References**

- Bates, M. (1989) The Design of Browsing and Berrypicking techniques for the Online Search Interface. *Online Review*, 13,5. 407-424.
- Bates, M. (2002) The cascade of interactions in the digital library interface. *Information Processing and Management* 38:381-400. Available from [www.gseis.ucla.edu/bates/articles/cascade.html](http://www.gseis.ucla.edu/bates/articles/cascade.html)

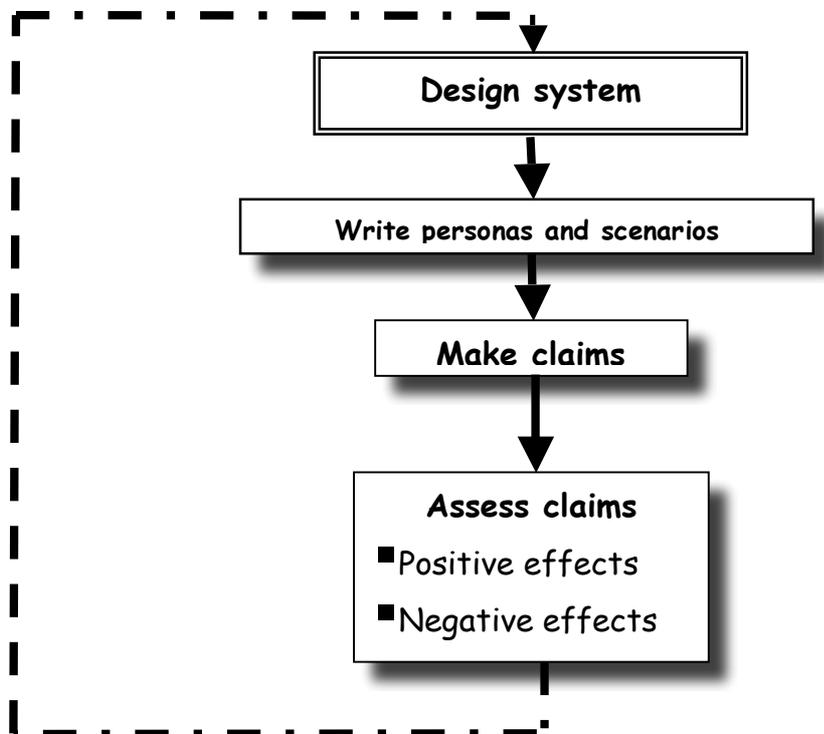
- Beyer, H. & Holtzblatt, K. (1998) *Contextual Design*. San Francisco : Morgan Kaufmann.
- Blandford, A., Keith, S., Connell, I. & Edwards, H. (2004) Analytical usability evaluation for Digital Libraries: a case study. In *Proc. ACM/IEEE Joint Conference on Digital Libraries*. 27-36.
- Blandford, A. E., Stelmaszewska, H. & Bryan-Kinns, N. (2001) Use of multiple digital libraries: a case study. In *Proc. JCDL 2001*. ACM Press. pp 179–188.
- Borgman, C. (2000) *From Gutenberg to the Global Information Infrastructure: Access to Information in the Networked World*, The MIT Press.
- Carroll, J. M. (1996) Becoming social: expanding scenario-based approaches in HCI. *Behaviour and Information Technology*. 15.4. 266-275.
- Carroll, J. M. (1998) On an experimental evaluation of claims analysis. *Behaviour and Information Technology*, 17(4), 242-243.
- Carroll, J. M. (2000) *Making use: scenario based design of human computer interaction*. MIT Press
- Carroll, J. M., Keonemann-Belliveau, J., Rosson, M. B. & Singley, M. K. (1993) Critical Incidents and Critical Themes in Empirical Usability Evaluation. In J. Alty, D. Diaper and S. Guest, Eds. *People and Computers VIII, Proceedings of HCI'93*, 279 - 292. Cambridge: Cambridge University Press.
- Carroll, J. M. & Rosson, M. B. (1992) Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Transactions on Information Systems*, 10(2), 181-21.
- Carroll, J. M. & Rosson, M. B. (2003) Design Rationale as Theory. In J. M. Carroll (Ed.) *HCI Models, Theories and Frameworks: Towards and Multidisciplinary Science*. San Francisco: Morgan Kaufmann. pp. 431 - 461.
- Champeny, L., Borgman, C., Leazer, G., Gilliland-Swetland, A., Millwood, A., D'Avolio, L., Finley, J. & Smart, L. (2004) Developing a Digital Learning Environment: An Evaluation of Design and Implementation Processes. In *Proc. JCDL 2004*. 37-46.
- Cooper, A. (1999) *The Inmates are Running the Asylum*. Indianapolis: Sams Publishing.
- De Bruijn, O. & Spence, R. (2005) *Theory-based Interaction Design: From Cognitive Theory to Interactive Artifacts*. Information Engineering Report 2005/1, Department of Electrical and Electronic Engineering, Imperial College London.
- Fields, B, Keith, S. & Blandford, A. (2003) *Usability Evaluation of Digital Libraries: A Tutorial*. Middlesex University Interaction Design Centre Technical Report: IDC-TR-2003-001. Presented at JCDL'03 and ECDL'03. Available from <http://www.cs.mdx.ac.uk/research/idc/techreports.htm>.
- Fields, B., Keith, S. & Blandford, A. (2004) Designing for Expert Information Finding Strategies. In S. Fincher,

- P. Markopoulos, D. Moore & R. Ruddle (Eds.) *People and Computers XVIII – Design for Life, Proc. HCI04*. Springer. 89-102.
- Hartson, H. R., Shivakumar, P. & Pérez-Quñones, M. A. (2004) Usability inspection of digital libraries: a case study. *International Journal of Digital Libraries*. 4.2. 108-123.
- Heinbokel, T., Sonnentag, S., Frese, M., Stolte, W. & Brodbeck, F. (1996) Don't underestimate the problems of user centredness in software development projects – there are many! *Behaviour and Information Technology*. 15.4. 226 - 236.
- Hertzum, M., and Jacobsen, N.E. (2001) The Evaluator Effect: A Chilling Fact about Usability Evaluation Methods. *International Journal of Human-Computer Interaction*, 13(4), 421-443.
- IEE (2006) Inspec. <http://www.iee.org/publish/inspec/>. Viewed 14<sup>th</sup> February 2006.
- Iverson, E. (2003) The Starting Point Project: Moving Beyond Usability and Web Logs to Refine Evaluation. Position statement for NSDL workshop. Available from [http://eduimpact.comm.nsd.org/evalworkshop/\\_iverson.php](http://eduimpact.comm.nsd.org/evalworkshop/_iverson.php) (viewed 11.11.05).
- John, B. & Marks, S. (1997) Tracking the effectiveness of usability evaluation methods. *Behaviour and Information Technology* 16, No. 4/5, 188-202.
- John, B. (1998) On our case study of claims analysis and other usability evaluation methods. *Behaviour and Information Technology* 17, 4. 244-246.
- Komlodi, A. & Marchionini, G. (1998) Key frame preview techniques for video browsing. In *Proc. ACM Digital Libraries 1998*. 118-125.
- Kuhlthau, C.(1988) Longitudinal case studies of the information search process of users in libraries. *Library and information science research* 10 (3) pp. 257-304
- Kyrillidou, M. & Giersch, S. (2005) Developing the DigiQUAL protocol for digital library evaluation. In *Proc. 5th ACM/IEEE-CS joint conference on Digital libraries*, 172 - 173
- Muller, M. J. & Kuhn, S. (1993) Participatory design. *Communications of the ACM*, 36.6. 24 - 28
- Nielsen, J. (1994) Heuristic Evaluation. In J. Nielsen & R. Mack (Eds.), *Usability Inspection Methods* (pp. 25-62) New York: John Wiley.
- Norman, D. A. (1986) Cognitive Engineering. In D. A. Norman and S. W. Draper, Eds. *User Centered System Design*, Hillsdale NJ: Lawrence Erlbaum, 31-62
- Papadakis, I., Andreou, I. & Chrissikopoulos, V. (2002) Interactive Search Results. In M. Agosti & C. Thanos (Eds.) *Proc. ECDL 2002*. Springer LNCS 2458. 448-462.

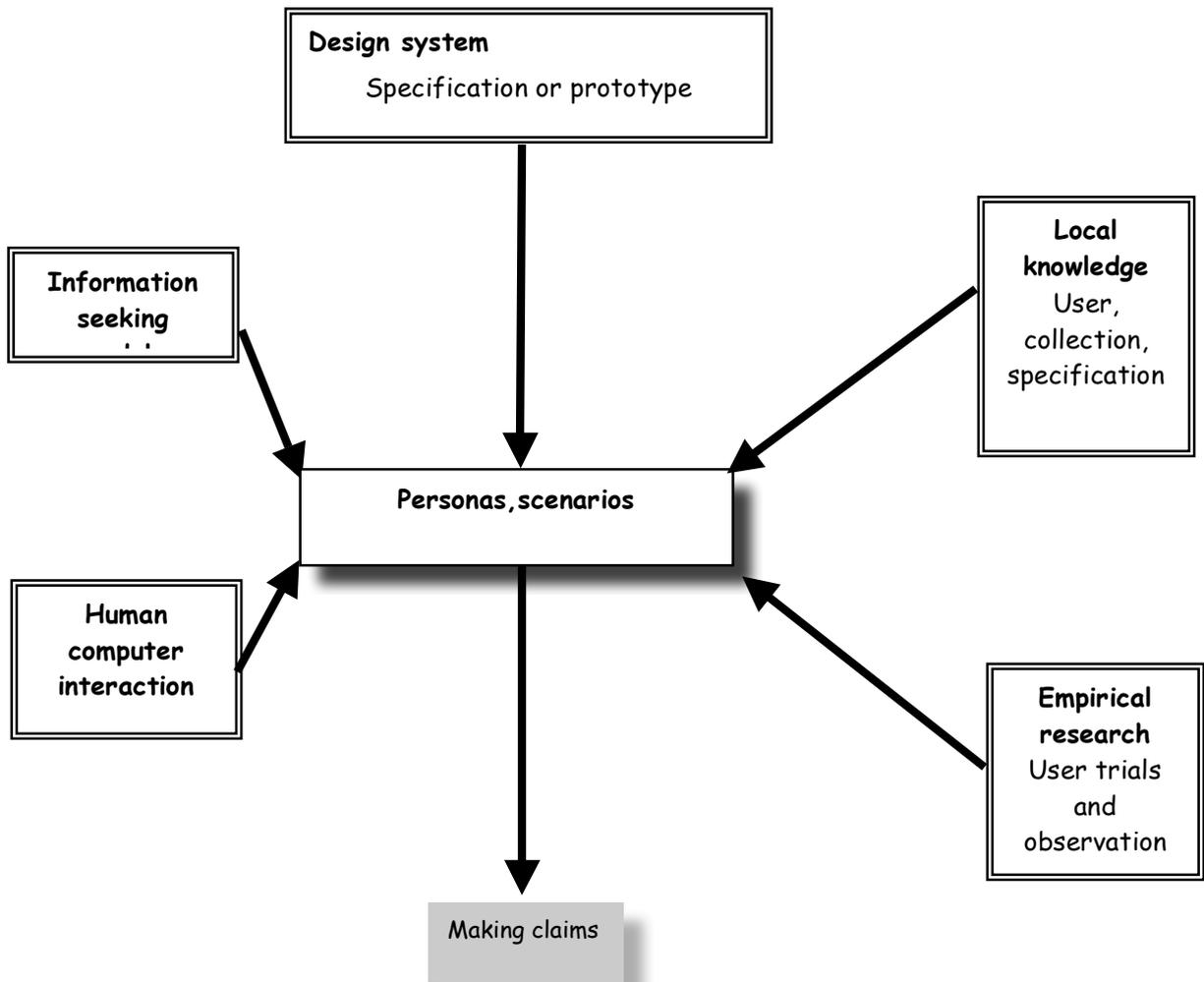
- Rosson, M. B. & Carroll, J. M. (2002) *Usability Engineering*. San Francisco: Morgan Kaufmann.
- Sutcliffe, A. G. & Carroll, J. M. (1999) Designing claims for reuse in interactive systems design *Int J Human-computer studies* 50 213-241
- Sutcliffe, A., Fickas, S., Sohlberg, M. M. & Ehlhardt, L. (2003) Investigating the usability of assistive user interfaces. *Interacting with computers* 15. 577-602
- Theng, Y.L., Goh, D., Lim, E.P., Liu, Z.H., Yin, M., Pang, L.S., Wong, B.B. (2004), Applying Scenario-Based Design and Claims Analysis on the Design of a Digital Library of Geography Examination Resources. *Information Processing and Management Journal*, 41, 23-40.
- Wharton, C., Rieman, J., Lewis, C. & Polson, P. (1994) The cognitive walkthrough method: A practitioner's guide. In J. Nielsen & R. Mack (Eds.), *Usability inspection methods* (pp. 105-140) New York: John Wiley.
- Wong, W. & Blandford, A. (2002) Analysing Ambulance Dispatcher Decision Making: Trialling Emergent Themes Analysis. In *Proc. HF2002* (electronic publication).



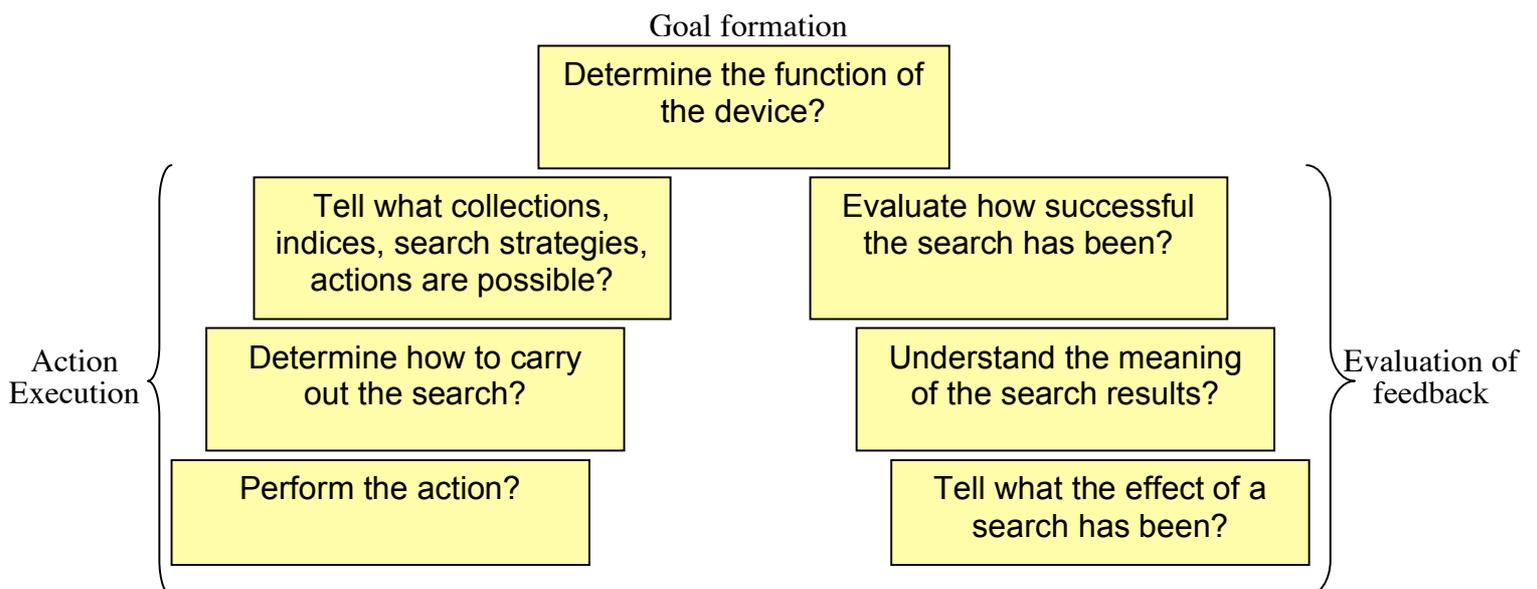
**Figure 1: Norman's Action Cycle**



**Figure 2: including Claims Analysis in the system development process**



**Figure 3: The sources of information for generating personas and scenarios**



**Figure 4: Examples of questions corresponding to the different phases of Norman's (1986) Action Cycle for generating claims**

Activities	Timeline
Phase 1 Investigation of usability evaluation techniques. Investigation of development context.	January 2001
Phase 2 Focus on CA. Interview developers to explore their design rationale. Conduct first case study (Jasper: section 4.1). Study BT DL novice users in context.	January 2002
Phase 3 Study BT librarians' information seeking skills (Fields <i>et al.</i> , 2004) Conduct second case study (Information Spaces: section 4.2) Develop and deliver tutorial at 2 conferences.	January 2003
Phase 4 Debrief with BT Developers.	January 2004

Figure 5: timeline for the study

The screenshot shows the BT Digital Library website. At the top, there is a teal header with 'BTexact' in large white letters and 'Digital library' in smaller white letters. Below the header, there are navigation links: HOME, SITE MAP, A TO Z, and CONTACTS. The main content area is divided into several sections:

- My**: A vertical list of links including BookmarkBank, Journals, Options, Password, Reading, Requests, Searches, and Spaces.
- Useful Links**: A vertical list of links including About Us, Acronyms, BT Patents, Datapro, Good Websites, Journal List, Library Databases, MediaWatch, O'Reilly Books, Talk to Us, and Translation Service.
- HELP!**: A simple text link.
- BT Digital Library**: A central section with a search bar containing 'adsl', a 'Search All Areas' dropdown, and a 'Go' button. Below the search bar is a search tip: 'Search tip: Adding phrases will give you better results eg 3G wireless, third generation wireless, 3rd generation wireless, broadband wireless.' Below this is a welcome message: 'Welcome to the Digital Library. (Already a user? [Log in](#) to get more information. New users should [register](#) to take full advantage of the Digital Library.)'
- Business & Management**: A section with links to [Project Management](#), [CRM](#), [Knowledge Management](#), [E-commerce](#), and [Personalisation](#).
- Computing & Software**: A section with links to [XML](#), [Content Management](#), [Software Technology](#), [Intelligent Systems](#), and [Application Service Providers](#).
- Communications & Networking**: A section with links to [Voice over IP](#), [Bluetooth](#), [Mobile and Cellular](#), [Network Management](#), and [Large scale IP network design](#).
- Industry Sectors**: A section with links to [E-Government](#), [Medicine](#), [Utilities](#), [Retail and distribution](#), and [Consumer Products](#).
- Companies**: A section with links to [BT](#), [Cisco](#), [Accenture](#), [Logica](#), and [Oracle](#).
- Science & Technology**: A section with links to [Optical Components and Materials](#) and [MEMS](#).
- Need help?**: A button with a right-pointing arrow.
- What's this?**: A section with a link to [Register for the Digital Library](#).
- What's hot in the library?**: A section with links to [Popular library journals](#), [What are people reading now?](#), and [What's New?](#)
- What's cool?**: A section with links to [Information Spaces](#), [BookmarkBank](#), [Contents pages](#), and [Add a note](#).

Figure 6: The BT Library interface