SCHOOL OF ECONOMICS
AND MANAGEMENT
Lund University

Department of Informatics

Exploring Usability Guidelines for Rich Internet Applications

Master thesis, 10 credits, Department of Informatics

Presented: June 2007

Authors: Lukas Gwardak
Linus Påhlstorp

Supervisor: Konrad Tollmar

LUND UNIVERSITY
Informatics

Exploring Usability Guidelines for Rich Internet Applications

Master thesis, presented June, 2007
Supervisor: Konrad Tollmar
Size: 80 pages

Abstract:

Usability guidelines are commonly considered a useful tool for developers to enhance the usability of interactive systems. They represent distilled knowledge from many disciplines related to usability and provide developers with solutions and best practices to achieve usability goals. However, the newly developing field of highly interactive web applications (Rich Internet Applications) still lacks appropriate usability guidelines. This work takes desktop usability guidelines and web usability guidelines as a basis to create an outline of Rich Internet Application usability guidelines. Three professional developers are being interviewed in order to get an insight into their work with guidelines and get their ideas of how possible Rich Internet Application guidelines should be structured.

Keywords:

Usability, accessibility, guidelines, user interface, web design, Rich Internet Applications, Web 2.0

Contents

Index of Figures

List of Abbreviations

AJAX	Asynchronous Javascript And XML
ASP	Application Service Providing
API	Application Programming Interface
CSCW	Computer-Supported Collaborative Work
CSS	Cascading Style Sheet
CVS	Concurrent Versioning System
DHTML	Dynamic Hypertext Markup Language
GUI	Graphical User Interface
IRC	Internet Relay Chat
HCI	Human-Computer Interaction
HTML	Hypertext Markup Language
OLPC	One Laptop Per Child
RIA	Rich Internet Application
RCP	Rich Client Platform
SSH	Secure Shell
UI	User Interface
VoIP	Voice Over IP
WAI	Web Accessibility Initiative
WCAG	Web Content Accessibility Guidelines
W3C	World-Wide-Web Consortium
WYSIWYG	What You See Is What You Get

1 Background

1.1 Introduction

The presented work deals with usability considerations of so-called RIAs (Rich Internet Applications) within the Web 2.0 (O'Reilly, 2005) paradigm. The characteristics of these applications are to facilitate desktop-like user interaction with modern web technologies like AJAX (Asynchronous JavaScript and XML), Flash, Shockwave and increase the web application's interactivity, speed, and usability. The term "Web 2.0" summarizes developments on technical, social and collaborative level within the understanding of the web by its users. The definition of this term is still not clear and will be discussed more in detail in the literature review.

Regarding the development of usable applications, designers and developers are encouraged to use guidelines to make their applications or web pages consistent and predictable. Desktop usability guidelines have an older tradition and are considered to be mature enough to be used by developers. As the web is a fundamentally different medium, web design guidelines are more difficult to establish and are not followed that strictly. Additionally, there is a strong corporate identity factor in web design that causes a consistency vs. usability dilemma.

In case of web 2.0 applications, the situation is even more difficult, as they constitute a combination of desktop-like interaction and web design, which is bringing in all existing problems with it.

The purpose of this work is to explore, how possible usability guidelines for Rich Internet Applications could be structured, what they should contain and how they should be developed.

1.2 Problem Description

The problem we face in this work is the lack of interaction consistency in RIAs, which results in poor overall usability, compared both to static web pages and desktop applications. We assume that the usability of RIAs could improve if developers had guidelines on hand - just like traditional web site usability can improve when guidelines are being used (Adkisson,

2002; Nielsen, 2004). Our first literature research and conversations with developers have shown that no usability guidelines have been yet developed for RIAs.

A sub-question is how to deal with the problem of accessibility in RIAs. as accessibility is strongly linked to usability (Theofanos, 2003) and RIAs are a big problem for screen reader software (Gibson, 2005). Regarding this problem, we face the question how RIA developers should deal with this problem.

1.3 Research Question

The main question of the work will be, how RIA usability guidelines should be designed to be useful for web developers and at the same time support the affordances of current development trends in web applications.

In order to answer this question, we have to deal with the following sub-questions:

- ➔ How is web developers' work connected to guidelines?
- ➔ Are guidelines used at all? How are they used? What kind of guidelines are these?
- ➔ How do web developers expect RIA guidelines to be in order to be useful?
- ➔ What do web developers think of users' mental models of RIAs?

1.4 Delimitations

As the "Web 2.0" and web technologies in general are a big, rapidly changing domain, we have to put certain limitations to our work and focus on specific aspects only. That means, we cannot focus too much on technical details of implementing these technologies, and we don't want to focus on the otherwise important social aspect of web 2.0. We would like to regard RIAs that are unique to the web, and not merely re-implementations of desktop applications.

Due to the high volume of a full guideline document and the time available to complete this work, we will only develop an outline for RIA guidelines, which will serve as basis for further evaluation with developers and as a foundation for future work in this area.

2 Methodology

2.1 Research Strategy

To gain an insight into the work of developers we conducted a case study of companies involved in development of web 2.0 applications. We conducted the same study with multiple companies resulting in a multiple case study design (Yin, 2003, p.40). From this multiple case study we made a cross analysis, which helps us to get an answer to our research question.

We motivate the selection of the case study approach with the fact that we could explore particular cases, where web 2.0 applications were being developed and we had access to the developers themselves. The reason why we have chosen to involve the in this research is because they are the target audience for guidelines and they know most about guideline usage in their company. By conducting a case study we could identify problems and needs of developers in direct relation to their work, resulting in an in-depth study of particular cases (Creswell, 1998). The fact of designing an application within a company puts some restrictions onto developers: the company's goals have to be fulfilled and still the application has to be usable and finished on schedule, even with limited resources for usability testing. Thus, the developers' work is tied to its context, which calls for a case study to investigate it as a whole (Yin, 2003).

We could also generalize our findings about usability aspects and include them into our discussions of a guideline outline.

2.2 Research Methods

2.2.1 Literature Review

The literature review presents a general picture of the Web 2.0 and RIA usage context and shows which work has already been done in the field of designing usability guidelines. It helps us understanding the topic and focussing on the relevant parts before conducting the case study. For the readers, the literature review serves as a further introduction to the area, making them familiar with existing guidelines. The knowledge gained is crucial for designing

the case studies and interview questions in the following chapters. Also it serves as a basis for our discussions about designing an outline for RIA usability guidelines.

2.2.2 Case Study / Interviews

We conducted a case study by analyzing the work of professional designers in various settings (commercial & academic). The case study involves the background of the organizations the designers work for and employs interviews as a core data collection technique. Interviews are an important information source, as they show the “real world” situation from the point of view of our target group. To gain a balanced picture we interviewed both “traditional” web designers (working with conventional web technologies) and developers working primarily with RIAs. Additionally, we interviewed a guideline developer to gain a better insight into development methodologies for guidelines.

Due to the geographical distribution of our interview subjects the interviews were conducted on distance. Depending on availability and time frames we sent out the questions by e-mail beforehand for the subjects to prepare and conducted the interviews by VoIP (Skype). This has put some limitations to our ability to ask follow-up questions with hindsight and to the way we could interpret the answers. As we had limited time available for the interviews, it was crucial to formulate the initial questions in a very clear and precise manner. The interviews were digitally recorded for easy reference during the research. The fact that interviews were not conducted face-to-face also limited our ability to observe implicit behaviour of the subjects during the interview, thus concealing hidden emotions about the topics discussed. In addition to the actual questions we also had to deal with ethical issues of interviews. Therefore we created interview policies, where we explained the purpose of the interview and how the gained data will be used in our work. We believe that this step was crucial for creating an atmosphere of trust even before the interviews began. We also decided not to publish the interviewees’ names, only their companies’ names and their positions.

After finishing the interviews we analyzed and summarized the answers in a discussion subchapter after each interview. The condensed answers were used in the final discussion.

We conducted interviews to gain in-depth knowledge about how developers design Internet Applications. We have chosen developers and experts from various backgrounds to gain a broad perspective on the goals and requirements for web applications in various domains. We believe that selecting practitioners for the interviews will get valuable results for the research.

Our case study involved three companies / organisations:

1) **Netvibes**[1] is a start-up company providing a personalized starting page, which can be populated by widgets. A widget is a small, window-like page element that can be connected to a newsfeed, e-mail Inbox, weather forecast, search interface etc. Widgets are provided by Netvibes but third party widgets can be also included. We interviewed a usability consultant, who develops user experience guidelines for *Netvibes*.

2) **Mindlab**[2] is a company based in Esslingen, Germany, providing solutions for online user testing. It works for customers from the pharmaceutics, financial and automobile industries. We interviewed a user interface designer from *Mindlab*.

3) **sTeam**[3] is a web-based open-source environment for structuring and maintaining virtual knowledge spaces. It has been developed at the University of Paderborn in Germany and is an integral part of their e-learning research infrastructure. We interviewed a web developer who is dealing with the web interface of *sTeam*.

2.2.3 Usability Guidelines Discussion

In the main discussion chapter (ch. 5) we combine the knowledge gained from the literature review together with the results from the interviews to create a possible outline for RIA usability guidelines.

We begin by explaining why guidelines are important and how designers can use them to improve their work. We then carry on by discussing distinct qualities of usability guidelines (Preece 1994, pp. 26-28) that could be applied to RIAs. To do this, we look among other things for patterns in web 2.0 RIAs, identify similarities in existing guideline documents and use the interview data gained from developers to find even more desired features to include.

As stated before, this work doesn't present a ready-to-use set of usability guidelines for Rich Internet Applications, but rather discusses an outline on which guidelines could be based.

2.3 Research Quality

In order to enhance credibility of our work, we use methods to enhance research validity and we put a strong emphasis on the ethical aspect of our research. Both topics will be covered in the following.

---

[1] http://www.netvibes.com

[2] http://www.mindlab.de

[3] http://www.open-steam.org

2.3.1 Validity

We employ two methods to ensure validity of our research: triangulation and member validation (Seale, 1999).

Triangulation is a method used to strengthen research evidence by replicating it with different methods or gathering it from independent sources. We use triangulation by comparing our findings from the literature study with the empirical data from the interviews. We are aware of the fact that literature presents a more theoretical point of view, whereas a case study and especially interviews with developers will highlight practical aspects. Yet, if both data sources will generate similar outcome, then our research outcome will gain more validity.

Member validation on the other hand is used to check the accuracy of research accounts with the respondents involved in the study. In our case we ask present our theory about users' mental models of the web applications to our interviewees and ask them about their opinion. We find the risk of misinterpreting the theory by our interviewees fairly low, as the field of our study is in the core of their work.

2.3.2 Ethics

Using interviews as a data collection technique requires sensitivity and imposes ethical considerations. In fact, all research involving or affecting humans should be conducted with ethics in mind. Kvale (1996) points out three main points to be well thought-out when conducting interviews:

- Informed consent: the interviewees have to approve participating in the research, including knowing the subject and purpose of the research, and how their answers will be used. In our case we have informed our subjects beforehand about the purpose of our work and how we are going to deal with their answers. As all interviews have been recorded, we have stated the exact point when the recording starts and when it stops. The subjects were given the possibility to communicate any remarks off record after the interview.
- Confidentiality: interviews can sometimes contain personal details that should not be published. The interviewees must therefore be assured that their answers will be treated with highest confidentiality. In that way they also gain trust and can speak more freely. We have chosen to keep our interviewees anonymous, revealing only the companies they are working for and their positions within the companies. However, as the company sizes are rather small and the positions of our interviewees are unique, it could be possible to trace back the persons in question. We found that it was not possible to completely anonymize the interviewees by hiding all related

data, because this would require reducing the companies to the type of product they develop and hide many details that are directly linked to our research question, resulting in a more abstract work, and stepping away from the case study research strategy.

- Consequences: minimizing the risk of harm to the interviewees by balancing harm and benefits of the research is an overreaching principle when conducting research based on interviews. The type of harm possible during an interview in our work would less likely have a psychological nature (like the intimacy of a therapeutic interview) but rather a possible work-related conflict, if the employer of the interviewee would use his or her answers in a disadvantageous way. This could be the case if the interviewee would accidentally reveal information not intended for the public or if the employer would find out about something that can lead to negative consequences for the interviewee. Israel & Hay (2006) argue for going beyond avoiding harm, and act to benefit others. This includes the interviewees and their community, not only the scientific community. The basis of this approach is derived from the ethics of virtue and the simple fact that interviewees should be rewarded for their commitment. The central aspect of our work is finding out how usability guidelines could benefit creating Rich Internet Applications. If the outcome of the research will benefit someone, it will affect web developers who can do their job better, and in consequence, all web users.

3 Literature Review

In this chapter we present a literature review on the topics relevant to our research question. After introducing the basic affordances of the web, we explain how web applications fit into this mental model and how usability and accessibility are two inseparable concepts. Finally, we discuss existing usability guidelines, their purposes and implications.

3.1 Web Navigation

The World-Wide-Web, as envisioned by its inventor, Tim Berners-Lee was created to simplify cross-platform access to documents and unify access to database resources (Berners-Lee, 1999). The tool to access the web was the web browser, referring to the action of *browsing*, a read-only process of accessing information. Information was organized on web pages, which were formatted in a simple mark-up language (HTML: Hypertext Markup Language) and could be cross-referenced (hyperlinked). Every page had its distinct location (the URL) and could be bookmarked for random access (state-less). The browser controls included the basic four buttons: back, forward, reload and home, and thus stressing that one can move between hyperlinked pages, reload them to get the newest version and get back to a certain home page. In fact, browser controls helped the user to form a mental model of navigating in the web space and one of the success factors for the web surely was the simplicity of the web browser and the basic design didn't change a lot during the last 15 years (Fig. 1).

NSCA Mosaic 1.0 (1992) Mozilla Firefox 2.0 (2007)

Fig. 1: Web browser User Interfaces, 15 years apart

The only type of interaction usage of the web was accomplished with forms, which allowed simple textual information to be entered and submitted to the web server for further processing and generating a result page. This type of simple interaction was perfect for querying databases, using web search engines or submitting a guestbook entry. As these processes consisted of one single step, it was still perceived as page-based and state-less. HTML was meant to put a semantic structure to documents, indicate a heading, a paragraph, a sorted list, etc. It was never meant to specify how the document should be presented on the screen. This is the reason why early web browsers displayed web pages differently. It didn't matter if a heading was printed in 16 pt or 20 pt typeface or if the background was white or grey. The information was what mattered, not the presentation.

However, with growing popularity of multimedia-equipped computers, modems and public Internet Service Providers, the web has become commercial. It was not anymore an exclusive place for the scientific community, but it has become available to anybody. The web as a medium started to change rapidly: web sites started to grow bigger and more complex, which called for methods to structure the information available at a given site. Numerous best practices and guidelines emerged on how to structure the content and navigation of a website dependent on their type or target group (Fleming 1998, Schmeiser, 1997) and making them a more usable and enjoyable experience (Nielsen, 2000). Web pages started to make heavy use of graphics and multimedia, leading to yet more guidelines on how to design web pages optimally, making best use of available technology. The basic interaction paradigm inherent to the web at that time was a read-only, page-based, hyperlinked medium.

3.2 The Advent of Web Applications

Another type of web services started to evolve in the mid-1990's. *Hotmail*, the first web-based e-mail service caused a paradigm shift: an application moved from the desktop platform to the web. The term "web application" is not easy to define, as any active server-based form processing (i.e. search engine or mail form) can be technically seen as a web application. However for the user the term is tied closer to a kind of applications that are imaginable on the desktop and ideally provide desktop-like functionality. From an enterprise perspective the term is quoted *Application Service Providing* (ASP) and is often expected to be the future model of delivering applications to users (Graham, 2004).

Another kind of interactive web services was introduced by online shops. While on the one side, online shops worked like search engines and provided searchable product catalogues, they revolutionized mail order shopping being efficient, up-to-data and fast. The combination of a rich medium with instant ordering possibility, online payments, customer product reviews

and personalized suggestions wouldn't be possible in any other medium. Online shopping has created a new usage paradigm, an application, which makes only sense in the web medium and has quickly become one of the Internet's "killer apps".

Basically, all kinds of applications that can benefit from an online community can work well on the web. Online communities, featuring discussion forums, existed since the very beginning of the Internet in the form of the Usenet. Web-based bulletin board systems gained huge popularity, because they opened up the possibility for unifying the information or functionality of a website with a community, all in the same place.

Because web applications are based on client-server technology, each interaction with the application required contacting the server and loading a new page. This made the interaction slow if compared with desktop applications, but the added value was high enough for the users to use web applications. In case of an web-based email client the added value was ubiquitous access to email, in case of online shopping it was the comfort and independence of opening hours and in case of community-based applications it was the access to a community itself. A web application being a simple copy of a desktop application didn't make much sense if it didn't provide added value. It was too slow and complicated to develop with conventional web technologies to be used in a productive environment.

Technically, the concept of Application Service Providing reminds very much of the early days of mainframe computers with connected terminals. The application was running on the server and the terminals only provided means for text input and information presentation. A standard web browser doesn't do much more than that, except formatting the content before displaying it. This is however changing in regards to so-called Rich Internet Applications.

3.3 Rich Internet Applications

The term "Rich Internet Application" (RIA) describes a kind of web-based applications, which shifts interactive features of a web application towards the client (web browser). The goal of this measure is to make Internet Applications work more fluently and thus resemble desktop applications. The prerequisite for Rich Internet Applications is using the browser as a rich client, which can actively execute script programs (like in case of Javascript) or is equipped with plug-ins, which execute flash movies or Java applications. In contrast to traditional web applications where the web browser takes the part of merely displaying and formatting the information and proving forms for entering new information, Rich Internet Applications use active technologies like Javascript, AJAX (Asynchronous JavaScript and XML), Flash, Shockwave or Java to involve the web browser in active processing of the content. The idea behind Rich Internet Applications is about making web applications more responsive, allow direct content manipulation (like progressive disclosure) without having to request a new

page after each user action. The success of Rich Internet Applications is often attributed to their responsiveness (Norman 2006, Karnell 2004)

From the technological perspective Rich Internet Applications are client-server applications, so they are dependent on a web server, which provides all the code to be executed (with some portions of the code, including the database being executed on the server-side and oher portions of code, like JavaScript or Flash files executed on the client-side). Technical considerations and performance measurement techniques are discussed in Loosley, 2006, who distinguishes between active and passive measurement, discusses locations of measurement probes and points out complications when measuring Rich Internet Applications. As technical aspects are not at the main focus of this work, we won't explore them in depth, but just acknowledge that they exist.

From an interaction design point of view, Rich Internet Applications originate from web applications, making them fundamentally different from desktop applications: with Internet applications, users are used to enter data, submit it and wait for the server to deliver results. Errors or omissions will be only visible after the server has responded. Internet applications require the network connection to persist between the client and server for the application to be functional. If client and server get disconnected, the application becomes unusable. As Rich Internet Applications are still developing, there are often no mechanisms, which detect a connection loss and the user may eventually lose data if she didn't notice the connection breakdown. Gmail[4] approaches the problem by auto-saving newly created emails and replies and displaying the time of the last auto-save, but in case the connection is lost, pressing the "save now" button ends up in displaying a "Saving…" status message, which never terminates with an error message. This and many other aspects are still not solved or not dealt with properly. Another difference between Rich Internet Applications and desktop applications is the mental model users apply to them. The Web browser serves literally as a frame for the application within, whereas a desktop application is complete in itself. Even if all interaction happens within the web browser and the browser controls are not necessary for operating the application, different mental models apply to these two paradigms (cf. 3.7.1 Mental).

Usability aspects of Rich Internet Applications will be discussed later in the literature review.

A general problem for Rich Internet Applications is the fact that they operate within a technology, which was never meant to be used for application providing. The Hypertext Transfer Protocol (HTTP) is state-less and has to be augmented by session management and cookies to identify particular users; handling multiple windows is difficult: due to abusive

[4] http://mail.google.com/

usage of popup-windows by some web sites, web browsers include mechanisms to confine opening additional windows by web applications (popup-blockers); maintaining the same looks of a web application across different browsers / versions and platforms is a big challenge. Ultimately, the web browser itself suggests a page-based mental model of the information content presented within and its controls are mostly useless when operating a Rich Internet Application. The back-button is not working; the reload button can cause data inconsistencies (i.e. double order), the URL of web applications is often not bookmarkable because it contains session information, which is valid only for one particular client. There is a need for modelling and development techniques for Rich Internet Applications (Preciado, 2005)

To sum up, web browsers are very good at displaying page-based, static content and at the utmost simple form-based interaction (as forms can literally understood as paper forms that one fills out and sends in, receiving a paper receipt in return). Rich Internet Applications however break with the page based model and try to impose an application-based mental model on the user, who is deprived of the browser functionality.

One solution to provide a client-server application in a way a user expects an application to work is by providing interfaces to designated client applications. Some web applications use Eclipse's Rich Client Platform (RCP), which is a Java-based generic client for dynamic client-server applications. For instance *sTeam*[5], the open source platform for collaborative knowledge management and e-learning can be accessed with a variety of clients, including a web interface and a RCP client:

[5] http://www.open-steam.org/

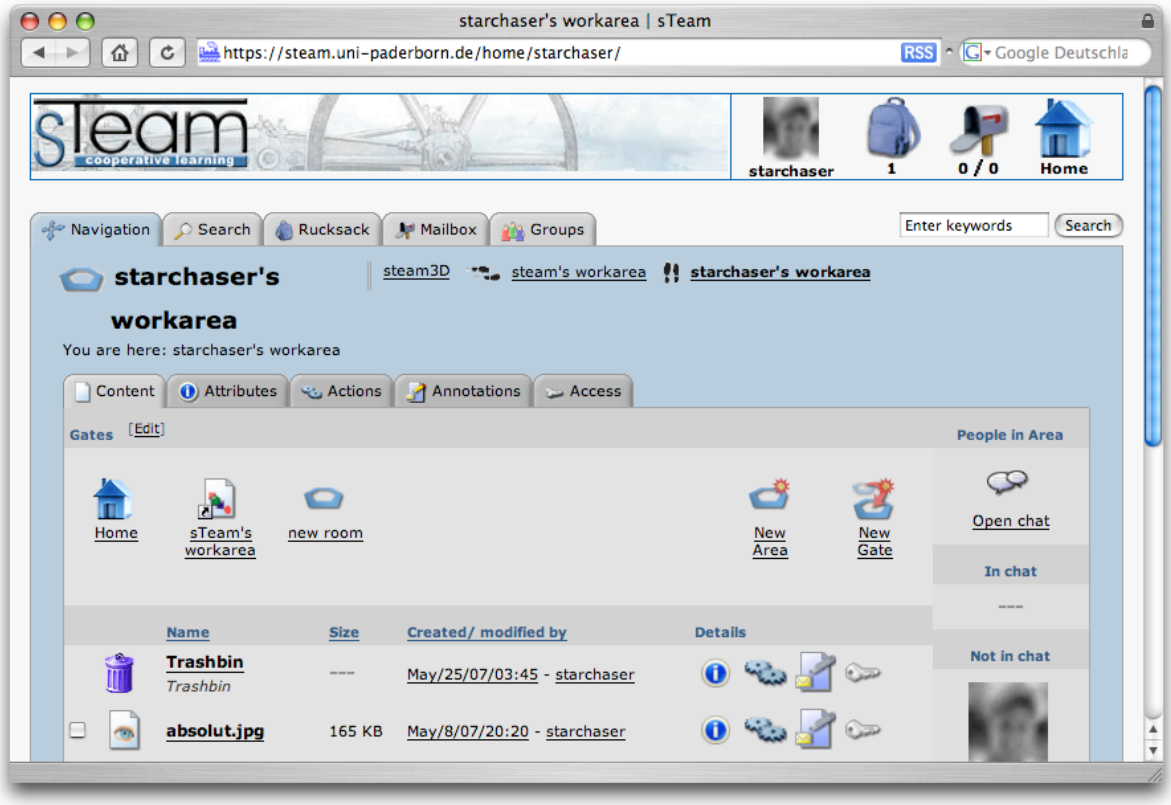

Fig. 2: sTeam web interface

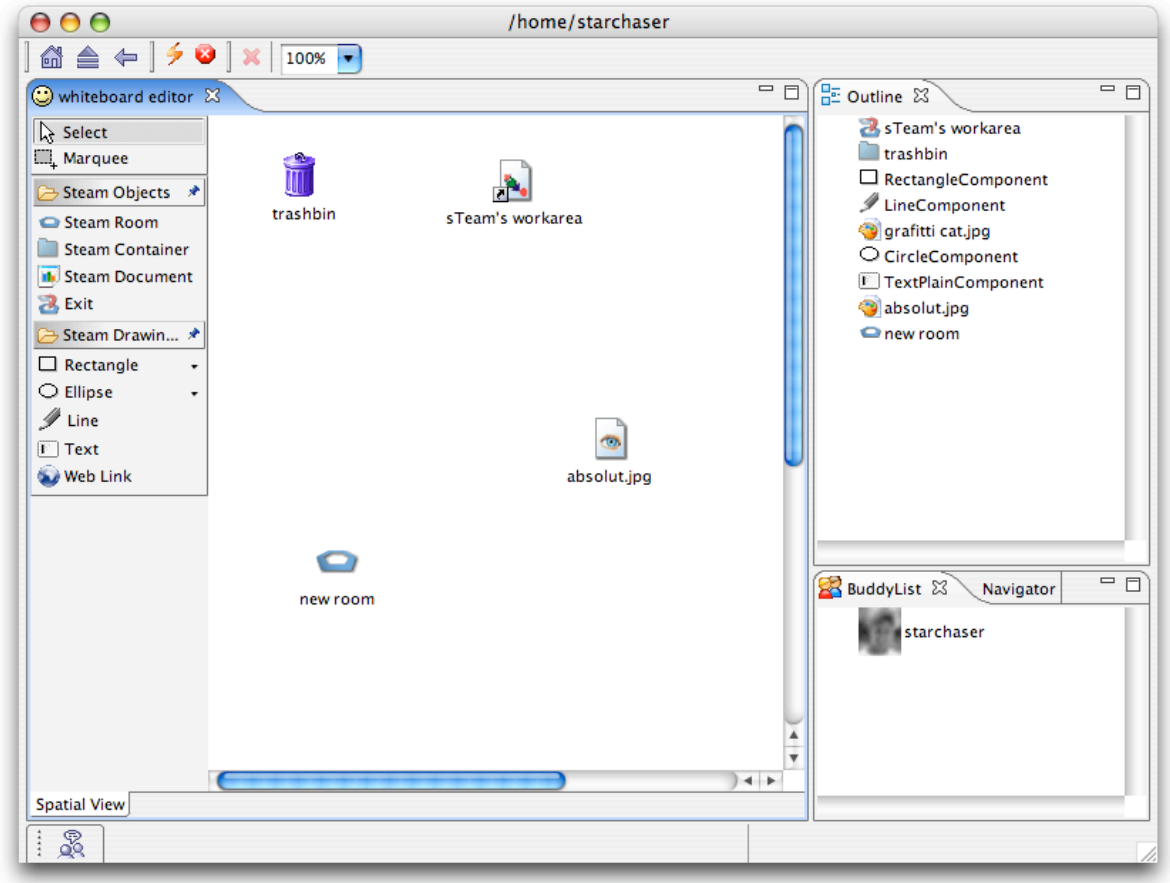

Fig. 3: sTeam on the Rich Client Platform

Other client-server applications provide a proprietary client application, like the FirstClass CSCW platform. The web interface is an almost exact copy of the client application:

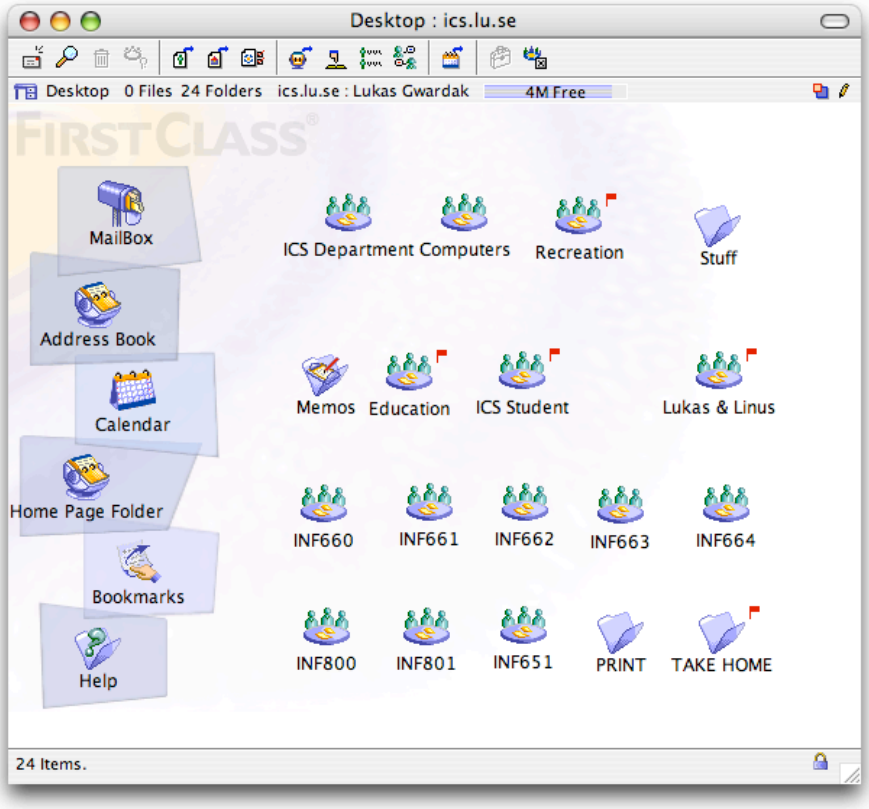

Fig. 4: FirstClass client application

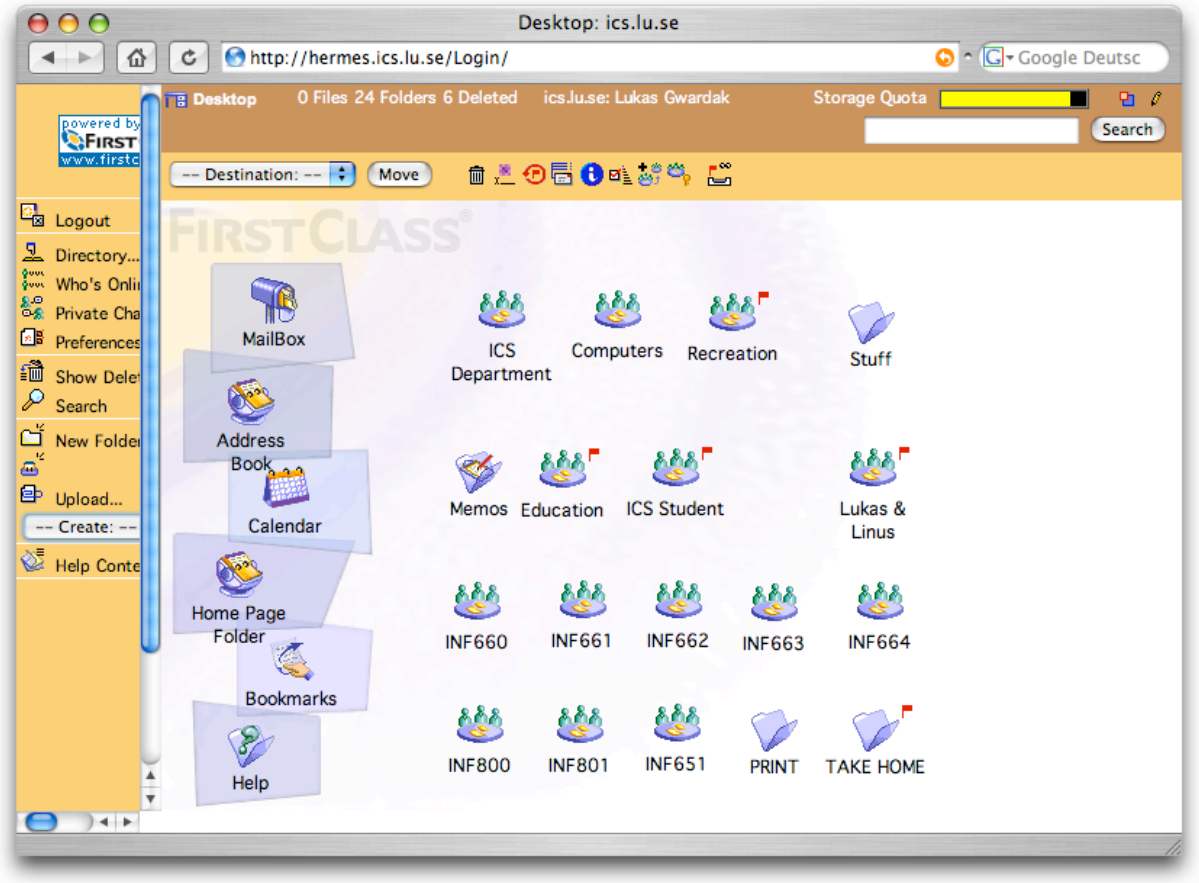

Fig. 5: FirstClass web interface

3.3.1 Web 2.0

The term “Web 2.0” was shaped during the O'Reilly Media 2004 conference by Tim O’Reilly and summarized his article “What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software” (O’Reilly, 2005). It describes recent technological developments and trends in users’ attitudes towards using the web as a social and collaborative medium. Some examples of social trends include the growing popularity of Wikis, Blogs, social bookmarking services and user-generated content in general. This trend is following the initial phase of a more consumer-oriented web and resembles more and more the initial vision of its creator Tim Berners-Lee who envisioned the web browser to be an integrated reading and writing tool. (Berners-Lee, 1999). Wikis and blogs seem to represent the closest approach towards Tim Berners-Lee’s original vision.

Along with new technologies, there are also new information organization paradigms emerging with the idea of user-generated metadata in form of “collaborative tagging”. In contrast to fixed taxonomies to organize information, the users are encouraged to assign freeform keywords (“tags”) to information objects, creating a “folk taxonomy”, or short: folksonomy (Mathes, 2004) which works based on users’ associations (Sinh, 2005) and has been found to be highly effective for information retrieval in the context of content discovery.

On the one hand, user-based metadata creation is a novelty in information organisation; on the other hand, there are no strict rules imposed upon the users about how the metadata is supposed to look like and thus difficult to describe. Folksonomies are “grassroot” information structures, which can be used either for personal information organization or as aggregated source of information by a community of users. Social bookmarking services like del.icio.us (del.icio.us, 2007) or Simpy (Simpy, 2007) use aggregated tags to structure the bookmarks shared by other users and make them searchable by assigned tags. Popular tags are displayed bigger then less popular tags in a so-called “tag cloud” (Fig. 6: Del.icio.us tag cloud), which is a representation of other user’s perception of a topic of resource. This knowledge can be used to discover other resources by clicking on a tag to see other pages tagged with a same tag. This can be also seen as a form of a social recommendation mechanism. The overreaching phenomenon of harnessing collective intelligence has been discussed by Surowiecki, 2004.

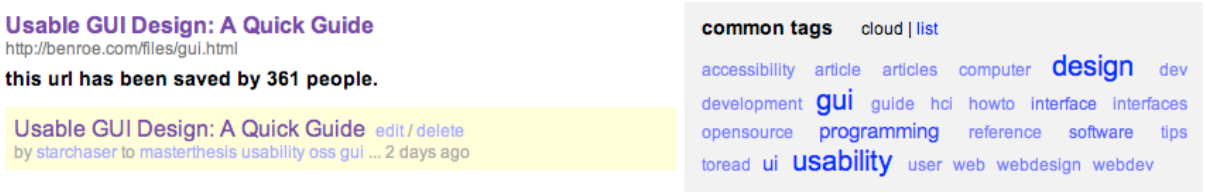

Fig. 6: Del.icio.us tag cloud

Another, more technical phenomenon observed in Web 2.0 applications is the usage of so-called “microformats”, which are simple, XML-based formats that can be easily used by other applications. The most prominent example of a microformat is Really Simple Syndication (RSS). It is normally being used for syndicating news from a website and allows accessing the content with a RSS reader software or integrating it in another website or blog, but it can transport virtually any periodically updated content. Because RSS is an application of XML, it is by nature platform- and device-independent. Other microformats include hCard, an electronic business card format or Friend of A Friend (FOAF, 2007), a machine-readable representation of a social network.

The Web 2.0 is also closely connected to the idea of Rich Internet Applications (cf. chapter 3.3), mostly based on AJAX. Users are expecting slick, responsible interfaces like Google maps (Fig. 7), which allows instant, panning and zooming of the map, switching between different map representations and setting placemarks. The distinctive feature of Google maps compared to traditional map services is the fact that it offers an Application Programming Interface (API) to third party developers. In that way anyone can integrate Google maps into own services or build new applications on top of it.

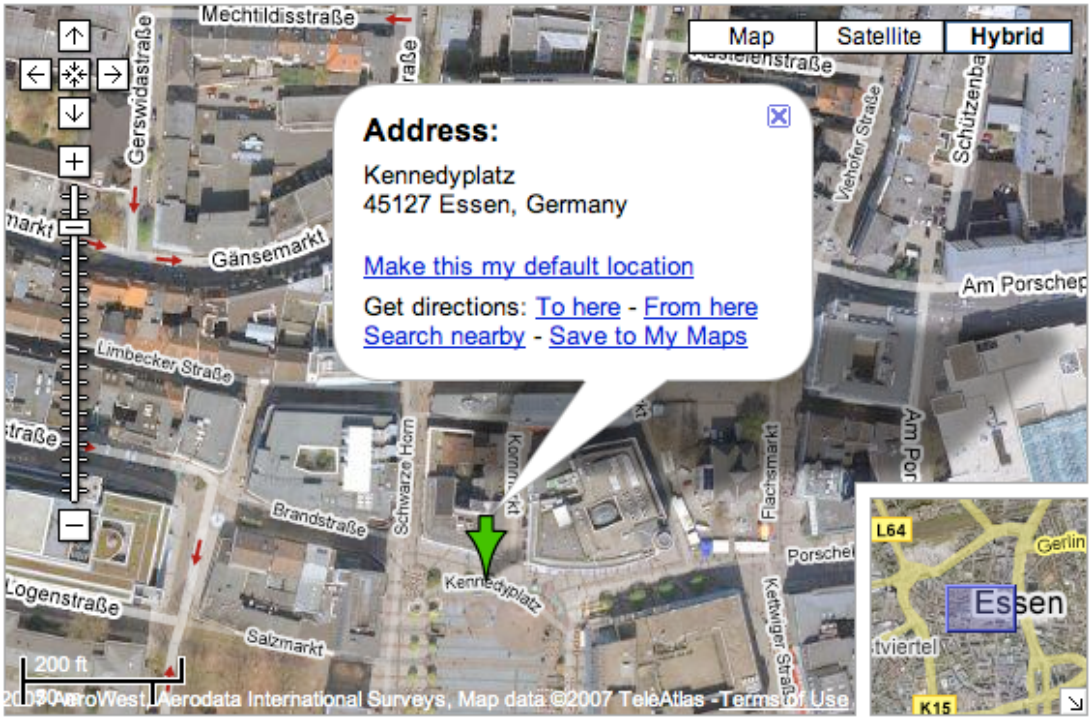

Fig. 7: Google maps interface

Another example of a web 2.0 application is Flickr (Flickr, 2007) - a picture sharing website and an online community for users to upload their pictures, exchange opinions, create interest groups and assign tags to pictures. The various possibilities to use Flickr services outside the actual website make it a good example for a web 2.0 applicaction. One can upload pictures in various ways, either through the website itself or by using plug-ins for popular photo management tools; the pictures can be shown on external sites by integrating an RSS feed or a flash applet showing the most recent pictures. Last but not least, Flickr uses a tagging system to let the users categorize their pictures.

One distinguishing aspect of web 2.0 applications is the fact that they provide interfaces for other web applications. The concept widely known as Service-Oriented Architecture (SOA) in enterprise applications means the ability of web applications to provide interfaces for other applications, resulting in the possibility of creating composite applications, or so-called “mash-ups” (Linthicum, 2007). A possible application could combine an RSS feed from flickr and integrating it with Google maps. The created service could display all pictures tagged with a certain tag with respect to their geographical location. The interesting fact is not the mere possibility to create such a service but the fact that anybody can use the freely available services to create these services. Neither Google nor flickr have to be involved in the process. Websites like Dapper (Dapper, 2007) enable users to create their own mash-ups, even with no programming skills.

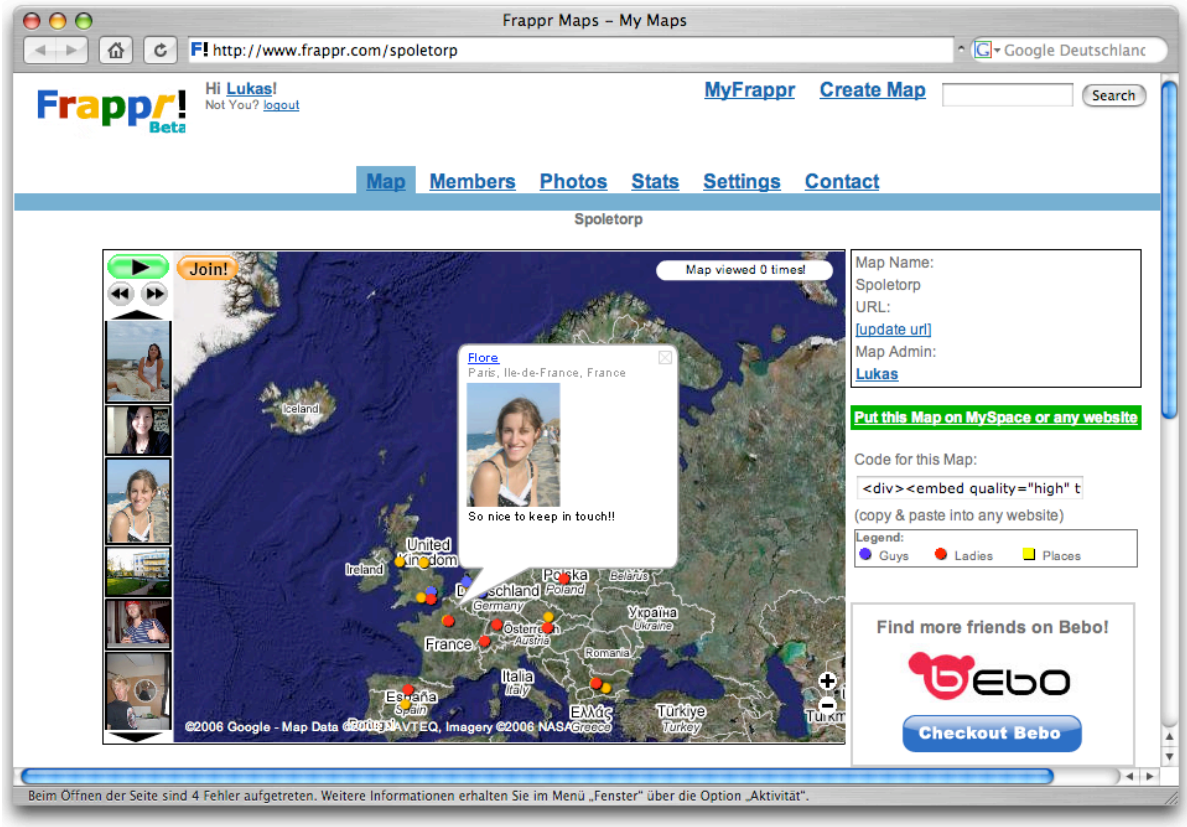

Fig. 8: Frappr! as an example web 2.0 mashup application

From an interaction design perspective, web 2.0 applications constitute a never before seen challenge of designing user interfaces that are universal enough to be integrated in other applications and provide an acceptable level of usability. With applications breaking out of its own boundaries and the original developers not knowing where parts of their applications will be used, it seems like an impossible task to design the "right" interface. Two possible solutions are emerging:

1) Establishing UI guidelines for mash-up applications in the hope that they reach a status of universal design rules and are followed by most developers. The benefit of this solution would be a consistent look & feel of mash-up applications, the downside would be possible lack of innovation, if developers are constrained by always solving problems the same way

2) Separating the functionality of an application from its user interface and give developers the possibility to provide external applications with an own user interface. This would involve more development effort but could result in better consistency and a common look & feel of the resulting mash-up application. The downside of this solution is the loss of control of the service providers over their own applications. For the user, the mash-up application would be indistinguishable from a monolithic

application, which probably would make no benefit for the service providers to offer services for free.

The first solution seems more feasible, as web usability guides seem to have broad influence on how web sites are designed. As an example, one rule states that the main navigation should be placed on top or left side of a web page, because most users expect to find them there. Web pages can look very different, but still, most of them follow this rule. If there were UI guidelines for mash-ups and users would start to develop habits, maybe more developers would develop their services by rules and general best practices would emerge, just like in the example with the site navigation.

Apart from that, there is a long way to go for Web 2.0 applications to become usable, as they just started to emerge and new interaction possibilities are still being explored.

3.4 Web Usability

The term usability refers to the degree of how usable a system is from a user's point of view (Cato, 2001) or more specific, whether the intended audience finds the product easy to use and helpful in completing the goals at hand (Fleming, 1998). Mullet (1995) defines the term as improving approachability and memorability of a product.

Shneiderman (1992) has identified five attributes of usability:

- ➔ *Learnability*: The system should be easy to learn, so the user can quickly get some work done
- ➔ *Efficiency*: Once users have learned the system, they should be able to use it productively.
- ➔ *Memorability*: The system should be easy to remember, so the casual user is able to use the system again without having to re-learn everything.
- ➔ *Errors*: The system should have a low error rate, so the user feels they are making positive progress and are in control, and if they do make errors they should be able to recover from them easily. Catastrophic errors should not occur.
- ➔ *Satisfaction*: The system should be pleasant to use, so users are subjectively satisfied when using it.

The original standard issued by the International Organization for Standardization (ISO) define the term usability as: "A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users" (ISO 9126). As ISO provides technical standards, the definition is made from a technical perspective in form of a non-functional requirement. However, with software products and the Internet becoming

ubiquitous and accessible to a broad population, the impact of these technologies have a subjective component to it, including emotional considerations of aesthetic values und pleasure of use. Therefore, it has been suggested to shift the term “usability” towards a broader perspective of “quality of experience” (McNamara, 2005) or “Universal Design” (Choi, 2006).

Jakob Nielsen (2000) claims that usability has assumed a much greater importance in the Internet economy than it has in the past. This is attributed to the fact that Internet users are more powerful than regular customers: if a customer buys a physical product and is not satisfied with its usability, it is unlikely that he will return the product and get a different one. But if a user looks for information on the web and doesn’t find it on a particular site, he will move on a competitor’s website.

Web usability is a very complex field, as it involves aspects from many different disciplines like graphic design, typography, cognitive psychology, interaction design and many more. However, because nobody has complete knowledge about all these sciences, there are usability guidelines (cf. chapter 3.6.3) that summarize general findings about how users are using the web and which practices exist to solve a particular problem.

One of the most crucial usability challenges in the web is navigation. Fleming (1998) states that users perceive the web as a space and thus they need answers to four basic navigation questions:

- ➔ Where am I?
- ➔ Where can I go?
- ➔ How will I get there?
- ➔ How can I go back to where I once was?

It is important to know the target group one is addressing with a web site. Fleming (1998) addresses this topic by providing guidelines for various types of sites, like shopping sites, community sites, entertainment sites, etc. Nielsen (2000) deals with it by discussing best practices of various facets of web design, like page design, content design, site design, intranet design, accessibility and internationalization and focussing on concrete solutions for most common problems.

Nielsen (1994) also established a set of ten usability heuristics for UI design, which were originally intended for desktop applications but have been later on found to be universally applicable to web usability. McMullin (2003) discusses how these heuristics can be applied to Rich Internet Applications and provides suggestions how to incorporate them in Flash-based applications.

Usability and accessibility are strongly interrelated domains. Accessibility has been identified as a key characteristic of web applications that affect usability (Bruno, 2005). Usable sites are not necessarily accessible to people with disabilities, but accessible sites are most likely more usable for people without disabilities (Theofanos, 2003). Therefore, designing accessible web sites is a double-gain because it creates equal access to information for all users and improves the overall usability. Sullivan (2000) undertook a quantitative comparison of the relationship between accessibility and usability, which led to the suggestion that there may be in fact a fundamental correlation between the two. A newer study by Petrie (2007) gathered empirical data from a study with 6 sighted and 6 blind participants, which has shown that the severity of usability problems goes in pair with accessibility problems.

Shneiderman (2000) has taken a more holistic approach and shaped the term “Universal Usability”, referring to the ACM Code of Ethics (ACM, 1997), which formulates universal access to information as one of its objectives. According to this code, computer resources should be equally available to everyone, regardless of race, sex, religion, disability or nationality. This code includes accessibility, but at the same time touches upon fundamental principles like net neutrality (in relation to censorship of information), tolerance and usability. Universal usability is therefore more than pure “user friendliness” but rather the concept of all users having equal opportunities in accessing information.

It is difficult measure the degree of usability of a computer system. One attempt to model the interaction involving the cognitive processes of a user was the GOMS (Goals, Operators, Methods, Selection rules) model (Card et al., 1983): It consists of four elements:

- ➔ *Goals***:** users’ intended actions
- ➔ *Operators*: actions performed in order to achieve the goal
- ➔ *Methods*: sequence of operations necessary to achieve the goal
- ➔ *Selection rules*: specify which method to use when there are more to choose from.

The GOMS model’s advantage is the possibility of predicting the efficiency of usage of a system without having to involve users for costly measurements, which is again a disadvantage, because it doesn’t account for errors or users’ fatigue while performing repetitive tasks (Preece et al, 2007)

Another metric (Babiker, 1991) deals specifically with hypertext usability and is based on three attributes, which are considered to be important in any hypertext system: access and navigation, orientation and user interaction. Empirical testing of the metric has shown that the computed values approximate the usability ratings closely, suggesting that an effective metric can be developed, once the right weights of the attributes have been determined.

3.5 Accessibility

3.5.1 Computer Accessibility

There are many kinds of disabilities, which developers of interactive systems should account for. There are distinctions between physical, sensory and cognitive impairments. Physical impairments include motor disabilities, which make it difficult to operate the keyboard and mouse, like in the case of Paraplegia or Parkinson's disease, but can also occur in a weaker form for elderly people. Sensory disabilities include hearing and sight impairments, which result in difficulty or inability to read the contents of a computer screen, recognizing colors or hearing audio information. Lastly, there are cognitive disabilities like learning disability, Dyslexia, etc.

Assistive technologies aim to make information accessible to people with disabilities by transforming them into other representations. Visually impaired people can use screen reader software and a speech synthesizer to perceive the information; other possibility is to use a Braille-terminal, which can mechanically transform text to Braille characters. Hearing disabilities have most impact with audio/visual content and can be accounted for by including captioned text in the content.

The probably most challenging task is to handle cognitive disabilities, because they lie at the very source of understanding information. It would be with no doubt very difficult to design an information resource for illiterate people, but is possible to help people with spelling problems (Dyslexia) to find the right resources by suggesting them the correct spelling.

It is a moral and ethical responsibility to design information systems for maximum accessibility to enable equal access to information for all people, regardless their physical predispositions. As the web is a very important information resource, it is particularly important to design web content in an accessible way.

3.5.2 Web Accessibility

The Web Accessibility Initiative (WAI) driven by the World Wide Web Consortium (W3C) is aiming towards improving the accessibility of the Web for visually impaired or otherwise disabled people using screen reader software. As screen reader software only can interpret the text portion of a web page, it is of great importance that the web page is written in semantically correct HTML, making proper use of headings, paragraphs, tables, links, etc. Images should contain an `ALT="…"` attribute, which describes the content of the image.

There are guidelines available for designing accessible web sites, like the Web Content Accessibility Guidelines (WCAG) (Chrisholm, 2001), and there are local regulations prescribing that government and communal web sites have to be accessible, like the Section 508 of the Rehabilitation Act for the U.S. or the Behindertengleichstellungsgesetz (bill of equal access for people with disabilities) in Germany. The background of enforcing accessibility on the web is to create equal access to information, regardless of physical predispositions, age or background.

3.5.3 Problems With Web Accessibility

Designing accessible web sites is not only a technical, but also an awareness challenge. Most web developers simply don't think of users with special needs. Web Content Accessibility Guidelines don't seem to be difficult to implement, as many rules relate to simply giving proper names of headings and link descriptions. Various studies have uncovered that even web pages designed according to WCAG still have accessibility issues (Colwell, 1999, Theofanos, 2003). One reason for these results could be the fact that many designers don't know how to work with accessibility guidelines (Law, 2006).

One difficult aspect of web accessibility is interactive content. Screen reader software supports simple form-based data input, but JavaScript and DHTML remain unsupported and thus interactive web applications will be inaccessible to users with disabilities. Although attempts have been undertaken to solve the problem by assigning shortcuts keys to JavaScript-enabled page elements (Gibson, 2005) or other work-arounds (Lemon, 2006), this solution would have to be implemented on a wide scale and kept consistent over all web applications. Reality however shows that different web applications are very heterogeneous in regards to the frameworks and approaches used. Taken into account the fact, that nearly all Rich Internet Applications rely on using JavaScript for dynamic interaction, these types of applications will remain inaccessible to screen readers as long as they provide no support for JavaScript. Also, given the fact that graphical user interfaces for Rich Internet Applications were designed for visual access, makes the attempt of using them with a screen reader not feasible. A promising solution is being worked on by the World Wide Web Consortium (W3C): in their current working draft for Accessible Rich Internet Applications (W3C, 2006) they propose extending the XHTML standard by several accessibility features, which could enhance support by screen readers, given the premise that web developers will follow the proposed extensions and screen reader developers integrate extensions in their products (Kliehm, 2007).

Until this happens, the only feasible workaround seems to be creating two separate versions of an application: one for standard web browsers and one for screen readers, which reduce the interaction to simple forms. The online store *Amazon.com* is providing a good example of such an approach. This involves additional development effort, but serves its purpose.

In addition to the accessibility problem itself, designers apparently have difficulties working with accessibility guidelines, which was found to be caused by knowledge gaps in developers' perception of the conceptualized system of designers, end-users and guideline-setting committees (Law, 2006). A field study revealed that developers had trouble implementing Web Content Accessibility (WCA) guidelines and even pages developed according to these guidelines had accessibility problems to visually impaired users using screen readers (Colwell, 1999).

Recapitulating, there seems to be a fundamental conflict between web accessibility and Rich Internet Applications in their current form. Existing web standards have to be extended and embraced by developers in order for the situation to change.

3.6 Guidelines

Guidelines in the context of this work are collections of principles, conventions or directives, put together into a single document to be used by developers of products or services. Depending on the level of specificity, they can be illustrated by examples. They should provide the developers with sufficient guidance to help them with making the right design decisions without having to consult designers or usability experts. They are in fact a means of communication between designers and developers.

The benefits of using guidelines in HCI have been recognized by many researchers (Ianella 1995, Rosenzweig 1996, Scapin 2000). Using them can ensure consistency among products and services, which provides a better user experience.

There are however also some shortcomings to guidelines. Scapin (2000) focuses specifically on web usability guidelines when he identifies the following five weak points:

1) The level of guideline expressiveness and the confidence in applying guidelines heavily depends on the source where the guidelines come from.
2) As a consequence, guidelines require interpretation for the intended context of use.
3) The jargon used in guidelines may prevent designers to easily understand and apply guidelines correctly

4) The ease of interpretation of guidelines depends on their classification within the linguistic level of interaction model (goal, pragmatic, syntactic, lexical, alphabetical, physical).

5) The workload involved by using a guideline depends on several guideline properties, like linguistic level, quality of statement and scope.

Rosenzweig (1996) mentions that using guidelines shouldn't replace thoughtful design, prototyping and user testing, but it can improve the starting position for these activities if the initial design is done "almost right" by using guidelines.

3.6.1 Classification of Guidelines

Usability guidelines come from different sources; they are being created for various purposes, target audiences and application domains. During our research we encountered various guidelines and we categorized them roughly into several types (Table 1). We developed the categorization originally for internal organization of the guideline documents we found, but we think it is also useful as an overview over the various field of usability guidelines as a whole. This is a high-level categorization. Web design guidelines have been already categorized and the scheme is presented in chapter 3.6.3 Web Guidelines.

Table 1: Classification of Usability Guidelines

Category	**Distinguishing features**	**Examples**
Corporate identity guidelines, Style guides	Focus on colours, graphical layout, typefaces, Look & Feel	- Oracle Browser Look & Feel UI Guidelines - Lund University Visual Identity Programme
Desktop UI guidelines	Focus on a specific operating system or desktop UI	- Apple Human Interface Guidelines - The Windows Interface Guidelines for Software Design - Gnome Human Interface Guidelines - Indigo Magic™ User Interface Guidelines - KDE 3 Styleguide
Accessibility	Focus on accessibility	Web Content Accessibility

guidelines		Guidelines (WCAG)
Web Usability guidelines (collections)	Focus on web pages / web applications, universal rules	- Designing Web Usability (Nielsen, 2000), - Web Style Guide (Lynch, 2002), - Research-Based Web Design & Usability Guidelines (National Cancer Institute, 2002)
Product experience guidelines	Focus on a particular product and its context of use	- One Laptop per Child guidelines, - DVD menu guidelines - Nokia S60 Platform Visualization and graphic design guidelines - Usability Principles for CMS products (Robertson, 2001)

One could also distinguish between principles, heuristics, specifications and conventions, but as they can all be incorporated in high-level guidelines, we will refer to them as guidelines further on. We understand all documents that can be used as a helpful advice in developing interactive systems as guidelines.

3.6.2 Desktop User Interface Guidelines

Desktop user interface guidelines (Apple 1992, Microsoft 1995, Sun 1989, Benson, 2004, SCO 1996, **Silicon Graphics, Inc. 2001**) are one of the first types of usability guidelines published and they typically cover usability issues of the user interface for a specific computer platform, operating system or desktop interface. They describe in detail how windows, dialog boxes, widgets and general conventions are envisioned and how to develop software to stay consistent with these principles.

In general, the guidelines begin with high-level principles and general metaphors used by the system in question. Terms like WYSIWYG (What-You-See-Is-What-You-Get), feedback, forgiveness etc. are explained

The general focus of desktop user interface guidelines lies on the elements of the graphical user interface elements (menus, windows, dialog boxes, controls, widgets) and their behaviour. Very little stress is put on multi-step interaction processes, content structure and content navigation. Remarkably, internationalization receives very little attention in the guidelines.

3.6.3 Web Guidelines

Web usability guidelines can exist in a variety of forms and levels of specificity. Scapin (2000) has identified five basic categories of web usability guidelines:

- ➔ *Design rules:* Functional and/or operational specifications that clarify the design of a particular user interface.
- ➔ *Ergonomic algorithms*: Aggregation of single design rules into a comprehensive and systematic procedure that can be applied more quickly than a series of single guidelines.
- ➔ *Style guides*: A set of guidelines and/or functional or non-functional specifications aiming at consistency for a collection of distinct user interfaces.
- ➔ *Compilations of guidelines*: Several prescriptions written for a wide range of user interfaces. Each prescription is presented as a statement, sometimes along with examples, with or without clarifying explanations and comments.
- ➔ *Standards*: A set of functional and/or operational specifications intended to standardize design. Standards are promulgated by national or international organizations for standardization.

Scapin (2000) also positions the different categories on a scale of specificity and need of interpretation (Fig. 9), with Standards at the top of the interpretation scale, meaning they are general and require much interpretation by web developers before they can be applied; design rules on the other hand are specific and require little to no interpretation with the drawback of possible lack of flexibility and transferability to other problem domains. Also the jargon used in the guidelines can be attributed to different disciplines (i.e. cognitive psychology, graphic design, ethnography) can be difficult to understand by the web designers and experience is needed to apply the guidelines properly.

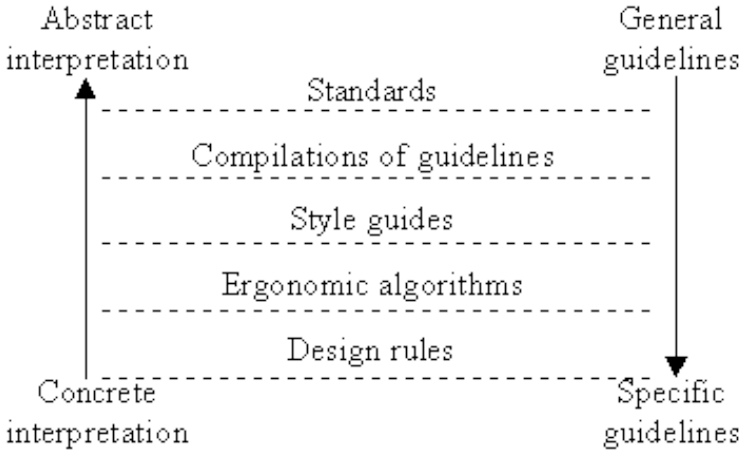

Fig. 9: Level of guideline expressiveness (Scapin, 2000)

Marriage (2004) presents a matrix, which relates the different guideline types to the level of implementation precision and need for interpretation (Fig. 10).

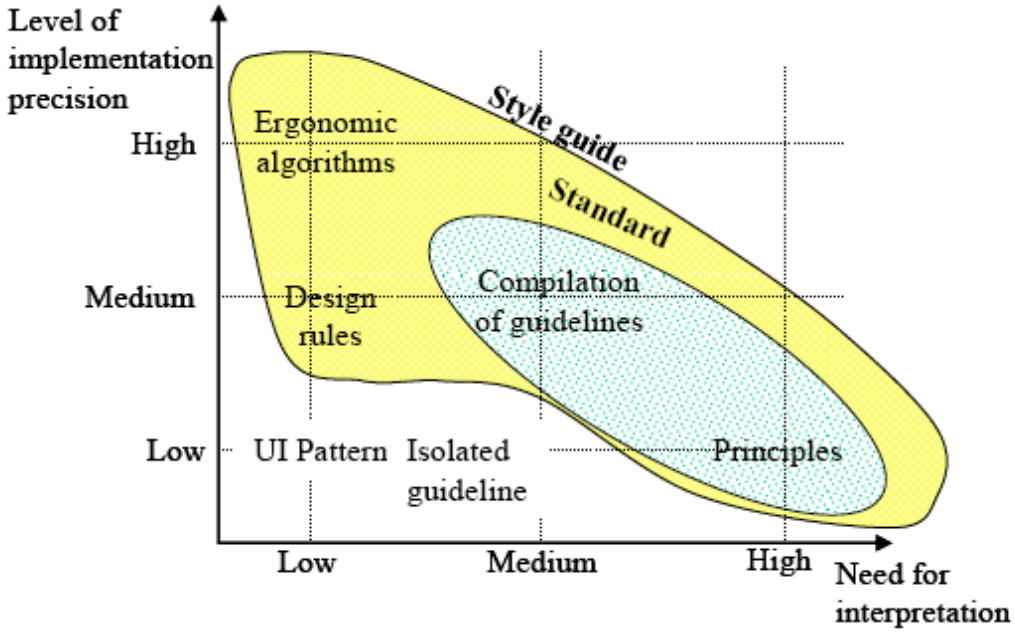

Fig. 10: How to select appropriate guidelines (Marriage, 2004)

Ohnemus (1997) divides web guidelines into three basic categories with the same scale of specificity: principles having a general and conventions a specific character.

- ➔ *Principles*: The goals, which guide design decisions.
- ➔ *Guidelines*: Middle layer, based on principles specific to a particular domain of design.
- ➔ *Conventions*: Dictating specific decisions one has chosen and reflecting the needs and terminology of the organization.

It has been found that web style guidelines emphasize common look & feel, information display and navigation issues (Ratner, 1996). Other topics found are site/page structure, graphical layout and typography.

Other approaches to structure guidelines is not the level of abstraction, but the type of the website and its intended target audience. Fleming (1998) presents guidelines for different types of websites, including:

- ➔ Shopping sites
- ➔ Community sites
- ➔ Entertainment sites
- ➔ Identity sites
- ➔ Learning sites
- ➔ Information sites

The advantage of this approach is a holistic view on the intended user experience, including fundamentals like mental models (i.e. shopping cart or checkout process on a shopping site) and user goals, presenting case studies of successful sites and giving references to further literature dealing with these types of sites. The disadvantage is obviously the question of transferability of these guidelines to other, not mentioned or not even invented types of sites. In that way, this approach is both specific in regards to site types, requiring little interpretation, but still holistic enough for the reader to conclude general rules.

A similar approach towards creating guidelines is done by Schmeiser (1997), which focuses on graphical layout of different types of web pages (not web sites). Examples include: search pages, surveys, articles, annual reports, etc. The web developer can than choose among several alternatives and compose a complete web site from the given templates. This visual approach however lacks underlying principles and presents only ready solutions. If a new type of page (i.e. calendar page) will appear in the future, the knowledge will be not transferable.

An early approach towards designing hypermedia spaces (not only the web, but also multimedia products like help systems, references, encyclopaedias, etc.) was presented by Isakowitz (1995). The proposed Relationship Management Methodology (RMM) is an approach rooted in software engineering and aims at developers and project managers with previous knowledge in this area. Back in 1995 the web was not widely available and many aspects like usability or accessibility were not considered yet.

3.6.4 Differences Between Web Guidelines And Desktop Guidelines

It seems obvious that high-level principles of interactive systems should be very similar. Nielsen's (1994) usability heuristics, which were originally developed for desktop applications apply very well to web applications, mostly because they are formulated on a high level; the same seems to be true for Tognazzini's (2003) First Principles of Interaction Design.

However, when Desktop guidelines are compared with guidelines made specifically for the web, there overlapping has found to be only approx. 20% (Ratner, 1996). It depends of course on the type of guidelines (styleguides, specifications, etc.) and their speficity. High-level principles are the same for any interactive system, but each medium has its own affordances and characteristics, which require different approaches towards designing the interactive space. Nielsen (1997) discusses the differences between web design and GUI design and points out that web design is a device-independent, user-controlled space, which is being used as a part of a whole, in contrast to desktop GUI being developed with a WYSIWYG paradigm and as closed entities. Wroblewski (2001) states that web usability guidelines don't address issues relevant to web applications and advocates enhanced design guidelines to close this gap.

The affordances of the web are connected with the users' mental model (cf. chapter 3.7.1) and the technological limitations of the web (client-server applications in general have a lower response latency, rich interaction is not possible with HTML/JavaScript, slower I/O compared to internal hard drives, etc.). However while technological limitations may change in the future, the mental model of users browsing web pages with a web browser is still fixed. Web guidelines are reflecting this model of a page-based interaction and put very low importance to interactivity and process flows between web pages. Instead they mainly stress upon information presentation, content structuring and navigation, which lies at the very foundation of the web medium. Desktop guidelines on the other hand focus very much on the interaction with windows, menus, icons and other GUI elements, mostly on a widget-level. Content presentation and structuring is not covered by Desktop guidelines, because desktop applications typically have very diverse purposes, of which content presentation is just one among many. However, the web's only purpose (until web applications evolved) was to present content and this is exactly what web guidelines are aiming at.

From the designer's perspective, web- and desktop usability guidelines are targeted at different audiences. Mariage (2004) presents a comparison of web guidelines and desktop guidelines in regards to the context of web development vs. Desktop application development (Table 2).

Table 2: Differences between interfaces: Web and GUI (Mariage, 2004)

Designer / Developer	**Web**	**GUI**
Who	Professionals and non-professionals (almost everyone can design a web page)	Professionals
Nature	Interface oriented towards navigation in contents	Interface oriented towards functionality and application domain
Technology	Low risk in deployment, user testing	Moderate risk in deployment, software testing
Disciplines	Information architecture, human factors, graphics, marketing, …	Information Technology and application domain specialists
Usability	Depending on the profile of designers/developers Usability may be hard to control as web navigators and user populations vary	Depending on the development process followed Usability built in the software and no UI variation
Interactivity	Ranging from almost non-interactive (contents viewing) to highly interactive (depending on technology used)	Generally highly interactive Potentially with immediate feed-back and direct manipulation
Life cycle of application	Fast development, short life time, risk to disappear quickly	Moderately long development, long life time, stay stable for a while
User	No license needed and no installa-tion. Hence, sites are competitive and zapping is frequent	License and installation are re-quired. Software is moderately competitive.

Content responsible	Content is updated regularly	New versions are produced from time to time

Ratner (1996) has done a qualitative comparison of recommendation types found in Desktop guidelines and web guidelines. The distribution shown in Fig. 11 illustrates the findings.

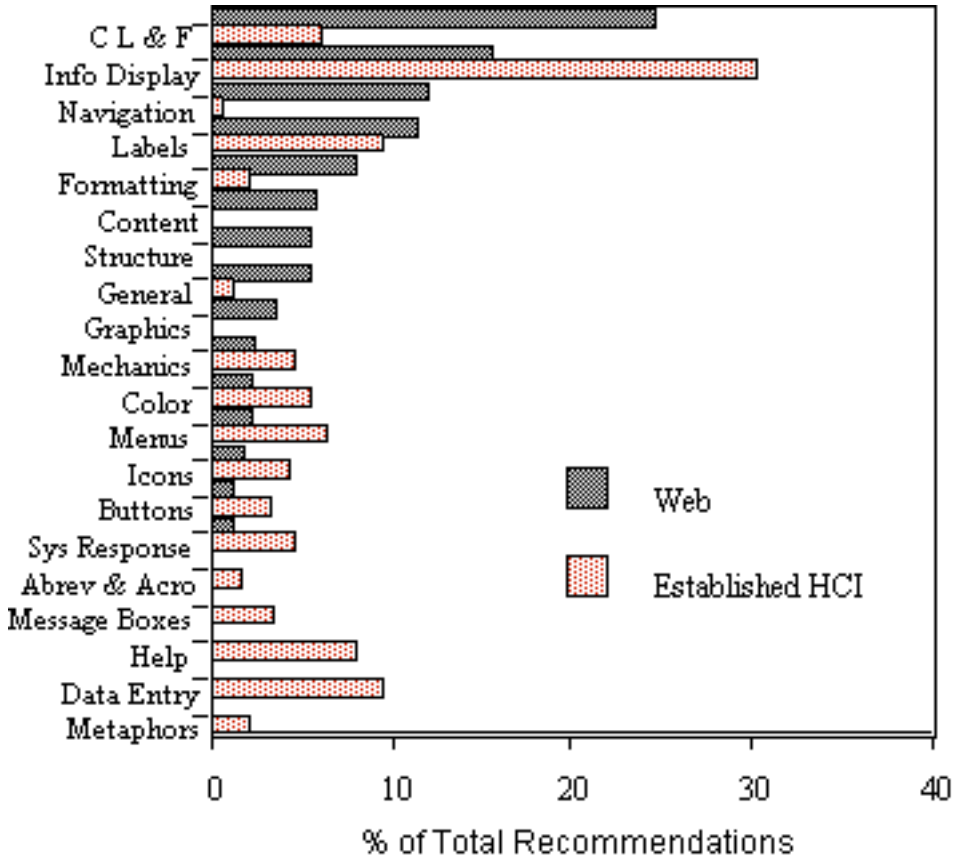

Fig. 11: Distribution of style guide recommendations across categories (Ratner, 1996)

Given the fact that interactive features receive low to no coverage in web guidelines, it becomes obvious that usability problems of Rich Internet Applications cannot be solved with current web guidelines and there is a need for a revised set of guidelines which take these issues into account.

3.6.5 Other Types of Guidelines

Usability guidelines are not limited to graphical user interfaces of computers only. There are guidelines existing for mobile phones, which encompass not only the special affordances of a

small mobile phone screen, but also the environment in which it is embedded, the whole context of use and the resulting high-level principles.

Another example are guidelines for DVD on-screen menus (Kappel, 2006) which take into account the fact that DVD users don't necessarily have previous computer experience, and the graphical design is a much more important part than in traditional user interfaces, as it has to integrate with the content of the DVD. Still, the menus should be understandable and accessible to users with disabilities. The input device is a remote control and as many users don't use special keys for language selection etc. the DVD menus have to be accessible only by using a 4-way remote control.

Yet another type of guidelines are product experience guidelines, where not only the user interface, but the whole product philosophy is an integral part of the guidelines. As an example, the One Laptop Per Child Project (OLPC, 2007) develops a low-cost laptop computer for young children in developing countries. The uniqueness of the target group, which has no previous experience with computers (or electronic devices in general) and the core idea of collaboration, expression and learning makes the guidelines very different from regular "user interface" guidelines.

The lesson learned from other types of usability guidelines is, that the interface has to be seen more broadly, not just as another "GUI", but as a combination of hardware and software, which is embedded in a specific context of use and used by a specific user group.

3.6.6 Developing Guidelines

As there are big differences in types and application domains of guidelines, there exist many approaches towards developing them, depending on scope, needs, available resources and purpose of the guidelines.

One development strategy for generic web guidelines is to perform a heuristic evaluation of websites. Web sites found to have good results in the evaluation are analyzed closer to find the distinguishing features that can be transformed to a guideline. This method has been shown to produce good results, as web sites created with help of these guidelines provided a better usability (Borges, 1996).

Another method to create guidelines is to combine a literature study, expert walkthrough and user testing (Kappel, 2006). The literature study gives a basic foundation on the topic, helping to find the development direction; the first set of guidelines is then refined by experts and finally tested on users. The results from user testing were finally used to improve the guidelines.

In an open source environment, like the GNOME Linux desktop project (Benson, 2004) guidelines are created de-centrally in a collaborative effort of many voluntary experts and participants, which consent to user testing. Also usability focus groups, heuristic evaluations and ethnography in workplaces have been performed to improve the guidelines. Typically for the open source environment, the communication takes place electronically, by mailing lists, Internet Relay Chat (IRC) and a wiki system. The guideline document itself is in constant change as user testing is progressing and new versions of the GNOME interface are being released.

If guidelines are being developed within an organization and the guidelines are tailored to the organization's needs, then they are often developed by a cross-disciplinary team, which represents the organization (Rosenzweig, 1996). The main challenge is to gain commitment from all participating departments and working groups, each with its own agenda. This can be achieved either by including everybody in the process and making the importance of the undertaking explicit, or by support on management/department level. If the departments understand that they can in fact save time when the guidelines are established, they will perceive the development process not as a burden, but as a useful investment in future development processes. Guidelines are particularly useful for organizations, as they help to create a consistent user experience of a product family and create a distinct Look & Feel of the company's products. Customers may prefer buying products from the same company if they can transfer previous knowledge they had with products from the same company (Rosenzweig, 1996).

All in all, there are not many methodologies established for developing guidelines; Beier (2003) presents the probably best-known methodology to develop UI guidelines for web applications. The particular challenge was to design for multiple web-based applications across a variety of different user profiles. The guidelines' goal was to design accessible, cross-browser compatible and localizable applications in the scope of the whole enterprise. The presented solution was the *Bull's-Eye framework for Web Application UI Design*, consisting of several levels of guidelines, ranging from specific components to page templates, page flows, patterns and overreaching features and principles (cf. Fig. 12). Combining guidelines in that way made them work better together, because they addressed different levels of interaction of a web application.

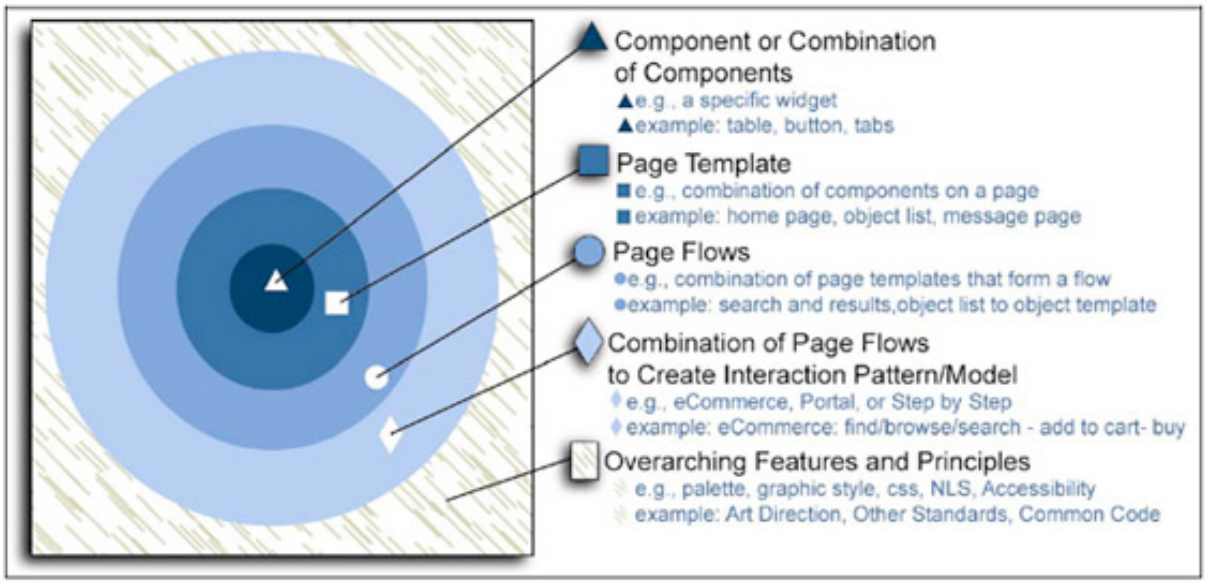

Fig. 12: The Bull's-Eye: A framework for Web Application UI Design Guidelines

Other organization-based approaches focus less on the inter-relation of guidelines, but on acquiring them based on experiences (Henninger, 1995) or embedding them in an organizational context (Henninger, 1997).

A methodology for creating guidelines for evaluating software products was presented by Cordes (1981) and it consist of several tasks:

1. Decide on a definable subject matter or goal for the guideline
2. Collect guideline material
3. Form a checklist of "point-ables" based on observable attributes of software
4. Collect and categorize products which typify the "best" the "worst" of your subject
5. Test both groups with your checklist
6. Weeding – only use those items which significantly reveal the difference between the two groups
7. Divide the remaining checklist items into meaningful component dimensions
8. Test and collect data on products
9. Establish normative scaled scores for each dimension (e.g. T-scores)
10. Develop weights for the dimensions along with a cut-off which will allow classification of future products (e,g, perform discriminant analysis)
11. Constantly collect more data and revalidate

Even though this methodology is not a strict usability-related, it shows that creating guidelines involves the essential steps of collecting suitable guideline material, categorizing it and testing. After finishing the guidelines have to be constantly revalidated to remain up-to-date.

Tools can greatly improve the process of guideline collection, as mentioned by Vanderdonckt (1999), who describes the development process of a tool for collecting and organizing guidelines, which is a methodology in itself, because guidelines have to be acquired before they can be managed. Similar solutions have been presented earlier by Ianella (1995) with *HyperSAM – A Management Tool for Large User Interface Guideline Sets* and Cohen (1995) who compares different guideline management tools. Regarding the broad variety of available tools for guideline management, we assume that they play an important part in the guideline creation process.

3.7 Cognitive Perspective

3.7.1 Mental Models

Mental (or conceptual) models are imaginary “scale models” of systems that build up in users’ minds when they are using these systems (Norman, 1988; Gedenryd, 1998). Remarkably, the mental model doesn’t have to match the actual technical realization of a system. If the user doesn’t know how a system works internally, she creates an own model, that reflects the perceived way of how it works. As an example, most people don’t know that an electric stove doesn’t heat up faster if the dial is turned fully up. Their mental model matches a valve-like behaviour, not the (correct) thermostat-behaviour (Norman, 1988).

Norman (1988) distinguishes three aspects of conceptual models: the design model, the user’s model and the system image, with the design model representing the designers conceptualization of a system, the user’s model representing the user’s perception of the system’s inner workings. The ultimate goal is to design a system in a way that the envisioned design model is equal with the perceived user’s model having in mind that often the system image is the only way to communicate the model between the designer and the user.

One can distinguish between the designer’s model (how the designer envisions the system) and the user’s model (how the user perceives the system). The resulting system image is a combination of both: the model envisioned by the designer has to be communicated to the user (Fig. 13).

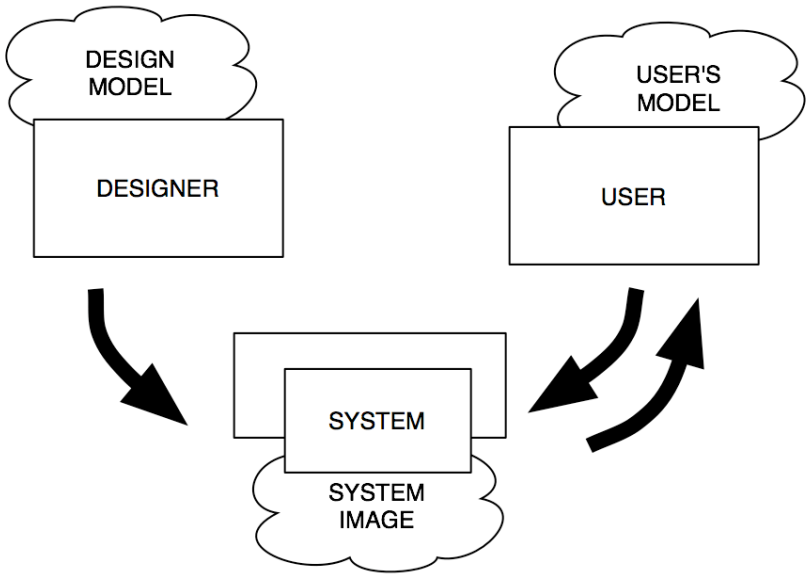

Fig. 13: The Design Model (Norman, 1998)

Eysneck (1995) explains that mental representations can be viewed from two main perspectives: as symbolic and as sub-symbolic representations. Symbolic representations are normally stored in the long-term memory. Sub-symbolic representations are "distributed representations" stored as patterns of activation in connectionists networks. Further on, there is a distinguishment between internal and external representation (Fig. 14).

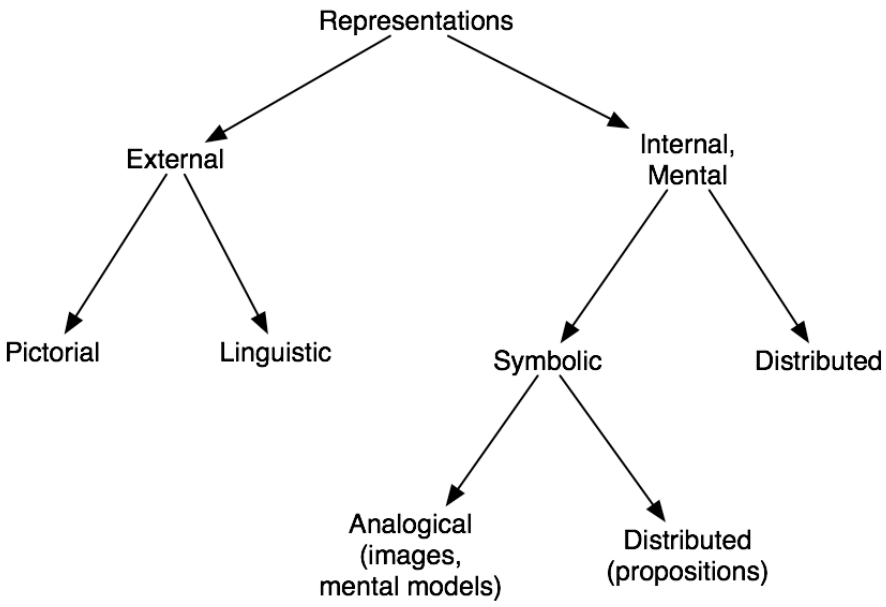

Fig. 14: Different types of mental representations (Eysneck, 1995)

According to this model, external representations are tied to pictures, or verbal descriptions, whereas internal representations are based on symbols and/or distributed features. Hence, the web is represented as a symbolic mental model in users' minds. It is something that emerges in the mind, without a verbal or pictorial description. Studies with web pages designed according to users' mental models have shown that their usability can increase (Dalal, 2000).

There have been approaches made towards cognitive modelling to use especially for user evaluations. To do this, a model of cognition has been developed and matched with the behaviour of human mind, but this approach has limited possibilities and can act only in a narrow scope. Higher-level features of human cognition like learning and abstract thinking were not accounted (Caroll, 2001; ch.6).

Mahlke (2005) furthermore expands the understanding of users' experience of usable interactive systems by adding subjective aspects as aesthetics and emotions to the established factors.

4 Case Studies

4.1 Introduction to the Case Studies

During our search for suitable companies to include in the case studies we were guided by the need of finding practitioners and experts in the field of usability and web development. Because we have only interviewed three developers, it was very important for us to get a broad spectrum of competencies and backgrounds of our interviewees. Previous work relationships and contacts helped us to find the right people for the interviews. The companies in our focus are dealing primarily with developing web design and web applications, but involve different roles in the development process. We have interviewed a user experience guidelines designer, a user interface designer and a web developer – altogether these roles cover almost all of the development process and provide a complete picture of the involved processes. At the same time, looking into three different companies showed us different approaches towards solving problems, which are based on different corporate cultures, working techniques and environments. *Mindlab* grew out from a traditional German software company with its processes not yet fully adapted to the workings of the web. In contrast, *Netvibes* is a young web 2.0 start-up with a closely working team, which understands the importance of involving its users in the development process of their product. *sTeam in turn* is a research project that follows the rules of academia and open source participation. The development is not profit-oriented and there is much more space for experimentation and new interaction ways to be explored.

4.2 Finding Interview Questions

Following our research question, our goal was to find out how developers work with guidelines and how they envision possible guidelines for rich Internet applications. We wanted to know more about our interviewees' work, the company's development process, tools and literature they use to get their job done. The question about literature was a roundabout (probing) question to find out if any other kind of literature is helpful – without mentioning the term "guidelines" in the first place. We also wanted to know more about our interviewees' professional backgrounds to better put their experiences into the context of our work. Finally, we addressed the topic of guidelines and tried to find out about attitudes towards using guidelines by out interviewees. Understanding these attitudes was crucial to decide which direction possible guidelines for Rich Internet Applications should go.

Following these considerations, we compiled an interview guide and divided the questions into three topic areas:

- **Topic A:** Professional Background
- **Topic B:** Design/Development process
- **Topic C:** Design Guidelines

Structuring the interview in three separate parts was a good way for our interviewees to get an overview on the topics we will ask them and also to get an orientation on the progress of the interview.

As our subjects were located in Germany and France, we conducted the interview session by the VoIP software Skype and recorded them digitally. This gave us the possibility to set chapter marks in the audio file and work with the recordings more efficiently. The interviews took between 30 and 50 minutes and were conducted on May 4th, 6th and 7th 2007.

4.3 Netvibes

4.3.1 Background on the Company

Netvibes, a start-up company founded in 2005 by Tariq Krim in Paris pioneered the concept of a personalized startpage – an alternative to traditional Web portals. The user can pick from hundreds of modules to create an own digest of news feeds, favourite blogs, e-mail inbox, current weather and much more (Fig. 15). Netvibes has an open Application Programming Interface (API) for anyone who wants to develop own modules and there is a growing number of 3rd party modules to choose from. With the Universal Widget API (UWA) Netvibes has created the possibility to develop cross-platform modules, which can be integrated into other personalized startpages, like *Google personalized Home (iGoogle, 2007)* or even being included as a widget on the desktop (outside a web browser).

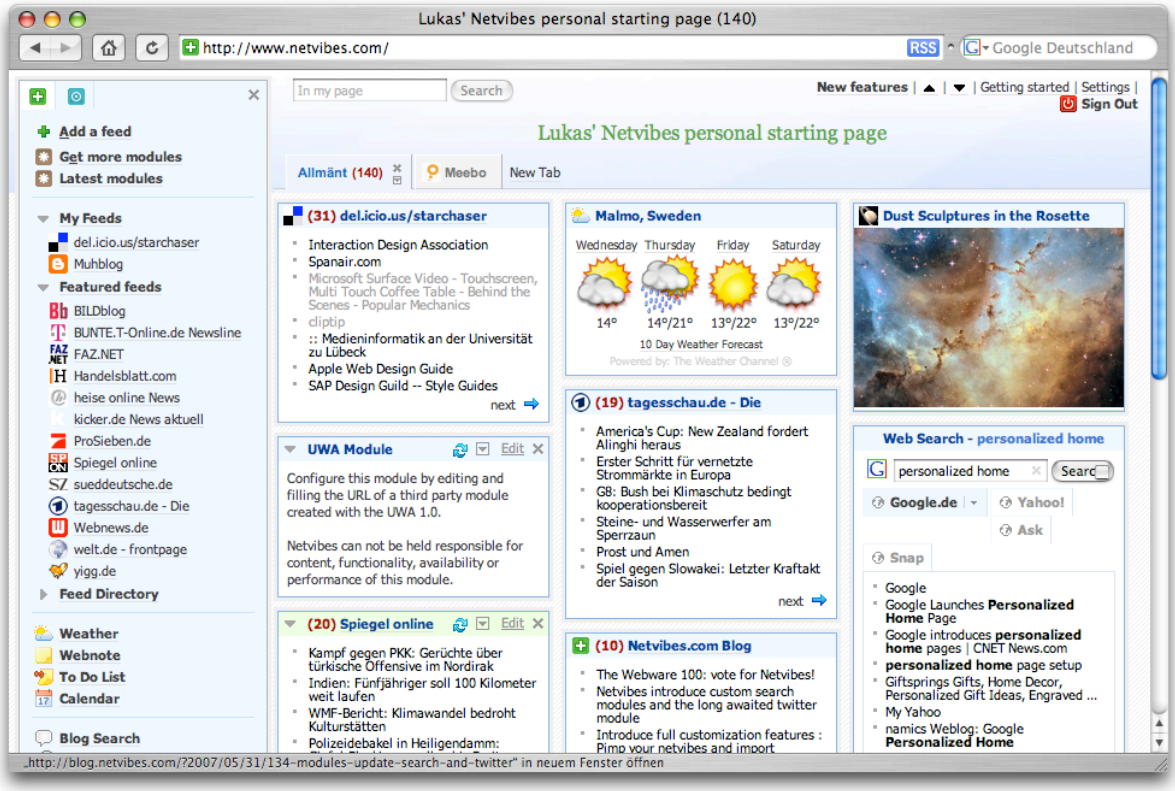

Fig. 15: Netvibes.com Web site

4.3.2 Interview With UX Developer

Our interviewee is a usability consultant for *Netvibes*. Her professional background is psychology and communication with focus on multimedia and usability. She was involved in web design, web development of both HTML and Flash and also worked as software product manager. Currently she is working on user experience (UX) guidelines for the *Netvibes* platform, working closely with developers on mock-ups of new modules and providing feedback on usability of the modules. Therefore she describes her work at *Netvibes* as rather conceptual, as it involves research on interaction and behaviour of the product rather than actual development (which is done by developers themselves).

Concerning the development process of UX guidelines, our interviewee has developed an own approach, which first involved comparing all available modules and making poster-like screenshots to compare their look, style and behaviour. In that way different solutions could be identified and compared to learn about different approaches towards a task of a module. After that the UX guideline document could be divided into different features and functionalities based on the identified approaches.

Our interviewee works closely with developers writing the technical documentation for the Application Programming Interface (API) and discusses the possible solutions with them. As in-house and third party developers are the main target audience for guidelines, working

directly with them and understanding how the features are technically implemented helped our interviewee to realize how other features can be realized in a user-friendly way. Sometimes it seems like a “back and forth” between her and the developers but the dialogue is a very important part of the development process. The guidelines haven’t been tested with users yet, but our interviewee plans to include user testing in future development.

Our interviewee has studied Apple’s Human Interface Guidelines and Microsoft User Interface Guidelines to find out what kind of topics they address and how the document is structured. For encouraging consistent iconography in *Netvibes* modules, she referenced the *Tango Desktop Project (Tango, 2007),* which deals with user experience for open source software. The general structure of the guidelines is very specific to creating modules and it doesn’t include general rules like usage of colours, wording, etc. Widget guidelines seem to be a primer (“I don’t think there’s a widget creating guideline already”).

Concerning the dilemma of corporate identity, this problem surprisingly has no big impact on the development of modules or guidelines. The goal for *Netvibes* is to create a highly customizable platform and therefore the corporate identity does not play such an important role (there is not even a company logo embedded on the main page). The modules’ behaviour and appearance are handled separately, and the UX guidelines mention explicitly that the branding of the modules should not be too strong “[…] so the user is not disturbed so much”, because the appearance of modules should be changeable by the user and/or the environment they are running in. Guidelines on how to incorporate a minimum branding in the modules are also included in the document.

Asked about the main benefit of having UX guidelines for *Netvibes* in place, our interviewee replied that she is sceptical, if developers really would use the guidelines, but if they did, then it could lead to a more consistent look & feel of the platform and possibly *Netvibes’* competitors would concur by designing their modules and platforms similarly, resulting in better compatibility across platforms “[…] which would make it generally easier for such a product”.

As the guidelines are still in development, there are no experiences with the extent to which developers will use them, but our interviewee predicts that they will read them once completely and afterwards use them as a reference. This would be a similar behaviour to the way how developers work with technical references right now: if they know how to solve a problem, there is no need to look it up, but in case they forget – they know where to find the solution.

4.3.3 Netvibes Interview Discussion

Netvibes employs dynamic web technologies to create a desktop-like interaction experience. The modules can be added by dragging them into the content area and they can be re-arranged by dragging them around. The interoperability character of the modules poses similar challenges to the ones we already faced with Java applications, which have to be designed to run on different operating systems and their interfaces fit into different GUIs. The problem has been approached in Java applications by employing different GUI toolkits. The Swing toolkit provides the same look & feel on every platform, whereas the Abstract Window Toolkit (AWT) provides a native look & feel on each of the platforms. Still there is the problem of different operating systems providing their widgets in a slightly different way and in different dimensions, so the native look lacks accuracy of a “real” native application. Also high-level principles of interaction like element alignment, default menu structures, keyboard shortcuts and general UI conventions vary from one operating system to another. (Ref: Java look & feel guidelines).

Netvibes, facing a similar problem with interoperable widgets, approaches the problem by specifying only their behaviour and leaving the presentation to the hosting platform and the user. The same widget can look differently on another platforms, or even for another user. The basic structure (like the settings panel) and behaviour (like draggable elements) remain unchanged.

Netvibes UX guidelines are being created by closely collaborating with the developers, since they will be the users of guidelines. One interesting aspect is the binding of the technical API to the guidelines; developers say, that some things are solved easier the one way than the other, so they would prefer usability guidelines to be fitted to the API. Apparently there is a conflict of the guideline developer and the technical developers, but according to our interviewee these conflicts are then discussed and solved.

Another unique aspect of the *Netvibes* guidelines is their novelty; there are no guidelines existing for web-based widgets yet, contrary to desktop-based widgets/gadgets (Apple 2006, Microsoft 2007). *Netvibes* is hoping to create a de-facto standard for usability of web-based widgets that will be followed by its competitors.

4.4 sTeam

4.4.1 Background on the Project

[open]sTeam (structuring information in a Team) is an open source platform for collaborative knowledge management and e-learning (sTeam, 2007). It is built around a Multi-User

Dungeon (MUD) paradigm, which employs the concept of rooms as the basic underlying information structure. Users can enter rooms, create gates between them and move around, taking all documents in a backpack with them. This metaphor was created to resemble the everyday life at the university: study groups meet at a room, they work on documents they bring with them from home and they drop their assignments at the post box of a lecturer in front of his room. Often universities create the representation of a campus within sTeam, so one can say: “let’s meet at the Cafeteria tonight and work on the project for tomorrow” – which means the Cafeteria in sTeam.

sTeam has been developed as a research project at the University of Paderborn in Germany, supported by the DFG - Deutsche Forschungsgemeinschaft (German Research Foundation) and is nowadays being used at many universities and institutions worldwide. *sTeam* is very versatile, as it consists of a core server, which can be accessed in various ways. One is the obligatory web interface, but one can use it with a Rich Client Platform, a PDA, or even via e-mail. *sTeam* comes with its own API for third party developers and there are other products which can use *sTeam* functionality.

4.4.2 Interview With the Chief Web Developer of sTeam

Our interviewee is employed at the working group for *Computers and Society* at the university of Paderborn in Germany and he is the chief developer of the sTeam user interface. He works on the official sTeam web interface, but also gives advice to third party developers who develop user interfaces for *sTeam*. The special working environment in academic and open source setting is a fertile ground for innovation, because there is much space for critical discussions with people from different backgrounds and no strict deadlines for product releases. There is a rough roadmap with development goals for the next releases, but it is flexible, so the team can change its development plan accordingly.

There is a separation between core system developers and web interface developers and our interviewee mainly develops the web interface, but there are programming aspects connected to the web interface as well, and it helps if the designer and developer can communicate about the functionality of the application. The designer has to understand the purpose of the high-level functionality (like the concept of rooms in sTeam) of the application in order to design a usable interface. Because sTeam can serve as a back-end data repository and be used by a variety of front-ends, very often the interface designers never speak with the developers. Documentation and APIs provide all information needed to make the necessary connections. However, concerning Rich Internet Application technologies like AJAX, designers have to cooperate very closely with developers to create a good user experience, because the interaction aspect is so closely tied to the interface and current web design tools are not capable of designing the interactive part. Also programming elements

like JavaScript interact with design elements like HTML and CSS, so no strict line can be drawn between the designer and the developer with regards to Rich Internet Applications and "[…] at that point you are not able to clearly differentiate which is design and which is the programming".

When it comes to accessibility, sTeam tries to be as accessible as a web application can be; recently a graduate student at the university of Paderborn has redesigned the web interface according to the WAI's (Web Accessibility Initiative) WCAG (Web Content Accessibility Guidelines). Our interviewee stated that no web application is truly accessible, but it was the best they could do.

The toolset our interviewee is using for his work consists of a web-programming environment (jEdit), a concurrent versioning system (CVS) to work collaboratively on the system with other developers and *Guiffy* as a tool to track the changes made in the sourcecode. The *Firefox* web browser including plug-ins to make the page structure visible is the main development browser; other browsers are used for compatibility testing. As our interviewee uses Mac OS X as his development platform, he has a virtual machine (*ParallelsI)* installed to perform tests with Windows-based browsers. An FTP client (Cyberduck) is used to transfer the files from the development platform to the server. Adobe Photoshop is used for creating Interface mock-ups and working on particular graphic files for the web application.

For communication with other developers, our interviewee makes extensive use of E-mail, Instant Messaging, VoIP (*Skype*) and the *iCal* Calendar to manage his work. Instant messaging plays the biggest role.

Regarding literature, our interviewee mostly uses web-based documents, including programmer's and HTML references. Guidelines for accessibility included German accessibility guidelines "*Barrierefreie Informationstechnik-Verordnung (BITV)"* and "Kriterien zum BIENE-Award 2006" (BIENE, 2006) by the *Stiftung Digitale Chancen* (Digital Opportunities Foundation). Another guideline document, which was developed by Prof. Keil, who is the head of the working group, is the "criteria for reducing sequentiality". These criteria are meant to design systems without sequential modal dialogs (i.e. multi-step wizards). Our interviewee stated that these guidelines are always followed, but as developers know them, they just keep them in their mind, without using the document in their daily work.

Asked about the question of consistency vs. corporate identity of websites, our interviewee thinks that consistent behaviour of websites is especially important with common features of all websites (i.e. contact form, search, etc.). However if new, "cutting-edge" functionalities are included, than this breaking up with the previous experience of users with websites is highly dependent on the level of experience of the user. For experienced users or professional web developers it is no problem to find way within a web site, which is designed differently, but for

a novice user it can pose a big problem. Exceptions mentioned by our interviewee exist however, taking the example of *Google maps* or *Flickr*: “If the consistency breaks, but it feels nice and easy to use, I think the consistency absolutely doesn’t matter”.

A question that connects directly to this is, if guidelines help building consistent applications or if they rather restrict developers in their creativity. Our interviewee stated that this is always a trade-off and every website owner has to deal with it. Either the web application will work like any other and developers will be restricted, or the developers will find new solutions, but the website won’t behave like all others. If new technologies are found to solve the problem better, then it’s ok to break up with consistency, but for regular web pages, which present information it is better to stay with established guidelines.

Because of the variety of technologies and new interaction possibilities, our interviewee thinks that RIA shouldn’t impose too specific rules on developers, but rather give them general advice and suggestions on how to solve problems. The advantage would be a consistency of the general user experience, even with changing technology. Another reason mentioned is the short “life expectancy” of a guideline document, if it’s too specific. With general guidelines there is a chance of establishing long-term rules for this domain. Considering using Desktop UI guidelines as a source for RIA guidelines, our interviewee mentioned that one could reuse general guidelines, but should also include specific affordances of the web into RIA guidelines. He mentioned examples like communicating page updates, handling of the back-button and bookmarkable URLs, which is often problematic with AJAX- or Shockwave-based applications.

Asked about users’ mental models of web pages and web applications, our interviewee thinks that it depends on the level of experience of the user, how his or her mental model is shaped. In general however, he thinks that users employ a page-based model on information-oriented sites and an application-based model on sites containing web applications. However, if Rich Internet Applications become more common, there is a possibility that users will get used to the application-based model and don’t feel the need to have control over it with the web browser. Moreover, in certain applications the function of the back button is not defined (i.e. after making changes to a web-based calendar) and users will restrain from using it, if they cannot predict its outcome.

Relating the functionality of the back button to the consistency of using the back button, it seems not important for the button to have a consistent functionality, because the users define its function in their mental model on the fly. If the back button makes sense on a certain site, it will be used, if it doesn’t make sense, users won’t use it. Our interviewee stresses that: “the consistency is not the main point; it’s one point, but it’s very low priority”.

Concluding, he says that it is all about a good overall user experience, which can be achieved even with an inconsistent user interface.

4.4.3 sTeam Interview Discussion

The interview touches upon many important issues to be considered when designing guidelines. First of all, it has to be considered, that developers and designers have to work hand-in-hand, because current technologies for Rich Internet Applications interweave the roles of the designer and the developer: the designer has to rely on the developer's help to implement rich interaction and the developer has to rely on the designer's expertise on how to design the interaction to be usable. This finding has been confirmed by our interviewee. Another finding is the strong link between usability and accessibility. It is difficult to design accessible systems, and accessibility guidelines seem to be very helpful for this task. Other guidelines however are used more on a reference-basis, as the general principles are well known by the developers and don't need to be followed step-by-step in daily development work. In a way, they are used similarly to programmers' references: the developers refer to them only of they don't remember how to use a certain statement, but are never read completely. The question about consistency vs. corporate identity has been found to be similar to the one about consistency vs. innovation. Both of them are trade-offs and have to be dealt with individually. In addition to this, experienced users don't seem to have a problem with inconsistencies caused by different corporate identities. This is mostly attributed to the fact that high-level design principles are retained, even of the corporate identity is different. Users seem to be experienced enough to navigate successfully despite different layouts and site structures, if basic principles are maintained. Our interviewee also distinguishes between the level of experience of users; more experienced users need less consistency than less experienced ones. According to our interviewee, consistency seems to be inferior to the general user experience and it is not important if a website or web application is consistent with others, but if it provides a good user experience. Taken Google maps as an example, it does not follow any web guidelines, and still is a usable web application. In our opinion however, there is still the question remaining, how to measure the quality of user experience. Claiming that guidelines are not necessarily needed for a good user experience doesn't solve the usability problem, nor does it help developers.

What we can agree on is the fact that high-level guidelines are more useful for developing Rich Internet Applications, as developers don't like to be constrained in their creativity and also the technology is evolving faster than guidelines, making any kind of specific guidelines useless if they cannot be employed with the latest technology. We found suggestions from our interviewee about using desktop guidelines as a source for high-level principles and investigating web-specific principles of Rich Internet Applications (handling the back-button,

bookmarking URLs, communicating asynchronous page updates, etc.) were useful and they have been included in our RIA guideline outline.

Talking about the users' mental model of web pages vs. Rich Internet Applications, our interviewee thinks that users' mental models of the web will change with time and user experience.

4.5 MindLab

4.5.1 Background On The Company

Mindlab is a company based in Esslingen, Germany, providing solutions for online user testing (Mindlab, 2007). It works for customers from the pharmaceutics, financial and automobile industries. *Mindlab* has also developed a tool for analyzing user behaviour on a website. User movements can be tracked and later on analyzed to find out how they interact with a particular navigation system. Our interviewee at *Mindlab* is mainly involved with the visualization tool for this data and he will talk about his work with the application.

4.5.2 Interview With An IT Consultant / Interface Designer For Mindlab

Our interviewee is employed as an IT consultant at *Mindlab*. His educational background is cognitive science and graphic design. At *Mindlab*, where he was employed for the past 9 months, 60% of his time goes for supporting the company's clients with *Mindlab's* product, a Web-tracking software, which tracks web users' behaviour on web sites and visualizes the gained data (i.e. number of sessions initiated, visited areas, etc.). 20% of his time is devoted to interface design of the web-tracking solution and the other 20% he is involved in the general design of the product.

According to our interviewee, the distinguishing aspect of his work at *Mindlab* is the kind and amount of data he is working with. The data represents user behaviour on web sites and there are different user groups that need to have different views on the data. The amount of data to be dealt with is ca. 40.000 clicks per day and it has to be visualized in a usable way to the different user groups.

The visualizing application uses HTML and JavaScript on the front-end and PHP on the back-end. This means it is not a Rich Internet Application in a strict sense. Our interviewee mentions that using flash was not considered because it has been found to be less accurate in presenting the data and the development (especially debugging) was "horrible". It also couldn't deal with big data sets very well. In the past our interviewee has developed a java

applet for the visualization of user paths on the website and this tool was truly interactive, but now the solution relies on HTML.

As *Mindlab* has been a pure IT company before our interviewee joined it, there were process models like the V-model employed for software development, but there was no User Experience involved in the development. This fact posed difficulties, because competing companies have put more effort into interface design than *Mindlab* did. “They all knew they had to do something [but] it was very hard to get this into the heads of the developers”. As the company moved towards web-based software there was a need for an improved development model, which incorporated aspects of User Experience. The communication between web designers and developers is good, but doesn’t follow any formal ways. Our interviewee sees some improvement potential here, as things are communicated ex-post instead of planning it in advance.

Apart from Email and Office tools, our interviewee uses *Eclipse* as an integrated development environment, an SSH client to connect to the server, and Adobe Photoshop, Illustrator and InDesign for the interface design. Asked about literature used, our interviewee mentioned Eysenck (1995), because it contains the basics about mental models, problem-solving strategies, perception principles, etc. Our interviewee states that he prefers to work with this book than working with more strict guidelines. Other literature he mentioned was “Software-Ergonomie. Grundlagen der Mensch-Computer-Kommunikation” (Software ergonomics, basics of human-computer communication) by Michael Herczeg, and “Web Usability - Das Prinzip des Vertrauens” (“Web usability – the principle of trust”) by Martina Manhartsberger and Sabine Musil. Both books have a guideline character, but the guidelines are on a high level, so they can be applied universally. Our interviewee mentioned Jakob Nielsen’s useit.com website, which he strongly dislikes for providing ready-to-apply solutions, which lack the originating principles. As every problem is different, specific guidelines make no sense to our interviewee. Another guideline, used especially when communicating with other companies is the DIN ISO 9241: Ergonomics of human-system interaction standard. The reason for using this guideline is to create a common understanding about usability among partners, who are not familiar with usability. The DIN ISO standards are well known in many industries and they are a good starting point for discussions. The most indispensable book however is Eysenck (1995), because of its universal applicability, but he mentions, “the best guide is your experience”.

Usability is also improved by so-called “advisory boards” which are feedback sessions from end users who use a beta version of the software before its final release. These sessions aren’t done in a lab, but at the customers’ work setting, using their productive environment. Since the software has very few customers, these feedback sessions are very important to

get the users' feeling about new features and it makes sense to test with the actual end users instead with a different user group.

Concerning the corporate identity, the situation is similar to the communication problem: there is no real strategy how to incorporate *Mindlab's* corporate identity into their product, but it is planned to stress more upon this topic in the future. The reason for that is partly the competition, which is putting more attention into a complete image of their solutions. *Mindlab* wants to strengthen its image by integrating its corporate identity, the behaviour and its product's interface into one consistent image. When speaking of the consistency vs. corporate identity problem, this doesn't seem to be a big problem for *Mindlab*, because their product's interface is customized very weakly for their customers, mostly involving changing colours.

Asked about how web design guidelines should look like to be useful, our interviewee answered that solid knowledge of the basics of cognitive psychology together with examples of good web pages is a good starting point, because it encourages reflection on why something works well: "When I see a good page [then] I see what's working, why it is working good". He was missing the link to the basic principles in existing guidelines, which would help transferring rules to other types of problems.

Regarding guidelines for Rich Internet Applications, our interviewee distinguished between browser-based applications (i.e. AJAX or Flash) and non-browser based applications, like the Eclipse Rich Client Platform (RCP). These two approaches solve the same problem, but are completely different in regards to the user experience. The Rich Client Platform comes represents a desktop environment, whereas a web application is embedded within a web browser, which represents a web metaphor. Our interviewee thinks that the user experience is dependent on users' attitudes [later clarified as mental models] and their expectations. If they are in a desktop environment, they use applications differently than when they are running in a web browser. According to our interviewee, the web differs significantly from the desktop, because it opens up a new space, which is not restricted by menus, windows or icons. The designers have the freedom to arrange it as they want and break free from the limitations of desktop applications. On the other hand, users are used to use desktop applications and it would be good to give them a "smooth transition" by offering similar behaviour with web applications. Hence, it would be a good idea to incorporate parts of desktop usability guidelines into RIA guidelines and adding web-specific guidelines to it.

Asked about mental users' mental models of web pages vs. web applications, our interviewee stated that this is a fundamental principle upon which web applications should be built. The users have a very clear attitude and expectations of the environment they are using. He points out: "If you would make a web application as a desktop application, it

wouldn't work". In case of web applications users expect different structure and behaviour than from desktop applications and so is their usage different. As an example, our interviewee mentions drag & drop functionality, which is only associated with desktop applications. Users don't expect drag & drop behaviour in the web environment and that is a reason why it is so rarely used. Here the users' expectations shape the way, how web applications are being designed. These mental models can change if such features would be more commonly used, but it will take some time, as mental models are consolidated in users' minds. One way to speed up this process is to communicate any "unexpected" functionality to the user, so he knows that they will work in the web environment. These are issues that also should be included in RIA guidelines.

4.5.3 Mindlab Interview Discussion

Our interviewee at *Mindlab* gave us a good insight into his work, including the tools and literature he uses, which turns out to be high-level oriented and focussing on principles from cognitive psychology and ergonomics. The only guideline he uses is the DIN ISO standard, which is helpful in communicating with customers from different industries. In Germany, the DIN ISO standards are widely understood across many industries and it is important to develop a common understanding of a topic when talking to customers. In his daily work, the DIN ISO standard doesn't play an important role.

The usability of Mindlab's product is ensured by advisory boards, that is collecting feedback on pre-release versions from customers. The advisory board is a useful institution because it doesn't require lab sessions and is conducted in a productive environment. The customer base is small enough that the product can be tested with actual users, so their suggestions can have a big impact on the outcome of the next version of the product.

Speaking about guidelines, our interviewee doesn't like strict guidelines, like those published by Jakob Nielsen. He rather wants to understand the principles behind guidelines and decide specifically if they apply or not. This requires more experience in cognitive sciences and graphic design and it makes sense, as this was his field of study. So is also our interviewee's suggestion for Rich Internet Applications to be formulated on a high level to be transferable to different problem areas and new technologies. Only in this way the guidelines can stay current over an extensive period of time.

The aspect of users' mental models is very central to our interviewee, as he thinks that the way, how users expect applications to behave shapes their development. Designing intuitive applications involves knowing and addressing these expectations. The question remains of course, if mental models should shape the guidelines or if the guidelines should shape users' mental models. It is imaginable that appropriately designed applications will change the way,

how users interact with them. In any case, the aspect of mental models should be a part of RIA guidelines.

5 Discussion

Before discussing our research, we would like to recapitulate our research outline in the following diagram (Fig. 16).

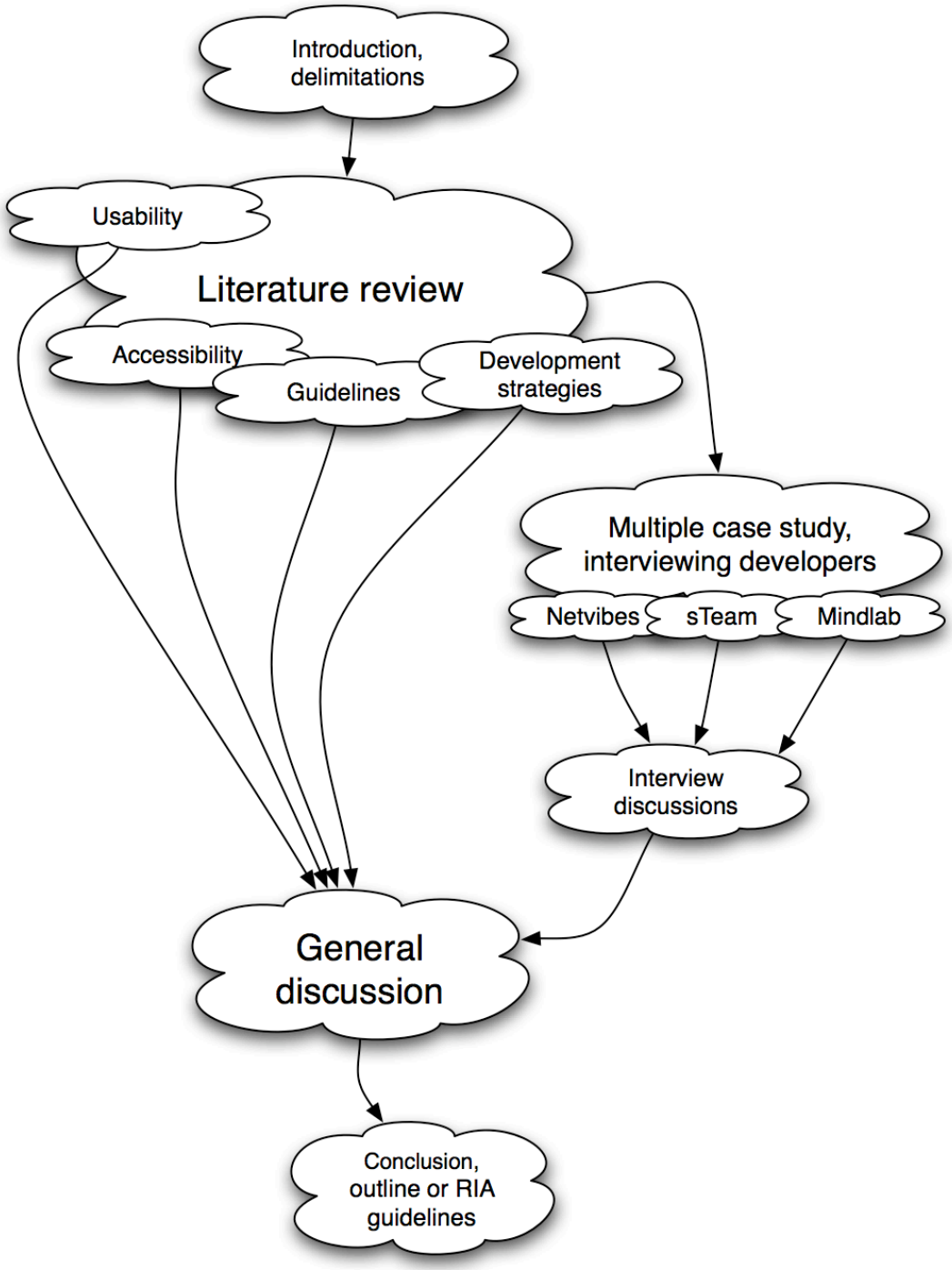

Fig. 16: Research structure

5.1 Web Usability vs. Desktop Usability

Compared to the relatively well explored fields of both web usability and Desktop UI design, there has been only little work done to support web developers in designing usable Rich Internet Applications. One reason is the novelty and ever-changing technology for developing such applications with new AJAX frameworks emerging constantly and new features made possible for web applications. Developers, facing new technologies and having no guidance how to use them, design interfaces for web applications that are neither consistent with other web applications, nor with existing desktop operating systems. The result is that users have to learn the features of every web application they encounter and they cannot transfer their knowledge from one web site to another. In case of desktop applications, this would be a violation of the consistency principle, but as every web application has its own interface, users have learned how to interpret interface elements that look differently but have the same function, like site navigation, dynamic menus, hyperlinks, etc. This is something that Rich Internet Applications don't communicate to the users, so they have to explore the interface in order to learn the application's features. Some features are easier to communicate than others, because they resemble desktop UI counterparts, but certain behaviours, like drag & drop were never before seen in web applications and it is unlikely that users will try to explore them, unless they are told to do so. New ways of communicating such features will have to be found and applied consistently on many websites. As an example, there exist some common understanding of how to communicate a web link to a file to be downloaded. In most cases it is indicated by a downward-facing arrow. Though the icon is never exactly the same, the meaning seems to be clear to the users and web designers and it seems to be universally recognizable. If the same development happens with Rich Internet Applications, there is a chance that although having completely different visual appearance, they will be universally understandable.

5.2 Perception Of Guidelines By Developers

In general, guidelines have been found useful both by researchers and practitioners. Our literature study has revealed that guidelines are successfully used at various stages of development and they exist in a variety of forms. Our case studies have also shown that developers appreciate having guidelines, however they prefer them to be on a general level, to be universally applicable and not constrain developers in their creativity. Our interviewee from *sTeam* has mentioned, that consistency is not necessary for good usability and we agree as well that the level of users' experience has a big impact on the perception of usability. Guidelines therefore should not be solely targeted at achieving consistency, but on the overall user experience, which is in fact difficult to define, but the best solution to the

problem is formulating guidelines on a high level. The same applies to our interviewee from *Mindlab*, who wishes guidelines to have a stronger connection the underlying rules of cognitive psychology (which is not surprising, as his educational background is exactly in this field). We think however that guidelines should be formulated on a level understandable for the average developer, who has no background in cognitive psychology or graphic design. In fact, guidelines are meant to fill this knowledge gap, without the developers having to know why they work. High-level guidelines can still have useful character, but in our opinion they don't need to legitimate their existence to the developers.

Another interesting aspect found during the interview with *Netvibes* is the fact that developing guidelines should be done with a bigger audience in mind. Originally, *Netvibes'* widget guidelines were targeted at in-house developers, but by opening the API to third party developers, these guidelines could be used in a much bigger scope. A similar development took place with Apple's Human Interface Guidelines (Apple, 1992), which became an important reference, not only to Apple developers, but also to web designers and desktop application developers on other platforms. The reason for that were the universally applicable principles that apply to any desktop (or post-desktop) platform. This is another argument for formulating guidelines on a general level. One important element desired by our interviewees was including examples for rules as they help to understand the rules better and it is always a good learning experience to see how something can be solved.

5.3 Accessibility of Rich Internet Applications

Then, there is also the problem of accessibility of Rich Internet Applications, which is connected to usability, but has a far more severe impact on people with disabilities that the usability problem has on regular users. Users can work their way around a system with poor usability if they have to work with it (and if not, they will use a competitor's system), but when a disabled user has no access to information at all, this is the same type of exclusion if a public library would have no wheelchair ramp. In the case of Rich Internet Applications the solution is not straightforward, because appropriate standards are still in development. Until then, an accessible alternative solution should be provided for equal access to information and services. Also, the awareness of developers for accessibility issues has to be increased and this can be only achieved if accessibility becomes an integral part of RIA guidelines.

5.4 Distinctive Features of Web vs. Desktop Guidelines

Regarding the question, why there are no guidelines yet existing for Rich Internet Applications may have several answers. We think that the most probable answer is the fact

that the related technologies are very young and so are the applications built with these technologies. There is a lot of experimenting going on with Rich Internet Applications and it is not yet clear which solutions are the best ones. If a part of a page has been updated without reloading, how to indicate the changes? How to communicate that the application has lost the connection to the server? These problems have no relevance in desktop applications and the solutions are still being explored. There are some issues that can be directly transferred from desktop UI guidelines to web applications, others are completely new and yet others are irrelevant. In order to design Rich Internet Application guidelines, one should compare desktop guidelines with web guidelines (regarding different levels of specificity) and find common and distinct features in both of them, and check their applicability in Rich Internet Applications. Previous research has shown that Web guidelines have very little in common with desktop UI guidelines (Ratner, 1996), and it seems obvious, since the desktop and the web are two essentially different platforms. However, with Rich Internet Applications gaining ground, it makes sense to learn from desktop guidelines.

In contrast to Desktop Interface guidelines, which are structured quite similarly and address the same topics for various operating systems, web usability guidelines can differ very much in the level of specificity and can address web usability from many perspectives. Also the border between web sites and Rich Internet Applications is vague: a simple JavaScript for revealing elements of the page could be already considered a Rich Internet Application in the same way as a complex Web application using desktop-like patterns of behaviour but not using any dynamic web technologies could be categorized as a “classic” Internet Application.

In our case studies we found out that developers don’t follow guidelines very strictly, all of them have read usability guidelines in some point of their work, but they rarely refer to them, because they have experience doing development work and they intuitively apply solutions that work best (so their claim). The question is if the work results confirm this statement or if guidelines are written in a way that is not suitable or attractive to web developers. An analogous study has been conducted with accessibility guidelines and it has been found that developers had troubles implementing them correctly (cf. chapter 3.5.3).

The fact that Rich Internet Applications develop so rapidly could be a reason for authors to restrain from publishing work that could be outdated in a few months. On the other hand, online magazines for web developers publish and discuss ideas related to interface design of Rich Internet Applications. As the discussion elements seems to be very important in this early phase of guideline creation, a suggestion that we got from one of our interviewees makes sense: creating the guidelines not as a linear document, but as a wiki, which can contain articles that anybody can discuss about and contribute new solutions on the fly. Once a stable set of guidelines is established, one could move on to creating a compendium as a static document. A wiki is often the preferred documentation and communication tool for

developers, and we believe that communication between developers is particularly important in this stage.

During our work creating a set of basic principles for Rich Internet Application guidelines we couldn't work closely with developers and users, therefore we decided for a literature review and a case study to find out how developers work and how they envision possible Rich Internet Application guidelines. We learned that web applications are so diverse and multi-facetted that it makes much more sense to discuss general principles and issues than trying to solve concrete problems of how to design widgets. We believe that a top-down approach is more suitable, because first one has to gain understanding of the principles and the context of Rich Internet Applications and only after doing that it makes sense to go into detail. As we believe that Rich Internet Applications are not yet fully understood by users and as long as developers are still exploring the possibilities, the understanding of Rich Internet Applications is still in the process of shaping. With our outline for RIA guidelines we would like to give suggestions and point out issues that we believe are important to consider while designing Rich Internet Applications.

5.5 Guideline Development Methodologies

We were considering employing one of the methodologies for creating guidelines (cf. chapter 3.6.6 Developing Guidelines), but they turned out not to be suitable for our needs, as they were intended for creating specific guidelines or to be employed in an organizational context. We therefore decided to create our own approach, taking interaction design aspects from existing desktop usability guidelines and content organization aspects form web guidelines. We also incorporated accessibility, Web 2.0 and mental model aspects from our literature review, all of which were confirmed as important elements by the developers. As we touch upon new ground within Rich Internet Applications, we believe that this "pragmatic" approach will produce a guideline outline, which can serve as a basis for further research.

When Rich Internet Applications will reach a sufficient level of maturity, we expect patterns to emerge, which will represent a consensus in how the user interface should look like and behave. The question whether guidelines are necessary for these patterns to emerge, or if the patterns are caused by guidelines is a typical chicken/egg problem, but we think it has also to do with a critical mass of applications being developed and actively used and shaping the understanding of the usage we mentioned before. In any case, guidelines could further ensure consistency within and between Rich Internet Applications. Novice developers would get an understanding for the context of use and mental models users have of particular types of sites and interaction modes. We again would like to stress that we believe high-level principles to be more important here, as they exhibit a general understanding without

constraining the developers in their creative work. Take a shopping web site as an example: there is metaphor of a shopping cart users can put products in and stroll around the shop while the shopping cart still contains the products. To buy the products the users use a “checkout” site where they choose a payment method, enter their address details and finalize the transaction. The metaphor of a shopping cart and the mental model that items are stored in a (virtual) shopping cart during a session on a shopping website has become the way how all shopping web sites work. If a new shopping web site is being developed, this mental model should be kept in mind if one wants the users to use the shop intuitively. One could introduce new ways of placing the items in the shopping cart (i.e. by drag & drop of the representing pictures), but the mental model of a shopping cart still remains unchanged.

5.6 Users’ Mental Models

Another question is the users’ general mental model of the web, which is changing because of Rich Internet Applications. The page-based mental model, meaning that users can navigate back and forth between web pages doesn’t work any more with Rich Internet Applications. This aspect has been confirmed by our interviewees who stated that users’ mental models differ not only from web pages to web applications but also by the level of their experience. Depending on the application, the back button doesn’t work properly or produces unintended results. Users have a different mental model if they use an Internet Application than if they use a static web page. Users instinctively know when they can use the back button and when it doesn’t make sense. The fact that the user has to decide whether a simple software function like going back to the previous page will work correctly is an unprecedented example of breaking all rules of simplicity that have made using the web so easy. The user now has to carry out the cognitive task of deciding whether the browser controls will work with a particular web application or not! If Rich Internet Applications won’t solve this problem through appropriate design, the browser controls will become unusable in the long run. However, just as the users’ mental models change when using web pages and web applications, respectively, possibly the browser navigation functions should change as well towards navigating between sites, instead between pages. In that way every web site would be a closed environment and provide own navigation controls which the user would use and the “global” browser navigation would be used for the “big steps” between sites. In this way the web browser navigation controls could catch up with the users’ mental model and become usable again.

5.7 Benefits of high-level guidelines

Keeping the guidelines on a high level would also help avoiding the two major drawbacks of guideline documents: firstly, guidelines could pose a barrier to innovation – developers would

rely on ready solutions and wouldn't try out new approaches to solve problems. Secondly, too specific guidelines wouldn't be transferable to other kinds of problems and only useful in particular situations – which would make the purpose of having guidelines useless.

Keeping the innovation going on has always been an important aspect of the Internet. But with emerging Rich Internet Applications we have the possibility not only to redefine the way how we work with the Internet, but also get rid of the Desktop paradigm which has been unchanged since the first Graphical User Interface was invented by Xerox PARC in 1981. Aza Raskin in his *GoogleTalk* presentation (Raskin, 2007) talks about the opportunities available with Rich Internet Applications and the risk of simply porting an old idea to a new medium. In this sense, one should think thoroughly before establishing guidelines, which come partly from the desktop and partly from the web, but introduce no groundbreaking paradigm shifts. The Web 2.0 concept of mash-ups, creating new applications by using services of existing ones poses a new way of thinking about web applications. They are not monolithic units any more, but consist of parts, which can be reused and thus re-contextualized. The same is true for information organization paradigms like collaborative tagging. These paradigms are not merely about organizing information with keywords, but a completely new way of thinking about how information can be managed collaboratively and what implications this approach has in terms of connecting information and communication in its very literal sense.

To sum up, we believe that establishing guidelines can have a much bigger impact on the very nature of information technology than just on how to design usable Internet Applications.

6 Conclusion and Future Work

6.1 Conclusion

Concluding our work we would like to recapitulate how our findings relate to the original research question: “How RIA usability guidelines should be designed to be useful for web developers and at the same time support the affordances of current development trends in web applications”.

Our empirical study has shown that developers that developers acknowledge guidelines as a useful tool for their work. Even though guidelines are not used on a daily basis, all developers have dealt with guidelines in some point of their work and use them as a reference if needed. The level of specificity of guidelines depends on the developers’ task. Some developers need more general guidelines; others would like to have instantly applicable solutions. In general, developers don’t like to be constrained by too strict guidelines, as they want to be creative. Moreover, giving developers readymade solutions ignores their potential to create innovation. Novel solutions like *Google maps* or collaborative tagging don’t follow any web guidelines, but they are still intuitive enough to be usable. Our conclusion to this is, that guidelines should rather *guide* developers, and provide them with general rules rather than give them ready solutions.

When it comes to Rich Internet Applications, all developers agreed that guidelines for this new kind of applications would be a useful addition in their work. The developers also agreed that as the field of Rich Internet Applications is developing, guidelines should be stated on a general level, referring to general principles mentioned in Desktop UI guidelines and Web usability guidelines respectively. One developer suggested publishing guidelines as a wiki to allow quick changes and involving a community to expand them collaboratively, which in our opinion is a useful option, considering the rapidly changing technology of Rich Internet Applications. A wiki-based guideline document would allow for non-linear expansion and many alternative solutions to a problem to be discussed directly on the page.

The developers agreed that users’ mental model of web applications is substantially different from web pages and guidelines should reflect this fact. Our suggestion to create RIA guidelines as a hybrid between Desktop UI guidelines and Web usability guidelines was acclaimed by our interviewees and we designed the outline of our guidelines accordingly.

The outline is just the first step towards a guideline document, and it should serve as a basis for further development of more specific guidelines.

The literature study has shown that usability and accessibility are very closely related topics and it is difficult to develop accessible Rich Internet Applications with current technologies. Currently, necessary standards to make Rich Internet Applications accessible are still under development and our proposed RIA guidelines should therefore specifically address this aspect to strengthen accessibility consciousness among developers. When Rich Internet Applications will grow out of experimental stages and eventually replace desktop applications, accessibility will play a big role and it is the responsibility of developers to ensure that these applications are accessibility in the very beginning. Therefore, both usability and accessibility are part of our proposed guidelines.

Based on these conclusions, we suggest the following aspects to be included in RIA guidelines. As stated before, it is a rough outline, which would need to be expanded in depth and filled with details after further investigation.

1. Who should read these guidelines?
2. Introduction to Rich Internet Applications
3. General Design Principles
    3.1. Visual Perception
    3.2. Gestalt rules (Mullet 1995, p.91 ff.)
    3.3. Metaphors
    3.4. Mental Models
4. Development process (how to plan & develop RIAs)
    4.1. Design
    4.2. Prototyping
    4.3. Usability testing
    4.4. Technological considerations (which technology suits the intended goal best?
5. Site/Application structure
    5.1. Information organization paradigms
        5.1.1. How to design Search?
        5.1.2. How to incorporate Tagging?
        5.1.3. How to include user suggestions?

5.2. Navigation

5.3. Human-readable URLs

6. Look & Feel

6.1. Colors

6.2. Typography

6.3. Layout

6.4. Presentation for different media (print / mobile / …)

7. Site/Application Interactivity

7.1. Interactive elements of the site

7.2. Responsiveness

7.3. Communicating spontaneous updates on the page (Maurer, 2006)

7.4. Handling the back button (when to include functionality on a new page, and when on the same page)

7.5. Consistency

7.6. Dealing with the undo feature

7.7. Communicating "non-standard web features" (like drag&drop)

8. Accessibility considerations

8.1. How usability and accessibility are interrelated

8.2. How to design for accessibility

9. Cross-platform considerations

9.1. Building with web standards

9.2. Browser compatibility

9.3. Device independence

10. Internationalization

10.1. Managing multilingual output transparently

10.2. Managing multilingual input (international character set)

11. Mash-up applications

11.1. Dealing with external modules

11.2. Offering serices to other sites (RSS-feed, Videoplayer, etc. / how to design the UI for outgoing services?

12. User Community

12.1. Involving/harnessing the community

12.2. How to deal with user-generated content? (Legal, ethical aspects)

12.3. Sustaining the community (how to keep users on the site)

13. Identity Management

13.1. Ensuring trust

13.2. Lost passwords

13.3. Single sign-on systems (i.e. openID)

14. Further Literature

6.2 Future Work

The presented outline is only a starting point of creating RIA guidelines. It presents the direction for further research on both high-level principles and specific guidelines that can be directly applied in development. Further research should therefore include:

➔ Evaluating the preliminary outline with developers to find deficits,

➔ Further expanding and detailing the guidelines,

➔ Formulating more specific guidelines as soon as the basic structure is found to be stable,

➔ As soon as the outline is more complete, conducting a bigger study with developers to see if the guidelines are useful for them and how to improve them.

➔ Review accessibility guidelines and see how they can be adopted to fit into RIA guidelines.

➔ Finding a good way to manage RIA guidelines, so they stay up-to-date, even with new technologies emerging. It would be interesting to find out if a wiki is a self-sustainable system to manage the guidelines over time.

References

ACM Code of Ethics (1997): http://www.acm.org/constitution/code.html

Apple Computer Inc. (1992): Macintosh Human Interface Guidelines. *Addison-Wesley Publishing Company.*

Apple Computer Inc. (2006): Apple Human Interface Guidelines. *http://developer.apple.com/documentation/UserExperience/index-date.html*

Adkisson, H. (2002): Identifying De-Facto Standards for E-Commerce Web Sites, *Technical Communication master Thesis at University of Washington.*

Babiker, E. N, (1991): A Metric for Hypertext Usability*. ACM Press, NY, USA.*

Beier, B. (2003): The Bull's-Eye: A Framework for Web Application User Interface Design Guidelines. *CHI 2003, Ft. Lauderdale, Florida, USA.*

Benson, C. (2004): GNOME Human Interface Guidelines 2.0 *http://developer.gnome.org/projects/gup/hig/*

Berners-Lee, T. (1999): Weaving the Web. *Harper, San Francisco.*

BIENE (2006): http://www.biene-award.de/award/kriterien/

Borges, J. A. (1996): Guidelines for Designing Usable World Wide Web Pages. *CHI'96 Companion, Vancouver, BC, Canada.*

Bruno, V., Tam, A., Thom, J. (2005): Characteristics of Web Applications that affect usability: A Review. *Proceedings of OZCHI 2005, Canberra, Australia. November 23 - 25, 2005, ACM.*

Card, S. K., Moran, T. P., Allen, N. (1983): The Psychology of Human-Computer Interaction. *Erlbaum, Hillsdale NJ.*

Carroll, J. M. (ed.) (2001): Human-computer interaction in the new millennium. *ACM Press/Addison-Wesley Publishing Co. New York, NY, USA*

Cato, J. (2001): User-Centered Web Design. *Addison Wesley Longman.*

Choi, Y. S. (2006): Are "Universal Design Resources" Designed for Designers? *ASSETS'06, Portland, Oregon, USA.*

Chrisholm, W. (ed.) (2001): Web Content Accessibility Guidelines 1.0. *Interactions, Jul/Aug 2001.*

Cohen, A. (1995): Tools for Working with Guidelines. *SIGCHI Bulletin, Vol. 27 / No. 2, April 1995.*

Colwell, C. (1999): Evaluation of Guidelines for Designing Accessible Web Content. *IFIF TC. 13 INTERACT'99 Workshop.*

Cordes, R. E. (1981): Software Guideline Development: A Proposed Methodology. *ACM, USA.*

Creswell, J. W. (1998): Qualitative Inquiry and Research Design: Choosing among five Traditions. *Sage Publications, Inc., CA, USA.*

Dalal, N. P. (2000): Cognitive Design of Home Pages: An Experimental Study Of Comprehension On The World Wide Web. *Information Processing and Management 36 (2000). Elsevier Science Ltd.*

Dapper (2007): http://www.dapper.com

del.icio.us (2007): http://del.icio.us

Eysenck, M. W. (1995): Cognitive Psychology: A Student's Handbook. 3rd ed. *Lawrence Erlbaum Associates Ltd., East Sussex, UK.*

Flemming, J. (1998): Web Navigation: Designing the User Experience. *O'Reilly & Associates, Inc., Sebastopol, CA., USA.*

Flickr (2007): http://www.flickr.com

FOAF (2007): http://xmlns.com/foaf/0.1/

Gedenryd, H. (1998): How Designers Work: Making Sense of Authentic Cognitive Activities. Jabe Offset AB, Lund, Sweden.

Gibson, B. (2005): DHTML Accessibility – Solving the JavaScript Accessibility Problem. *ASSETS'05, Baltimore, Maryland, USA.*

Graham, P. (2004): Hackers & Painters: Big Ideas From The Computer Age. *O'Reilly Media, Inc., Sebastopol, CA., USA.*

Google maps (2007): http://maps.google.com

Henninger, S. (1995): A Framework For Developing Experience-Based Usability Guidelines. *DIS 95 Ann Arbor, MI, USA.*

Henninger, S. (1997): Creating Organization-Specific Usability Guidelines. CHI 97.

Horton, S. (2006): Designing Beneath the Surface of the Web. *W4A at WWW2006, Edinburgh, UK.*

Iannella, R. (1995): HyperSAM: A Management Tool for Large User Interface Guideline Sets. *SIGCHI Bulletin, Vol. 27 / No. 2, April 1995.*

iGoogle (2007): http://www.google.com/ig

Isakowitz, T. (1995): RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM, Aug. 1995 / Vol. 38, No. 8.*

ISO 9126 (2007): http://www.issco.unige.ch/projects/ewg96/node13.html

Israel, M., Hay, I. (2006): Research ethics for social scientists: between ethical conduct and regulatory compliance. *Sage, London, UK.*

Kappel, K. (2006): Developing User Interface Guidelines for DVD Menus. *CHI 2006, Montréal, Québec, Canada.*

Karnell, I. (2004): An Introduction to Rich Internet Applications: Addressing The "Usability Wreck" Of Today's Web. One to One interactive whitepaper, May 2004. http://www.onetooneinteractive.com/

Kliehm, M. (2007): Accessible Web 2.0 Applications with WAI-ARIA. http://www.alistapart.com/articles/waiaria

Kvale, S. (1996): Interviews: an introduction to qualitative research interviewing. *Sage Publications.*

Law, C. M. (2006): Are Disability-Access Guidelines Designed for Designers? Do They Need To Be? *OZCHI 2006 Proceedings, Sydney, Australia*

Lemon, G. (2006): Making Ajax Work with Screen Readers. http://juicystudio.com/article/making-ajax-work-with-screen-readers.php

Linthicum, D. (2007): Practical Enterprise Mashups. *InfoWorld, Feb. 2007.*

Loosley, C. (2006): Rich Internet Applications: Design, Measurement, and Management Challenges. *Keynote Systems 2006 Industry Brief.*

Lynch, P. J., Horton, S. (2002): Web Style Guide: Basic Design Principles for Creating Web Sites. Yale University Press; 2nd edition.

Mahlke, S. (2005): Understanding Users' Experience of Interaction. *Proceedings of the 2005 annual conference on European association of cognitive ergonomics, Chania, Greece.*

Mariage, C., Vanderdonckt, J., Pribeanu, C. (2004): State of the Art of Web Usability Guidelines. *Lawrence Erlbaum Associates, Mahwah.*

Mathes, A. (2004): Folksonomies – Cooperative Classification and Communication Through shared Metadata. *UIC Technical Report.*

Maurer, D. (2006): Usability for Rich Internet Applications. Digital Web Magazine, Feb. 2006, http://www.digital-web.com/

McNamara, N. (2005): Defining Usability: Quality of Use or Quality of Experience? *2005 IEEE International Professinal Communication Conference Proceedings.*

McMullin, J. (2003): Usability Heuristics for Rich Internet Applications. Boxes and Arrows, July 2003, *http://www.boxesandarrows.com/*

Microsoft Corporation (1995): The Windows Interface Guidelines for Software Design. *Microsoft Press, Redmond, Washington, USA.*

Microsoft Corporation (2007): Windows Sidebar Gadgets. http://msdn2.microsoft.com/en-us/library/aa974179.aspx

Mindlab (2007): http://www.mindlab.de

Mullet, K., Darrell, S. (1995): Designing Visual Interfaces – Communication Oriented Techniques. *SunSoft Press.*

National Cancer Institute (2002): Research-Based Web Design and Usability Guidelines. http://www.usability.gov/guidelines

Nielsen, J. (1994): Ten Usability Heuristics. http://www.useit.com

Nielsen, J. (1997): The Difference between Web Design and GUI Design, Jakob Nielsen's Alertbox, May 1997. http://www.useit.com/alertbox/9705a.html

Nielsen, J. (2000): Designing Web Usability: The Practice of Simplicity. *New Riders Publishing, N.Y., USA.*

Nielsen, J. (2004): The Need for Web Design Standards. Jakob Nielsen's Alertbox, September 2004. http://www.useit.com/alertbox/20040913.html

Norman, D. A. (1988): The Design Of Everyday Things. *Basic Books, N.Y., USA.*

Norman, D. A. (2006): Emotionally Centered Design. *Interactions May/Jun 2006.*

Ohnemus, K. R. (1997): Web Style Guides: Who, What, Where. *SIGDOC'97, Snowbird Utah, USA.*

OLPC: One Laptop Per Child (2007): http://www.laptop.org

O'Reilly, T. (2005): What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software.
http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

Petrie, H., Kheir, O. (2007): The Relationship between Accessibility and Usability of Websites. *CHI 2007 Proceedings, April 28-May 3, 2007, San Jose, CA, USA.*

Preciado, J. C. (2005): Necessity of Methodologies to Model Rich Internet Applications. *IEEE WSE'05 Proceedings, 2005.*

Preece, J. (1994): Human-Computer Interaction. *Addison-Wesley.*

Preece, Rogers, Sharp (2007): Interaction Design. *John Wiley & Sons, Ltd.*

Raskin (2007): http://video.google.com/videoplay?docid=-6856727143023456694

Ratner, J. (1996): Traditional vs. Web Style Guides: How do they differ? *Human factors and ergonomics conference, Philadelphia, PA, USA, 2-6 Sep 1996.*

Robertson, J. (2007): 11 Usability Principles for CMS products. *KM Column, Step Twp Designs Pty Ltd.*

Rosenzweig, E. (1996): Design Guidelines for Software Products: A Common Look and Feel or a Fantasy? *Interactions, Sep/Oct 1996.*

Scapin, D. (2000): A Framework for Organizing Web Usability Guidelines. *6th Conference on Human Factors & the Web.*

Schmeiser, L. (1997): Web Design Templates Sourcebook. *New Riders Publishing, Indianapolis, Indiana.*

SCO Inc. (1996): Motif Style Guide.
http://www.s-and-b.ru/syshlp/motif_guide/MotifStyleGuide/TOC.html

Seale, C. (1999): The quality of qualitative research. *Sage Publications, London, UK.*

Shneiderman, B. (1992): Designing the User Interface: Strategies for Effective Human-Computer Interaction. 2nd ed. *Addison-Wesley Publishing Company.*

Shneiderman, B. (2000): Universal Usability: Pushing Human-Computer Interaction Research To Empower Every Citizen. *Communications of the ACM, May 2000/Vol. 43, No.5*

Silicon Graphics, Inc. (2001): IRIX® Interactive Desktop User Interface Guidelines.
http://wwweic.eri.u-tokyo.ac.jp/computer/manual/lx/SGI_Developer/books/UI_Glines/sgi_html/index.html

Simpy (2007): http://www.simpy.com

Sinh, R. (2005): A cognitive analysis of tagging. http://www.rashmisinha.com/archives/05_09/tagging-cognitive.html

sTeam (2007): http://www.open-steam.org/

Sullivan, T. (2000): Barriers to Use: Usability and Content Accessibilitay on the Web's Most Popular Sites. *CUU'00 Arlington, VA, USA.*

Sun Microsystems, Inc. (1989): Open Look: Graphical User Interface Functional Specification. *Addison-Wesley Publishing Company, Inc.*

Sun Microsystems Inc. (2001): Java™ Look and Feel Design Guidelines (2nd Edition). *Addison-Wesley.*

Surowiecki, J. (2004): The Wisdom of Crowds: Why the Many Are Smarter Than the Few. *Abacus, London, UK.*

Tango (2007): http://tango.freedesktop.org

Theofanos, M. F. (2003): Bridging the Gap Between Usability and Accessibility. *Interactions, Nov/Dec 2003.*

Tognazzini, B. (2003): First Principles of Interaction Design. http://www.asktog.com/basics/firstPrinciples.html

Vanderdonckt, J. (1999): Development milestones towards a tool for working with guidelines. *Interacting with Computers, 1999 / Vol. 12, No. 2.*

W3C (2006): Roadmap for Accessible Rich Internet Applications (WAI-ARIA Roadmap). W3C Working Draft 20 December 2006. http://www.w3.org/TR/aria-roadmap/

Wroblewski, L. (2001): Design Considerations for Web-Based Applications. *Proceedings of the 45th Annual Meeting of the Human Factors and Ergonomics Society. Santa Monica, CA.*

Yin, R. K. (2003): Case study research: design and methods, 3rd ed., *Sage Publications, Thousand Oaks, CA.*

Appendix

Interview Guide

General Information:

Date and time: ……

Interviewer: ……

Interviewee: ……

Company: ……

Framing of the interview: Briefing (Before starting the recording).

In this part will we introduce ourselves, define the purpose of the interview and explain why we record/take notes of the interview.

- Introduction (Introduce ourselves and what we are writing about)

- Purpose of the interview?
  "In our work, we are designing a framework for Rich Internet Application guidelines. The purpose of this interview to learn about your work and what you think of existing guidelines for web development."

- Why do we record the interview?
  "The reason for recording the interview is to capture all answers you gave us as accurate as possible. The original recording is kept confidential and we will send you a written transcript if you desire."

- *"Do you have any questions before the interview starts?"*

……..

……..

……..

Questions (Recording starts)

Topic A: Background

Purpose: gain basic information about the interview subject's background

Technique:

- *Introducing questions*
- *Follow-up questions*
- *Specifying questions (if we want a more precise descriptions)*

Questions:

A1: Please tell me what is your professional background?

A2: Please tell me what kind of work do you do currently? (Web developer, web designer, …)

A3: What are the specific affordances of your work, related to the company you work for? (What is special about the company you work for?)

A4: Have you previously dealt with Rich Internet Applications? (Flash, AJAX, extensive usage of JavaScript)

A5: Did you develop desktop applications previously? If yes, what kind of work was it and what was your responsibility?

Topic B: Design/Development process

Purpose: identify general approach towards design/development

Technique:

- *Introducing questions*
- *Follow-up questions*
- *Specifying questions (if we want a more precise descriptions)*

Questions:

B1: Do you follow any design/development process? (life cycle process? i.e. for implementing new features, Bugfixing, etc?)

B2: Is there a separation between the web designers and the programmers? (How does it work?)

B3: How do you deal with the interaction design / user experience? Is there an interaction designer involved?

B4: Which tools do you use for your work (including software, hardware, literature, you name it!)

B5: Considering the literature, which tool is most useful/indispensable for your work?

Topic C: Design guidelines

Purpose: Find out if / how the interview subject works with guidelines

Technique:

- *Introducing questions*
- *Follow-up questions*
- *Specifying questions (if we want a more precise descriptions)*

C1: Do you use any kind of guidelines or checklists in your work? Why?/Why not?

C2: Describe the type of guidelines you use (usability/accessibility/corporate identity guidelines?)

C3: Can you descibe the way **how** you work with guidelines? (More as a “reference“ or more specific?)

C4: If you have to design within your corporate identity, do you think it is difficult to maintain a balance between consistency (related to other web applications) and corporate identity? How do you approach the problem?

C5: What do you think of existing guidelines for web development in general?

C6: What are advantages/disadvantages of using guidelines from your point of view?

C7: Do you miss any aspects not covered by guidelines?

C8: If you think of the **structure** of existing guidelines, is there anything particular you miss or would like to have included?

C9: Now, please think specifically about Rich Internet applications. Do you use any guidelines specifically designed for Rich Internet applications? If not, do you think that existing guidelines cover Rich Internet applications sufficiently?

C10: If you would design guidelines for rich Internet applications, how would you proceed (methodology)?

C11: What would you consider to be the most important aspects to include in RIA guidelines? Can you think of a structure how RIA should look like?

C12: Do you think that adopting the interaction component from existing desktop user interface guidelines into web guidelines is a good idea?

Framing of the interview: Debriefing (Stop recording)

In this part the subject is given an opportunity to tell if there was something he or she had on her mind but didn't want to say during the interview. In this case the comments must be written down by the interviewer (off-record).

Question:

"Now as the interview is over and we are off-record, is there anything you have on your mind and would like to add or mention"

……..

……..

……

……

Finishing:

"THANK YOU VERY MUCH FOR YOUR PARTICIPATION! Your help is greatly appreciated!"