British Journal of Mathematics & Computer Science
1(1): 16-45, 2011

SCIENCEDOMAIN *international*
www.sciencedomain.org

User Interface Design and Usability Testing of a Reinforced Concrete Design (RCD) Beam Interface

L. O. Yusuf [1*], O. Folorunso[1], A. T. Akinwale[1], I. A. Adejumobi[2]

[1]Department of Computer Science, University of Agriculture, Abeokuta, Nigeria
[2]Department of Electrical and Electronics Engineering, University of Agriculture, Abeokuta, Nigeria

Research Article

Received 23rd December 2010
Accepted 8th January 2011
Available online 31st January 2011

ABSTRACT

This paper describes the design outcomes and user evaluation of a Reinforced Concrete Design (RCD) interface for Beams from a user-centric point of view. The design began with a user interface proposal that was shown to Reinforced Concrete Designers to gather first reactions. Based on feedback the style evolved and eventually interaction diagrams were made to make sure that the functionality was both possible and logical to implement. The interaction diagrams were presented to usability RCD experts to be initially evaluated and commented upon. The design decisions were implemented using Microsoft Visual Studio 2010 and AutoCAD 2010. A prototype was produced to test the theories in practice and to gain information about the RCD users. The implementation itself took advantage of emerging technologies such as WCF which make it possible to use Service Oriented Applications through the web services. A total of twenty-five RCD experts were recruited to perform various tasks on both Structural Analysis and Design (STAAD) Software Product and the prototype RCD applications and rate their experiences. The outcomes of the user-study were positive, with the majority of the users (twenty-two out of twenty-five) preferring the prototyped interface to the existing and fully working solution. This was considered a good result given the qualitative feedback from the users. Some of the findings emphasized the importance of the holistic experience and look-interact-and-feel over a pure set of technical features and merits.

Keywords: RCD, RCD Beam, AutoCAD, WCF, user-centred design, usability, design interface, visualization mashing;

**Corresponding author: Tel. +2348034037098; E-mail. truevisionconsulting@yahoo.com*

1. INTRODUCTION

The simplest type of bending element in RC structures is a beam (Andrew and John, 2010). A beam is used to span a gap between at least two supports and provide support for slabs that will act as floors or ceilings. Beams can be manufacture from RC or Steel. A beam of any given size can only span a finite gap (Tirupathi and Ashok, 2008), before the beam's self weight causes it to fail. By altering the support conditions of a simply supported beam, this distance can be increased. By pre-stressing or pre-cambering the beam, this distance can be increased even more (Bill et al., 2007).

The usability test for RCD was conducted in a laboratory-like area environment. It is a peaceful space, where RCD testers can concentrate on the given tasks. Although, usability researchers and practitioners have been concerned that the laboratory-like setting for evaluations do not simulate the context where systems are used (Johnson, 1998) and lack the desired ecological validity like interruptions, movement, noise, multitasking etc. (Tamminen et al., 2004) that could affect the users' performance which are not present in this type of test environment. We believe this type of criticism may not be fully applicable to RCD environment because RC designer are expected to work under a very conducive atmosphere so that the best judgement may come out of their design process. Even if there seems to be a common concern about the adequateness of laboratory-like setting evaluations, field evaluations have been rather rare. A literature study by Kjeldskov and Graham (2003) revealed that most (71%) system device evaluations were done in laboratory-like settings. This may be due to data collection techniques such as think aloud, video recording or observations being difficult in the field. As RCD tools have rapidly developed during past few years, conducting user tests in the laboratory-like settings has become easier. It is now possible to attach a small camera inform of webcam to record the activities of RCD testers and collect vital information for later review (Kjeldskov et al., 2004a; Roto et al., 2004). It is possible for test leader to follow what is happening on the screen and hear users' comments. This also allows the usage of think aloud protocol in usability test. Despite the development of suitable tools testing, field testing is still likely to be more time consuming (Kjeldskov et al., 2004a) than in laboratory-like setting. It may also require extra effort from test users and the test leader.

Resources for application development are limited in the RCD industry, and usability activities such as user-centered design and usability testing must be made very efficient. The goal in a product development process is to find the biggest and most fatal usability problems within the strict limitations of project budgets and deadlines. The focus of the usability inspection is not on finding every possible detail. Decisions made by usability expert when planning usability tests are related to risk management; how to optimize the effort and the outcome (Nielsen and Landauer, 1993). Choosing the right evaluation method is important; scientifically validated information on suitable testing methods is valuable for usability practitioners. Kjeldskov, Skov and Stage (2004b) presented a good example of information practitioners' need when making decision on the method in their article Instant Data Analysis: Evaluating Usability in a Day. Hertzum (1999) compared role of three different methods (laboratory tests, workshops and field tests) in a product development cycle. His goal was also to increase the efficiency of the tests. In this study, the main question is to find out whether field tests are critical when evaluating RCD application usability or can the sufficient ecological validity be simulated in laboratory-like environment? Naturally, the usability for RCD depends on both the vendor's tool and the method via which the subscription is done. There are good implementations and there are difficult ones as well. In some cases this involves creating shortcuts to other software through interoperability. This might happen on instances where existing graphics tool like AutoCAD, Double CAD etc are borrowed to enhance our new interface for RCD design and detailing so that we don't reinvent the wheel. The design for this paper was started in the beginning of 2009. The end prototype was to concentrate to user experience and usability over technical obstacles and to see if these elements could be enhanced via emerging technologies (in RCD context) such as WCF services through the web. The possibilities in regards to usability and user experience issues are also dealt with.

The objectives of this paper is to create a user interface for RCD Beam tool that is easy to use as possible, evaluate the results and demonstrate the benefits that could be achieved by taking advantage of new technical possibilities such as Service Oriented Application through WCF services. The user interface is to be verified via user trials that test the produced prototype. The obtained user-test results will then be analyzed and suggestions for further development will be made. A special effort will be made to evaluate the pleasantness of the user interface. The paper also concentrates on the usability and user-experience factors on user interface design and proposed a new user interface for RCD Beam. Furthermore, the paper describes the user acceptance tests performed with the prototype that was created to test the interface design. The technical implementation was done using Microsoft Visual Basic Express 2010, WCF through Microsoft Visual Web Developer Express 2010 and AutoCAD 2010. WCF is quite large and very flexible (Rod, 2010). It gives the ability to write secure, reliable services that support transactions and can use a variety of transport methods. The interaction design of the prototype focused on the design of RCD Beam only other components like RCD Slab, RCD Columns, RCD Foundation etc., are not considered. Thus the methods are studied and evaluated in the context of RCD Beam only and the input/output outcome features that are available for it. Different RCD components may have other requirements and possibilities, which make this design heavily RCD Beam components dependant.

2. BACKGROUNDS

2.1 REINFORCED CONCRETE DESIGN

Much of the groundwork for this work was laid by earlier work by Yusuf et al., (2009). They visualized a simple beam by automatically generating reinforcement properties for the purpose of beam detailing. Yusuf et al., (2010) observed that user's perception was suppressed by not allowing them to use their intuition to make their choice from the steel table. They consequently modified the SSRCBS tool to incorporate visRCD Table Advisor. We also noted that visRCD Beam interface was created as input visualization environment while AutoCAD interface was enhanced and borrowed as output visualization environment. The intermediate visualization environment which is very important in RC design was neglected. We shall therefore add visRCD Beam Table Advisor as intermediate visualization environment to incorporate visualization into every step within the RCD beam process by using Information Visualization approach to check and circumvent failure at every point where steel reinforcement is required. Fady (2008) also developed tool for analysis and design of beams up to three spans, upon the input of beam parameters, the program automatically fix bar sizes. This program was created using the relatively new Action script language. The limitation of Yusuf et al., (2009 and 2010) is that their tools will not design and visualize more than one span of beam while Fady's tools will analyze, design and visualize beam up to three spans pin end condition. RCD Beams encounter in real life are usually continuous (indeterminate) of many spans usually more than one span and can probably go beyond three spans with varying end conditions. We considered both simple and continuous beams of varying end conditions with infinite number of spans. The computer memory shall be our limitation. Yusuf et al., (2009) considers only four loads (Knife edge, linearly distributed, Equilateral triangular, and Right angled triangular loads) acting on beam structures while Fady considered only two loads (Knife edge and linearly distributed loads). We believe more loads need to be considered. Example of such loads is; couple or moment load, the loads considered by Yusuf et al., (2009) spread over the entire span; we believe that loads can be on any location within the span and should not necessarily spread over the entire span. We shall elaborate on this further under methodology. Our major contribution here will be the integration of a host of techniques to create a novel application that is both usable and useful in any RCD domain using SOA approach. A service-oriented architecture for RC design is an information technology approach or strategy for RC in which RC design application tools make use of (perhaps more accurately and in a synchronized manner) rely on Data-based services available in a network such as the World Wide Web. Implementing a service-oriented architecture can involve developing applications like RC design that use services, making RC design table advisor application tools available as services so that other RC applications can use those services. Folorunso et al., 2010 described in their paper ("SOA-RTDBS: A service oriented architecture (SOA) supporting real time database systems") how SOA can support RTDBS, their approach was highly theoretical; no actual implementation was carried out for a specific real-time database problem. We shall adopt their approach to implement SOA for RCD. Serviceability limit state for RCD will be exposed as a service through RCD table advisor.

2.2 SERVICE ORIENTED APPLICATION

Service-Oriented Architecture (SOA) is one of the hottest topics that is currently gaining momentum, its adopters (both business and IT executives) is increasing in a tremendous manner (Qusay, 2009). SOA represents a new paradigm that reflects a leap transition in both computing and software industries (Tsai et al., 2006). It has emerged after decades of using distributed computing technologies to add a new element to software stack. According to Mike et al., (2008) SOA is an architectural style for building enterprise solutions based on services. More specifically, SOA is concerned with the independent construction of business-aligned services that can be combined into meaningful, higher-level business processes and solutions within the context of the enterprise. The real value of SOA comes when reusable services are combined to create agile, flexible, business processes. Unfortunately, that does not just happen by itself. Achieving it might be easier to manage if a single organization is creating all of the services, but that is not the case at most large organizations. So, part of the architecture of SOA is responsible for creating the environment necessary to create and use composable services across the enterprise. SOA system can reduce development costs, result in higher quality of the design of the systems, and consequently yield higher reliability (Mike et al., 2008). In this work, we propose SOA as a new approach to building RCD Beam that allows RCD businesses to leverage existing assets and easily enable the inevitable changes required to support the RCD business. One of the most important aspects of SOA is that it is a *business*, a *technological* as well as methodological approach (Judith et al., 2007). SOA enables businesses to make business decisions *supported* by technology instead of making business decisions *determined* by or *constrained* by technology. And with SOA, the folks in RCD community finally get to say "yes" more often than they say "no." One of the biggest deals in the SOA world is the idea that things are not thrown away, the best of software assets used every day is packaged in a way that allow for use, reuse and keep on reusing it securely in the knowledge that future changes will be simple, straightforward, safe, and fast. This makes system less complicated and less expensive to maintain. Mike et al. (2008) described SOA as the careful balance and blending of the big picture and the immediate requirements to the practical application of theory to meet a set of goals in the present and in the future. Services are at the core of SOA. A service in SOA is. Just like the definition of SOA, defining the term service is not an easy task (Perrey and Lycett, 2003). The simplest idea is that a service performs a (reusable) function. This function can be anything from simple retrieval of data to performing a whole business process (Papazoglou, 2003) and Papazoglou and van, 2007). The services in SOA always have a business aspect to them instead of a more technical aspect. An example of a service is get 'check for RCD Beam deflection ', opposed to upload file, which is not a business service (Krafzig et al., 2004). A service is defined by (OASIS, 2006) as "the performance of work (a function) by one for another". This definition is related to the following ideas: capability to perform work for another, specification of the work offered for another and offer to perform work for another

2.3 VISUALIZATION

The human perceptual system is highly attuned to images, and visual representations can communicate some kinds of information more rapidly and effectively than text. For example, the familiar bar chart or line graph can be much more evocative of the underlying data than the corresponding table of numbers (Larkin and Simon, 1987a). The goal of information visualization is to translate abstract information into a visual form that provides new insight about that information. Visualization has been shown to be successful at providing insight about data for a wide range of tasks. The field of information visualization is a vibrant one, with hundreds of innovative ideas burgeoning on texts and images. However, applying visualization to RC design information is quite challenging, especially when the goal is to improve human perceptual activities relating to graphics over text collections. Graphics is a means towards some other end, rather than a goal in itself. When reading text, one is focused on that task; it is not possible to read and visually perceive something else at the same time. Furthermore, the nature of text makes it difficult to convert it to a visual analogue. Most likely for these reasons, applications of visualization to RC design have not been widely accepted to date, and few usability results are positive. For example, Chen and Yu, 2000 conducted a meta-analysis of information visualization usability studies, with a focus on information retrieval problems. The purpose of a meta-analysis is to combine many different points in the evaluation space in order to come up with more robust and general results. They focused on six visualization interface studies from five papers (Robertson et al., 1998, Allen, 2000, Sebrechts et al., 1999, Swan and Allan, 1998, Combs and Bederson, 1999).

Guidelines for designing information visualizations are available from writers such as Few (Few, 2006, Few, 2009) and Tufte (Tufte, 1983, Tufte, 1990b). Some of these guidelines overlap with guidelines from graphic design, including the need to present information clearly, precisely, and without extraneous or distracting clutter. Other guidelines relate to the special purposes of visualization. Good visualizations use graphics to organize information, highlight important information, allow for visual comparisons, and reveal patterns, trends, and outliers in the data. Visualization guidelines are also derived from principles of human perception, and urge the designer to be aware of the perceptual properties which can affect the design (Few, 2006).

2.4 USER EXPERIENCE

2.4.1 Usability in RCD perspective

Usability testing is a common tool used to evaluate the usability of RCD application in a development process. Usability tests are usually conducted using a think aloud protocol based on Ericsson and Simon's work (1980, 1984). Users are given tasks in a test environment and encouraged to think aloud while trying to accomplish the tasks. This gives us, usability practitioners, information we need on how the user interface matches the natural human way of thinking and acting and highlights the features and processes to be improved. Severity of the usability problems is an important factor when defining the urgency of actions related to RCD problem. The most urgent actions are needed when the problem prevents completion of RCD task. Dumas and Redish (1993) use four point scale, where the first severity level represents the most severe problems and the last the least severe. Also Kallio and kekelainen (2004) divide the severity of problems into categories; high (failure in task execution), medium (not so severe, task can be executed) and low (minor problems).

Usability is a term used to denote the ease of using an object, whether physical or digital, to reach a goal of the person performing the task. Usability is defined by ISO 9-241-11 as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". As much as this is a standard, it does not really mean anything on its own. Usability can be divided into two relevant categories; the physical usability and the usability of the software. Physical usability deals with interaction devices, design and ergonomics. These are of great importance when it comes to the appeal of the device. It should be noted that physical usability is in some ways related to software usability, as a study about mobile web design describes (Trewin 2006). This is only natural, as the software is not used directly but via physical interface devices such as keyboards, joysticks, mice and displays.

The study of usability has also lead to various methodologies to measure how well the users can cope with the tools. These can be further divided into qualitative and quantitative methods. Quantitative methods typically measure aspects such as the performance of how the users perform various tasks or ask the test users to rate their opinions with numerical values. Qualitative methods, on the other hand, try to gain insight into the users by seeking answers to questions such as why the users perform the tasks they do in the first place. This data can be obtained for example via interviews, questionnaires or observations. The methods can be used together to provide measurable qualitative results and use the qualitative information to gain insight. It is suggested that there are situations where some other needs compensate the lack of usability and quality of the solution (Reponen et al., 2006). Hence, there is definitely more to the whole satisfaction of the user than from strict usability. Even Jacob Nielsen notes that other aspects make systems such as web pages pleasing even if he does not describe them per se (Nielsen 2000).

2.4.2 User experience

To create a tool that will attract and be accepted by a number of RCD analysts, considerations dealing with usability alone are not enough. We need to take into account the holistic user experience, which, unfortunately is rather vague term. The user experience research is still not well understood and not well defined (Roto, 2006). The very definition of "user experience" is controversial. The following is a list of some suggested definitions:

"Every aspect of the user's interaction with a product, service, or company that make up the user's perceptions of the whole; user experience design as a discipline is concerned with all the elements that together make up that interface, including layout, visual design, text, brand, sound, and interaction. UE (user experience) works to coordinate these elements to allow for the best possible interaction by users." (UPA, 2007). Makela and Fulton (2001) also define UE as a result of motivated action in a certain context.

Hassenzahl & Tractinsky (2006) described it as a consequence of a user's internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of the designed system (e.g. complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (e.g. organizational/social setting, meaningfulness of the activity, if the use is voluntary, etc.). Alben (1996) see UE as all the aspects of how people use an interactive product: the way it feels in their hands, how well they understand how it works, how they feel about it while they're using it, how well it serves their purposes, and how well it fits into the entire context in which they are using it.

In essence, user experience means how the user feels about using the product. Naturally this depends on the individual himself as well as his experiences in the past. His location and surroundings may affect how he deals with the task he wants to perform with his tool. His expectations also matter. In fact, next to everything is likely to give some degree of "experience", whether good or bad, to the feeling of the user. For example, Mäkelä and Fulton (2001) define the user experience to consist of previous experiences and expectations that the user has towards the system he is going to use. The user has a motivation to use the new system and he makes an action by using it in a context (to be understood rather vaguely, as "on lunch break" for example or "finding commuting routes"). Motivation, action and context form the present experience at the time of the use. The present experience then moulds the future experiences and the expectations.

A common problem with user experience results thus far has been that the abstraction level is high. It is easy to refer user experience as something that is resulting from the mental state of the end user, his ambitions, lifestyle, location, if it is raining or if he is going to miss his last bus in the evening. This level of abstraction is insufficient for designers to actually help to improve the whole experience systematically and in a formal manner. It does help, however, as a mental model. Even evaluation of the user experience is not straightforward. It is highly dependent on when and how the evaluation is done. If we ask if a product is pleasing to the user right after he has unpacked it from the shiny covers, he is likely to give a different response than after he has used it a year and the device has just died without a warning.

Roto (2006) proposes that the term “user experience” would be narrowed down to mean the interaction between the person and a machine. The rest, for example, a person viewing a painting in a gallery, should be called simply “experience”. While I consider that this definition raises an immediate need to define a more holistic term to replace what “user-experience” used to mean (e.g. everything), this definition of interaction makes the problem area more manageable. While doing so, if one were to hold only to this definition we would omit the possibility to affect this very user-experience with pre- and post-use means (these include marketing and advertising which try to affect the expectations user has towards the product, packaging, taking care of the recycling to name some of the available options). I postulate that a product or software user-experience cannot be done as a black-box; yet, while engineering the software, one is likely to have a limited number of options to affect the rest of the process. This paper will use the term in the same scope as Roto presented in her doctoral paper (Roto, 2006).

2.4.3 Emotional communication

Only about 45% of the feelings and attitude messages shared between people happen by speaking (Mizutani 2006). Actions, such as shaking hands deliver vast amount of information via gestures such as the firmness of the shake and if the partner is smiling or being serious. In fact, Mehrabian proposed a theory that in each face to face communication situation that deals with emotional messages 7% of the message comes from the words themselves, 38% comes from the tone of the words and 55% comes from the body language of the speaker (Mehrabian 1981). If the above-mentioned messages differ, Mehrabian states that the most powerful ones dominate the message and rule how it is interpreted. Thus, to create a strong message each of the elements need to support each other. It should be emphasized that Mehrabian only stated the relative importance to apply when the messages dealt with feelings and attitudes. The rule he proposed was not meant to be generalized for any form of communication. However, if it is so that in emotional communication only 7% of the message is delivered by the words, it puts the communication via technical means into an odd situation. Via a voice call, we can still hear the tone of the speaker’s voice, but when that is taken away, the means to express emotion are relatively thin. Perhaps this has influenced the usage of *emoticons* (often better known as smiley) in text-based mediums such as IRC and instant messaging applications like Microsoft Messenger.

2.4.4 Theory of broken perfection

While the messages that pass between people are undoubtedly important to create an emotional contact, there are other aspects as well that affect how we perceive things.

Some objects have appeal to people because they stand out from the rest. This is particularly notable in regards to fashion industry, but is no limited to it. Some software have also created an avid audience by (or, one could argue, despite) providing unique interfaces that stand out from the norm. Blender could be considered one that differs greatly from the dominating 3D software line-up (e.g. Autodesk Maya, Avid XSI and SideEffects Houdini). Another example is Pixologic ZBrush, which too introduces a new interface style for sculpting 3D surfaces (in contrast to Skymatters Mudbox and Nevercenters Silo, for instance). If one designs what he wants in detail, there is a good chance one might end up with something that he does not like in the long run (assuming that the designer is the user as well). Knowing every little detail and being able to predict the future behaviour of an object can take part of the fun out of it. For example, Saabs (Swedish automobile manufacturer) placement of the keys next to the hand brake and not in the steering wheel like most other car manufacturers has decided to do. Mac OS X introduced the small window closing, minimizing and resizing icons that are on the “wrong” side of the window; on the left while on most other window managers they reside on the right. All these little quirks make the things different and often more attractive. It could even be thought that such personalities together create the very soul of a product. Together with the appeal to a certain set of people comes the risk of alienating a large (or even largest) group of users. There is no denying that a good percentage of the population is rather conservative and feels intimidated when things don’t work exactly as they have used to. In general, people have a tendency towards resisting change, or preference towards familiar systems (Butler 1996, Yusuf et al., 2010). Such behaviour, however, should not be taken too strictly. When cars emerged, people were rather suspicious towards them as well. For example, there were strict speed limits and a person was required to walk in front of the car with a red flag to warn others of the vehicle approaching them. If we do not challenge people to think differently, little progress can be made. Still it needs to be remembered that there is a fine line and possible penalty to pay when walking your own roads too much. To emphasize; this theory does not mean that usability should be neglected nor does it mean that usability has no value. Contrary, the usability has immense value for the end user, and this theory only claims that it is not the single aspect end-users care about. Quirks should not contradict usability and traditions too much; little might be allowed, but if they destroy the users’ ability to use the device or program there is little benefit from being different for the mere sake of it.

3. RCD BEAM APPLICATION

3.1 GENERAL DESIGN ISSUES

Software design has gone through various methodologies as well. Early on, personal computers were limited in what they could do with reasonable computational and programming effort. Simultaneously the software development was a relatively young profession, and it should be noted that personal computers have existed for only a relatively short period of time and the history of non mechanical computers is not very long either.

One of the approaches taken has been called system design and it is mentioned here as an example which can be considered as an almost opposite methodology to user centered design (in practice these methodologies need not to be mutually exclusive, however). In system design, the general approach has been to identify the system requirements and to design software that fulfils those targets. In a traditional sense those targets are set so that the system technically works but may or may not deal with the concepts of how the information is organized and presented to the actual users on the interface level.

At some point it was noted that fulfilling the technical needs was not enough. Users could not use the software easily despite the fact that it worked as specified. There were many causes for this. Sometimes the terminology used by the software was different from the one the RCD analysts used themselves. At times the most commonly used features were hidden under deep menu structures where the less used functions occupied screen estate. The interface might not have offered affordances and would require heavy use of a thick manual. Many books are written on the subject, such as “Emotional Design: Why We Love (or Hate) Everyday Things” (Donald, 2003), “The Design of Everyday Things” (Donald, 2002) and “Usability Engineering” (Nielsen, 1993). The situation eventually led to a methodology that is known as “user centered design”, though this is not to say that it is better than system design approach. In user centered design the process (or philosophy) is driven from the perspective of the user. User centered design emphasizes the involvement of the users early or in the design and evaluation phases of products (UPA 2007) and thus the product is not defined by a list of technical requirements alone. However, many methods that fall under the umbrella term “user centered design” rely on involving the users in the actual design of the product.

Some of the methodologies that rely on user input are listed here; participatory design, contextual design, co-operative design to name a few. Each of them is at least a semi-formal methodology that can be followed to help achieve a pleasing design. The difference between system design and user-centered design is not as black and white as it might seem from the earlier description. Usability factors can be taken into account even if the system is made from technical requirements and the interface designs might as well be separated component designed independently from the rest of the system. It should also be noted that no methodology developed thus far guarantees a pleasing result. In my opinion the major benefit gained from the formal methodologies is that they make it easier to avoid pitfalls that affect the user experience negatively.

3.2 DESIGN DRIVERS

Designing RCD Beam tool interface is rarely easy and it is nearly impossible to please all of the RCD audience. RCD analysts have different immediate needs for the same RCD Beam tool and not only do those needs differ from an analyst to another but they can do so even in the context of a single individual analyst.

The RCD users may even have different roles, or identities, based on where, when and why they use the tool. For example, a user may want the tool for analysis of beam only while another may engage the tool for a full design and detailing using RCD Serviceability service from a remote server or through the World Wide Web. As the RCD Beam tool are mostly designed to fulfill a purpose (whatever that purpose might be) and often one of the goals in development is user satisfaction, the solutions need to rely on decisions. These decisions are derived from what is called design drivers.

Design drivers are a set of assumptions and motivations that can be used as a tool to make decisions. They usually describe target audience and set goals that the design should fulfill. Some of these can be political in nature (e.g. “we wish to encourage every human above the age of ten to use RCD Beam tool”) whereas some may be set by the market research (“it seems that there is a demand for a RCD service that analyze and design any kind of RC Beam with any type of established load or load combinations”) or the user segment to which the product is to be targeted (“RCD analysts who have chosen structural designs and detailing of structures as a trade”). Factors such as estimated price (for production or the street price) and manufacturing can set their own limitations that may end up affecting the design. The eventual designs can be evaluated later against the design drivers by studying if the outcome can be used in the role it was specified. Evaluations can be done with the representatives of the defined target audience, which gives an indication if the solution is right for their specific usage.

3.2.1 Role of the application

The key issue for this RCD Beam prototype was to emphasize the use of RCD table advisor to pick reinforcement from the steel table which also check for serviceability and to monitor the messages delivered via the AutoCAD environment. The message, in fact, is that of monitoring the inputs for different type of beam end conditions, various types of loads and output for the bending moment, shear force and beam detail diagrams including explanatory text labels.

However, RCD users have to learn and master a challenging set of skills in RC analysis and designs to be able to enter valid input in other to produce content that delivers right messages in the way it is intended and tampering with that content would require a skill in the manipulation of drawings in AutoCAD environment. RCD table advisor expose RCD serviceability services (i.e., check for minimum and maximum steel, check for deflection) through messages from the remote server or through the internet. Also, the user might expect to get the message they subscribed to if and only if the service is available from the service provider. Figure 1 shows a message when the service is not available to the client.

The message is delivered through proxy. Alteration can be made to the drawings using the modify tools in the AutoCAD environment. Examples of such alterations would be using zooming tool to view context + details. In case of detailing, we might think of enlarging parts of the image to highlight hidden features in the hope to make it better stand out on screen.

Also, it would be naive to imagine that our preferences would apply to every RCD users. But, tampering with the message content is highly feasible or desirable and a plus to user centered design approach. The RCD properties and Beam Loading interface which was designed as a dialog boxes also help the user to pick valid input (Figure 2); user selects what he wants through the combo or list box. Jakob Nielsen states similar ideas in his book “Designing Web Usability” (Nielsen 2000). Speaking specifically about web pages he states that the emphasis is to be put on the content instead of various other aspects such as site navigation. The RCD Beam tool interface needs not to be too sophisticated, however. It should be responsive and concentrate on providing the content quickly and easily. It should also make the needed features pleasurable to use.

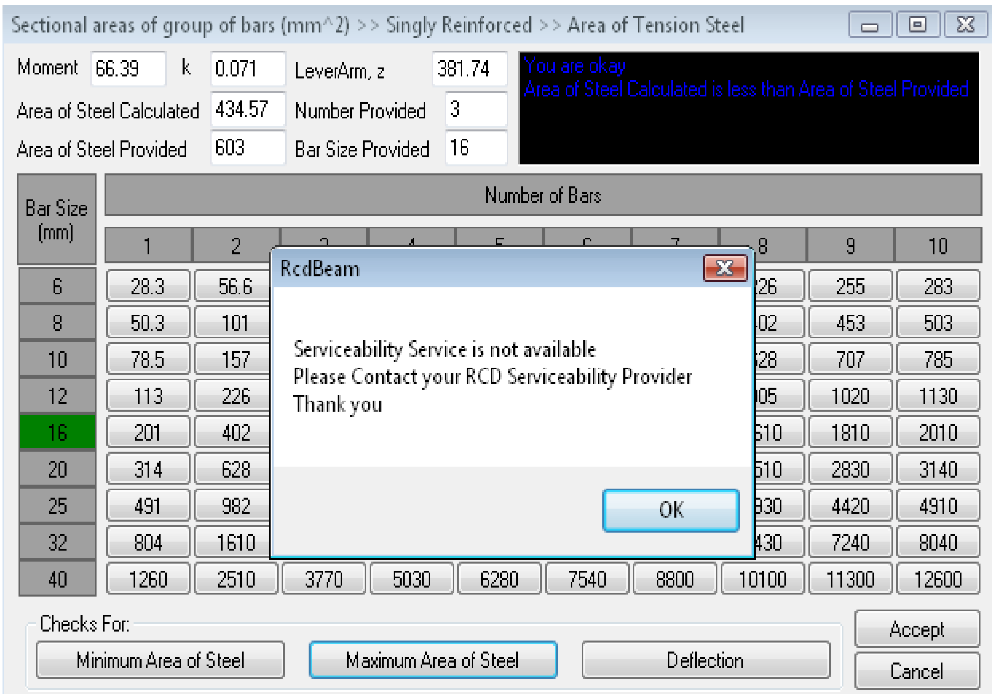

Figure 1: Message indicating that the service is not available when Maximum Area of Steel button is clicked

3.2.2 Targeted user group

As the RCD Beam tool will be a relatively new and growing phenomenon in the software market which is yet to fully mature, the emphasis of this study was to make the consumption of RCD Beam tool more appealing to RCD analysts. In practice, this means civil engineering undergraduates and graduates, architects, builders etc., people who may not have even undergone training in structural analysis and design but still might enjoy the content of RCD Beam tool without delving themselves too deep into the details of how the whole RCD Beam tool system actually works.

The preferred age group would be from the young adults to those in their late thirties. It is quite likely that the actual audience varies much more, but this age distribution was assumed to be potentially the most interesting one to take the new delivery mechanism into use. The gender of the audience was also considered to be a factor since there are more male in the civil engineering profession than the female in the third world countries like Nigeria. It was also very clear that requirements to use the prototype caused even further segmentation in the user group. For the prototype to work; it was needed to run on a system with .Net platform and AutoCAD 2010 or later is installed. The reasons for this were mainly practical like the availability of hardware and system software, the technical resources and feasibility of the development. There is no particular reason why the same concepts and even interface would not apply to other platforms as well.

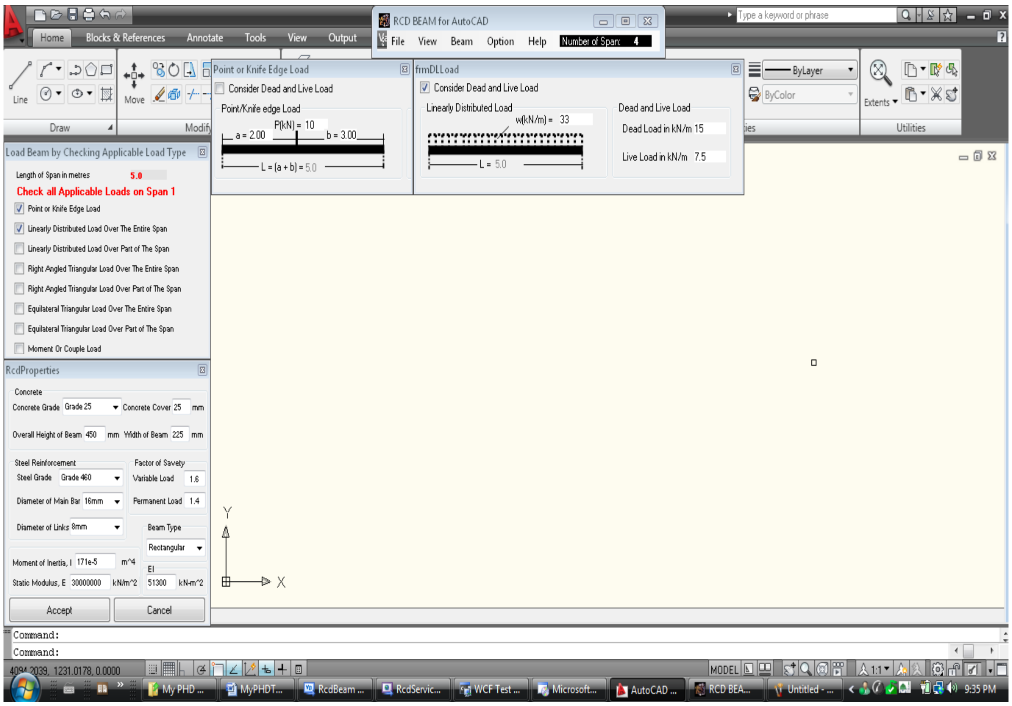

Figure 2: Beam Loading Interface (A) and RCD Properties Dialogue Boxes (B)

Users with hearing disabilities were accommodated in the design due to the nature of content itself. The visually impaired were not accommodated in the design to any great extent either. Effort was put to make the text and descriptions readable, but the interactions with the tool were expected to be done via available controls (i.e., textbox, list box, combo box, buttons etc.,) and the keyboard instead of more specialized methods such as Braille terminals. While technologies such as speech recognition and text-to-speech were in vogue, no implementation effort was made to incorporate them.

3.2.3 Appearing simple

While it could be debated whether an application should be simple or not, one of the basic assumptions behind this work was to make the RCD Beam tool interface client to at least appear as such. The emphasis was especially on the proficiency and reliability usage of the tool. The means to reach this aim were to use fewer key presses, less buttons, less navigation, not to resort to lengthy options menus as done commonly in the existing RCD tools found in the market. Also, features that the user does not really need to care for were to be put away from places where they otherwise would get the immediate attention. The paradigm of “simple use” ought not to be mixed with the concept of having fewer features; due to the scope of the application many features present on other RCD tools would likely to be needed, but that was not a driving motivation as such. The user interface was to hide unnecessary details from the end users, when the more relevant information could be. As it is difficult to know what the users expect from the end product and what the actual, relevant information that the users expect is, this work tries to offer a sensible guess and see how close of a match it will be with the feedback from the users. Such guesses were also to be tested in the user trials.

3.2.4 Assisting features

As the main motivation for the users should be to interact with the input in other to access the output content, the ease of use should be specifically considered and help provided where possible.

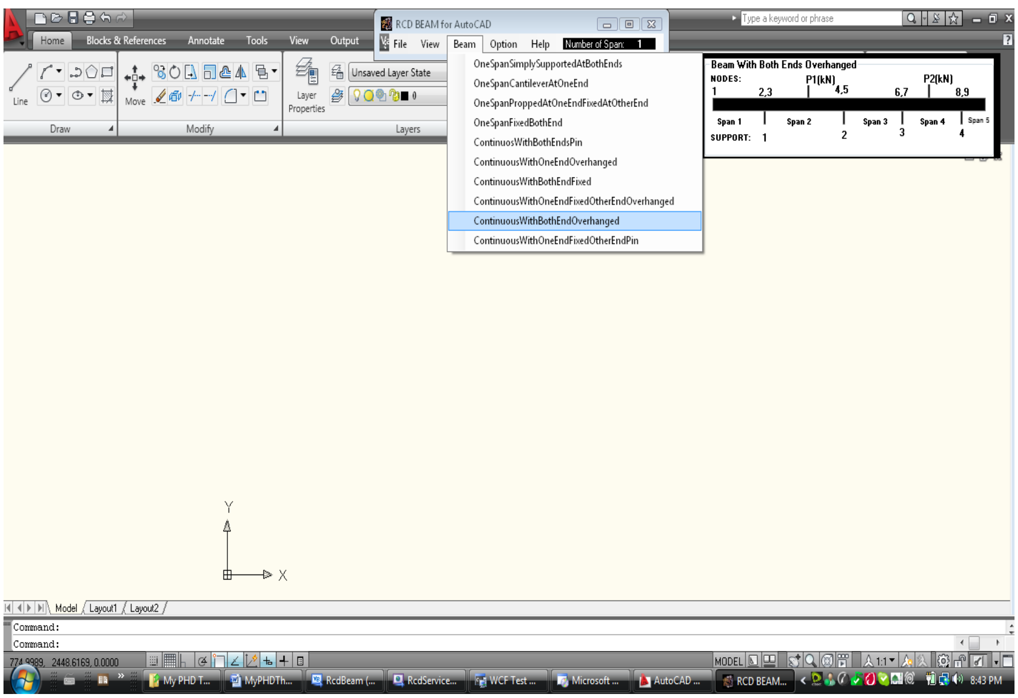

Figure 3: Animated List Control showing different types of beams with corresponding movie on the Right Hand Side

As many as possible of these assisting features should not interrupt with the flow of the tool. In practice, this would mean that the users would not need to stop to wonder what happened and that some questions would be clarified even before the need to ask about them would arise. Since it was likely that the interface would use animated elements, special attention was put to describe the requirements for such animations. Animations happen over time, and thus require a beginning and an end separated by some finite amount of time. This time is by its nature time taken away from doing something else that could take place instead. Users are put to wait for the tool, even if the time interval is minimal. Negative effect can be seen, for example, on interfaces where animations are used on list controls (Figure 3). The highlight is moved to next or previous one with a smooth animation as can be experienced, for example, different types of beams with various end conditions were animated. Animations should be relatively quick. A common problem with animation in user interfaces is that they hinder the usability by being too slow and making the user to wait as he is doing repetitive tasks. In these cases, such as list controls, the use of animation ought to be minimized.

3.2.5 Assumed Use-Cases

The main benefit of RCD Beam is the fact that it can handle any type of beam with any established known loading combinations (Figure 4).

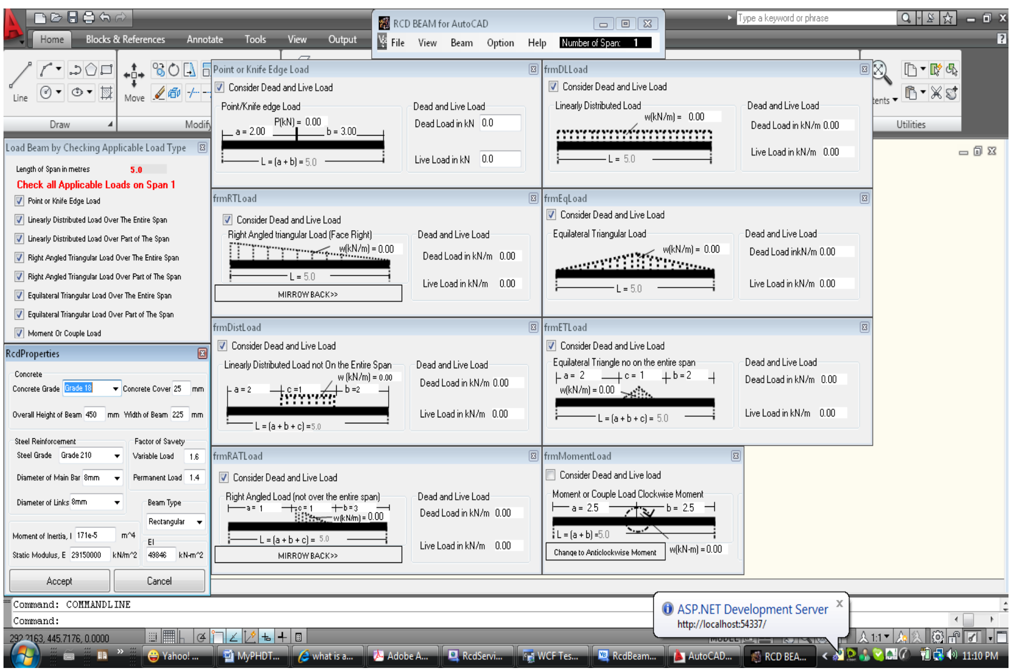

Figure 4: RCD Beam with known (common) Load Combinations

It addressed the needs of the RCD analysts as far as beam design is concerned. It is available most of the times when an analyst needs it if not hindered by RCD serviceability service provider (e.g. the tool may not be able to check for serviceability if the service is not available from the remote server).

This paper is mostly interested in the consumption aspects of RCD Beam by RCD analysts. The hypothesis is that the RCD users will subscribe to the tool for their routine analysis and design activities. In this case consumption means the use of RCD Beam tool to design RC Beams. It can include both determinate and indeterminate beam structures. The act of using the tool is the deciding factor when deciding if something has been consumed or not.

3.2.6 Product life-cycle

Products do not come from nowhere nor do they last forever. The result is that the products typically have a life cycle that is shorter than their owners and thus this section outlines the simplistic cases the product goes through in the hands of its user. While many of these stages deal with aspects that are out of reach for a single application developer to affect, they must be analyzed in order to understand the user and the implications the chosen courses of action may cause. For the user to start using a product, he must at first get it from somewhere. This can happen via a retail channel or free download through the web, but the actual methods may vary greatly. As a result, the user must somehow know about the product and something must have caused interest in the user.

The RCD Beam client tool is, ideally, installed software on a product. The actual product, a desktop or laptop computer and probably a mobile phone, is not assumed to be bought because of this single application but for the other benefits it provides to its users. Thus one obstacle to the adoption of the application is circumvented and the likelihood that users start to take advantage of the application is increased, as there is no separate process of downloading and installing the application. After the user finds and installed the application, he may actually launch it for the very first time. This also defines the initial experience and communicates further with the user what the application is used for. This may not be self-evident from the name of the application or even from the possible download pages that the user has gone through (in case he would have needed to install it through the web), even if on this occasion he is likely to have some idea why he got the application in the first place. But for an application that is installed on the platform by default, special attention is needed to make it clear what the application is and what it is used for. A common approach is to have some pre-determined content, which the user can browse from the very beginning. The content may not be to the users liking, but at least it is likely to give indication of the proposed usage. Some software products also use introductory screens that tell about the program and introduce the features to the users (including Adobe Photoshop, for example). As the user starts to use the software more often, the situation changes considerably. The instructions on how to use the software is needed less as they learn from the previous times that they have used it (hopefully, at least). In fact, what might have been a beneficial feature early on in their usage may now become a usability issue. In the case of a welcoming screen, the advanced user might already remember what it says but he still is required to go through it or inactivate it manually (again, as is done with Adobe Photoshop). Also, as time goes by, it is to be hoped that the users select content that they personally like, and thus they can form an understanding of what will be on the device. They can even anticipate when the updates arrive. Assisting features such as animations help not as much and can become an obstacle, as the user knows where he is going to and tries to get there quickly.

In the end, the user will stop using the product and the application. It may be that he has lost the interest to continue the software or that the application no longer meets his needs due to a change in his life. Again, the reasons to abandon a product are many, but eventually such a moment arrives. What can be done is to take the positive alternatives into account and make the parting from the product as pleasant as possible. In the scope of RCD Beam client tool, this could mean that the move to a new version or even alternative application is made easy; the user subscriptions are transferred to a new device without hassle together with the possible downloaded episodes and information of what he has and has not listened to.

4. IMPLEMENTATION

4.1 INITIAL PROTOTYPES

4.1.1 Initial feed-reader enhanced to RCD Beam tool

The very first attempt to create RCD Beam tool took place during the last quarter of 2009. The design was based on an existing SSRCBS by Yusuf et al. (2009) and this was enhanced to deal with other types of determinate and indeterminate beams. Also, support for serviceability limit state through the subscription to RCD serviceability service from a remote server was included.

The implementation then went to user trials in June 2010 and several usability issues were identified. The initial interface for loading beams in which different types of loads were accessed on a single form using the scroll bar was rejected by the users because it does not mimic the way RCD analysts carry out routine design work. They believe a load should not be visible when it is not needed. Various forms are therefore created for different loading system which will only be visible when the user request for it by clicking the option button for different loading systems. This was considered a novel approach by the users.

4.1.2 Paper prototypes & interaction diagrams

One of the methods to get early feedback is to have the designs evaluated with paper prototypes. The states, or views, of the application being developed are drawn to the paper and the uses-cases are gone through with the test-users. Paper prototypes are often used as a way to communicate with end-users and to emphasize that what they see is a draft and that making changes to the views is easy. This is said to make the commenting more relaxed and informal (Grady, 2000) as the users would express their opinions about the prototype more easily. Seeing that changes to paper prototypes would be easy to make, users would not feel bad about suggesting changes to the existing solutions done by the developers. Also, being able to make rapid changes without a new round of evaluations can speed up the development, it has, however, been noted that paper prototypes may have a negative effect as well (Lim et al., 2006). Given the nature of possibly unpolished drawings, some people may not take the situation seriously enough to give valuable comments as they otherwise might do. Also, some users seem to have difficulties evaluating the draft level paper prototypes (Rudd et al., 1996). The attention may drift to details that are inessential to the prototype at hands. Further, when the paper prototypes are produced by hand, the screen estate is no longer given and specifics such as the size of the writing may cause serious problems later on when the drafted designs are modified to fit the screen. In this instance, the paper prototypes were not tested as described above. Instead an interaction diagram was produced. This diagram visualized the interaction between the user and the software by connecting the key presses to the state changes within the application. Such a diagram helps to make sure that there are no lapses in the planned interaction and that the navigation remains logical and consistent. Further benefits include the easiness of explaining the design to others; thus it serves as a communication tool. However, this may not be a suitable tool to be presented to casual test users.

The actual interaction diagram was made via printed screenshots that resembled the drafted looks of the planned application. The reasons for this were many. First, it does give more concrete feedback about what is possible to fit on the screen and remain understandable. The drafting with an image-editing program was also efficient given the authors' skills with the selected program. And lastly, usability experts who were used to give feedback on prototypes even with more polished interfaces evaluated the draft.

4.1.3 Expert evaluations and changes to the final design

The interaction diagram was presented to a selected set of RCD usability experts and developers by taking printouts of the screenshots and collecting them to a large sheet of paper. Eventually the pieces were connected to each other by coloured marker pens to show the results of the key presses and navigation in each state of the software. The flow of the application was verbally explained and the evaluators could ask details where needed, even to the extent to follow through their own attempted use of the software. In fact, the full interface functionality could be tested. The outcome of the session had the greatest effect on the design of the tool, and uncovered the importance to show information as early as possible. Suggestions also dealt with the type of information present on the screen. Modifications were made to the actual prototype implementation where possible and when considered sensible. The original suggestion to enlarge the selected list elements for type of beams and show additional information that give users the idea of beam type with the capability of animation was implemented.

4.2 DRAFT IMPLEMENTATION

4.2.1 Color scheme development

The size and limitations of the target audience were considered while implementing the color scheme for the application. As the target audience was to be considered relatively at their middle age (at or below their near-thirties) a less conservative approach was taken. The first attempt visualized the RCD table advisor form display as rectangular black screen with white, blue or red colour display fore-colour depending on the message circumstance. All buttons are grey with black fore-colour; this was to imitate the appearance of successful tools in its category. While the general feedback was relatively good in regards of the looks, the pleasantness divided the peer opinions. The opinions were gathered from a group of 25 people to test their initial reactions (of which only five were females). The peer review of the changes was encouraging and thus the overall look remained to the actual user trials. The trials were expected to reveal the general acceptance of the user interface and see how the color scheme affected the user's opinions (Figure 6).

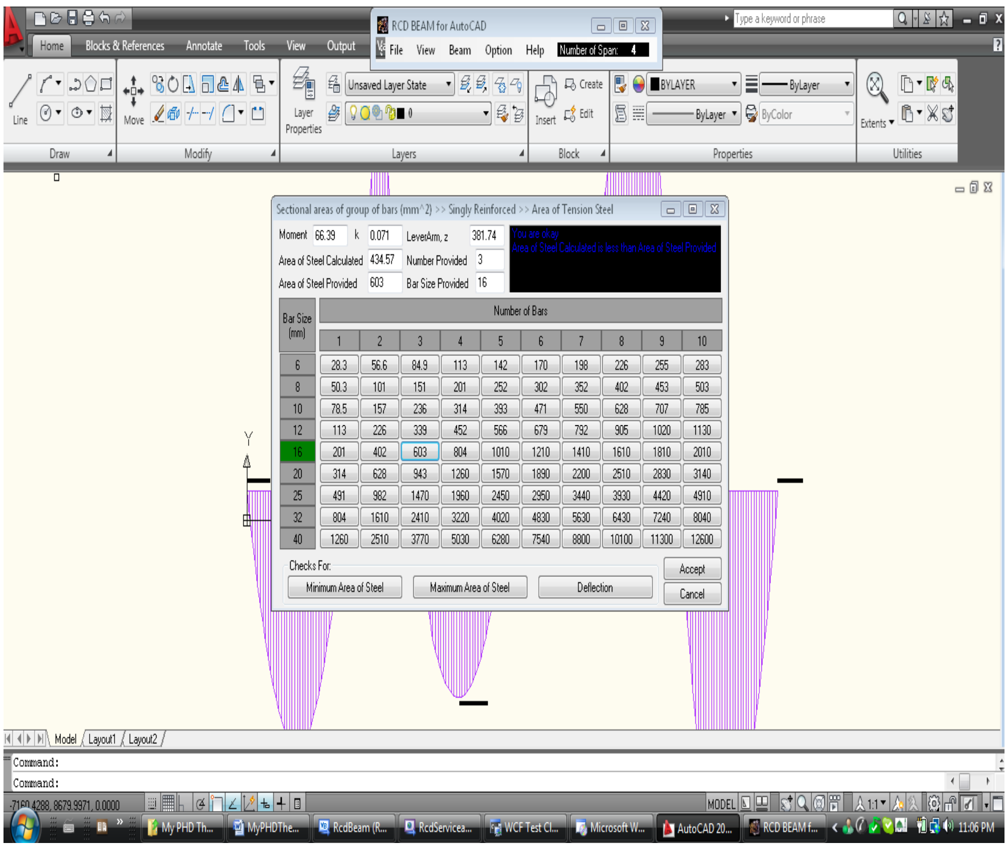

Figure 6 presents a screenshot from the eventual implementation. The RCD Beam for AutoCAD interface is located at the top centre of the AutoCAD interface. The interface has the Beam menu that houses the different types of beam and the option menu where different types of load can be activated. The bending moment diagram is automatically generated after the analysis. The RCD Table do the design and check for serviceability using services.

5. USER TESTS

5.1 USER SELECTION

RCD test users were recruited from various civil engineering based organizations, though not all the desired criteria were met; for example the gender distribution was not as even as was hoped; twenty out of the twenty-five participants were men. Also while each of the participants was expected to have some working knowledge on RCD, this proved to be in some occasions rather superficial and even the basic principles of RCD has been forgotten by the majority of the participants. This was due to the fact that many have never worked in the design office after their graduation from the polytechnic or university. Their major occupation has been in construction industries where they (most often) transfer drawings to physical objects. Use of computer was at times causing unnecessary problems in regards to the aims of this particular user test.

Each participant was also asked to have experience on the use of internet. In the end, ten participants stated they had experience in browsing. The requirement is necessary since users are expected to find RCD serviceability service from the internet. The participants were not technologically oriented in the midst of finding services on the internet with a few exceptions. Five out of the twenty-five had considerable technical background whereas the remaining twenty were not as smooth with finding services. The age distribution was rather wide, averaging about 36 years. The youngest of the participants was 25 years old while the oldest had turned 50. The exact ages of each individual user are found on figure 7.

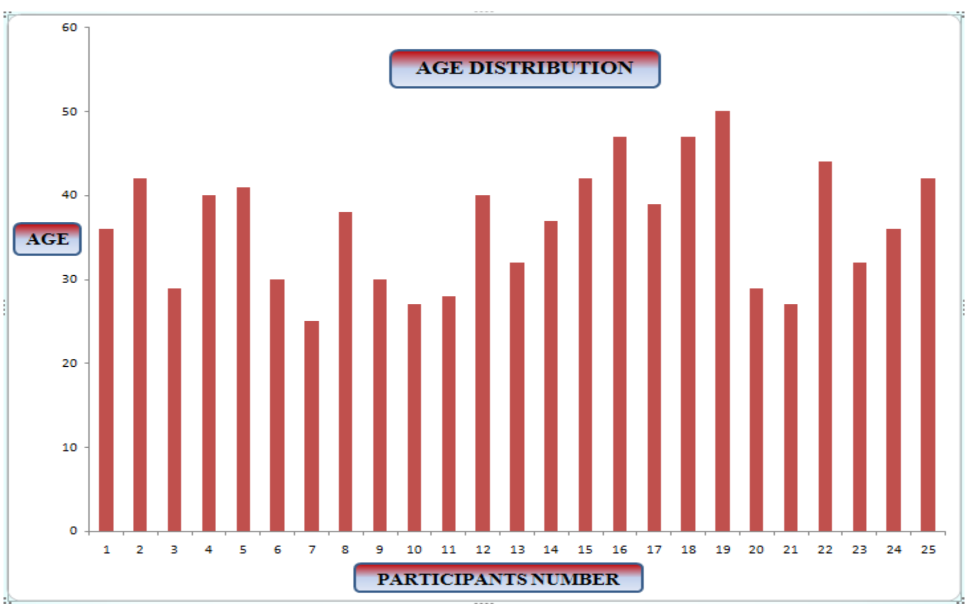

Figure 7: The exact ages of the users

In general, the users were older than in the designated target group, in some cases considerably older. This makes it harder to generalize the results for a younger audience. However, the participants were at least not more technologically adept than the targeted group of users.

Every participant had at least a satisfactory level of understanding English language. This was needed since not only were the content in English but the interfaces as well. The participants were not required to speak nor write English as they could comment in their native language if they so desire. Twenty out of the twenty-five participants were males whereas the remaining five were females. Only fifteen of them had heard about the concept RCD tools for structural engineers before and only three have actually used RCD tool for design of RC structure. The occupations of the participants varied from civil/structural engineering to building/architecture.

5.2 TEST PROCESS

The tests done were planned to be laboratory tests due to time and financial constraints. A one-hour time slot was reserved for each of the interviews and the events were lead by a usability specialist. In practice, the tests took less than one hour in total, to keep the fatigue levels of the users tolerable. As the tests were limited by time, the RCD Beam tool was pre-installed into the computer devices.

All the test devices came with camcorder which serves as a camera to transmit the image to the observer. Video stream was not recorded and the screen was visible to all persons in the room to avoid any confusion about what was being filmed. The image itself was focused on the display of the computer device and did not show more than the fingers of the participants. The tests were held in a private meeting room during weekdays. Two persons were present in the interviews in addition to the test user. One of the persons was a usability specialist who made the actual interviews while the other was part of the development team of the tested application. The interviewer was responsible of explaining the situation and led the users through the event.

The role of the observer was to take notes (with pen and paper) about the responses and reactions of the test person as well as to get first hand experiences how the users felt about the tool. It was not revealed that the observer was the developer to make sure the users would not be too kind in their replies. No audiovisual recordings were taken during the interviews, again to make the situation more comfortable to the test users. The interviews were given tasks to perform on both of the applications and the order was altered so that half the test users started with STAAD Professional Application and the remaining half with the prototype RCD Beam. Some of the planned test tasks were removed from the test process to make the situation more pleasant for the test users and avoid embarrassing them. Tests were held in laboratory, which caused some social restraints and some possible tasks such as walking around the room might have felt too awkward. Thus only tasks that the user could accomplish while sitting were given to test participants. Eventually the users were asked to perform five tasks:

a. Set up RCD Beam environment and load RCD Beam

b. Determine and pick type of beam for design and Process input.
c. Analyze and Design beam using RCD table advisor
d. What can you deduce from checking for serviceability?
e. What information can you find from the analysis and design process?

After following through the task assignments with both applications, the users were asked to fill in a questionnaire and evaluate their experiences. Also, an informal discussion was held about the experience.

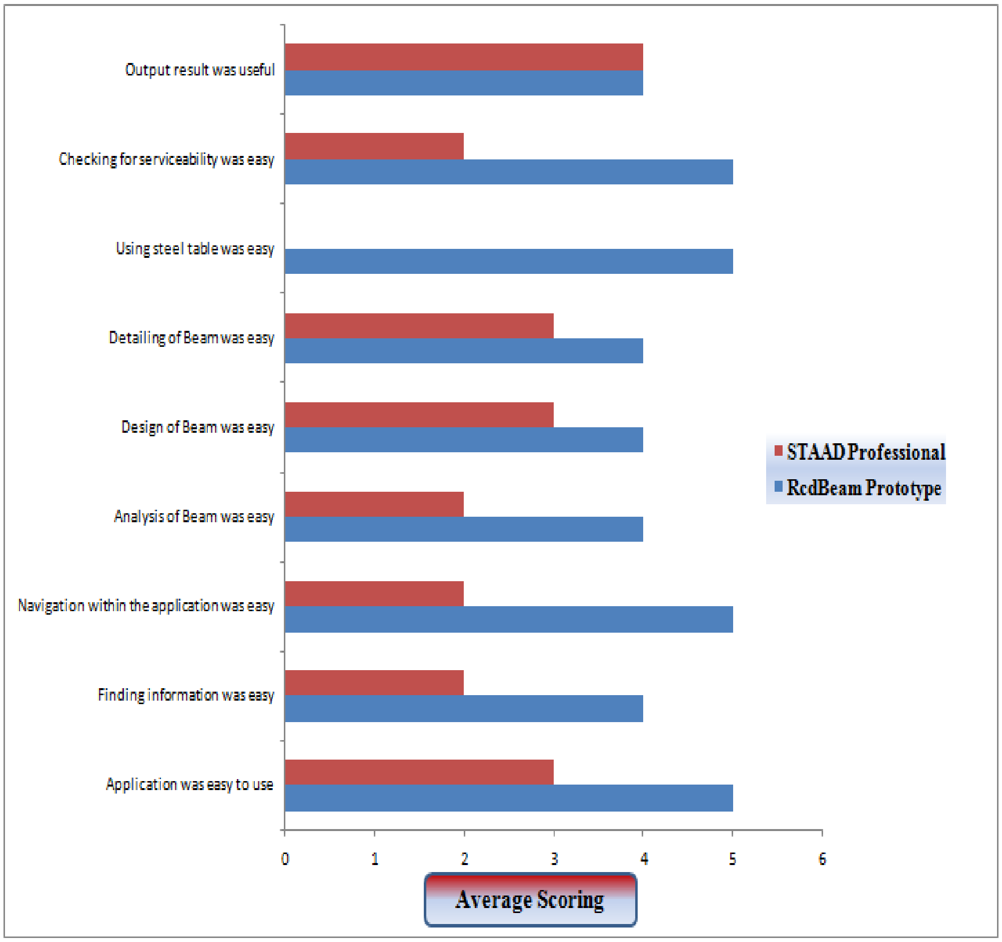

Figure 8: The bars represent the averaged scores for each of the questions the users were asked to evaluate. Blue bars indicate the scores given to RcdBeam prototype application whereas the red bars (dark grey) present the scores the tested prototype received.

5.3 TEST RESULTS

The users were asked to evaluate various aspects about each of the applications by giving a rating from 0 to 5. 0 was reserved to indicate the most negative answer possible (e.g. "settings were impossibly hard to implement") whereas 5 meant an extremely positive experience (e.g. "settings were in obvious place and easy to implement"). The averages of the scores as well as the test questions are indicated in the figure 10. RCD Beam application specific questions missing from figure 8 were "Could you imagine yourself using RCD Beam in the future?" and "What kind of a feature you would have liked to have?"

The STAAD application is consistently with one exception slightly behind the prototype RCD Beam tool implementation; the ease of navigation to the next appropriate step. The prototype lost only on the evaluation of the easiness of getting the output which is easily obtainable within the interface unlike the prototype which has to borrow the services of AutoCAD interface to render its output. The last part of the questionnaire dealt with two questions asking the users to make a choice between the applications. The values in figure 9 present the number of users that made the particular choice in regards to each of the questions.

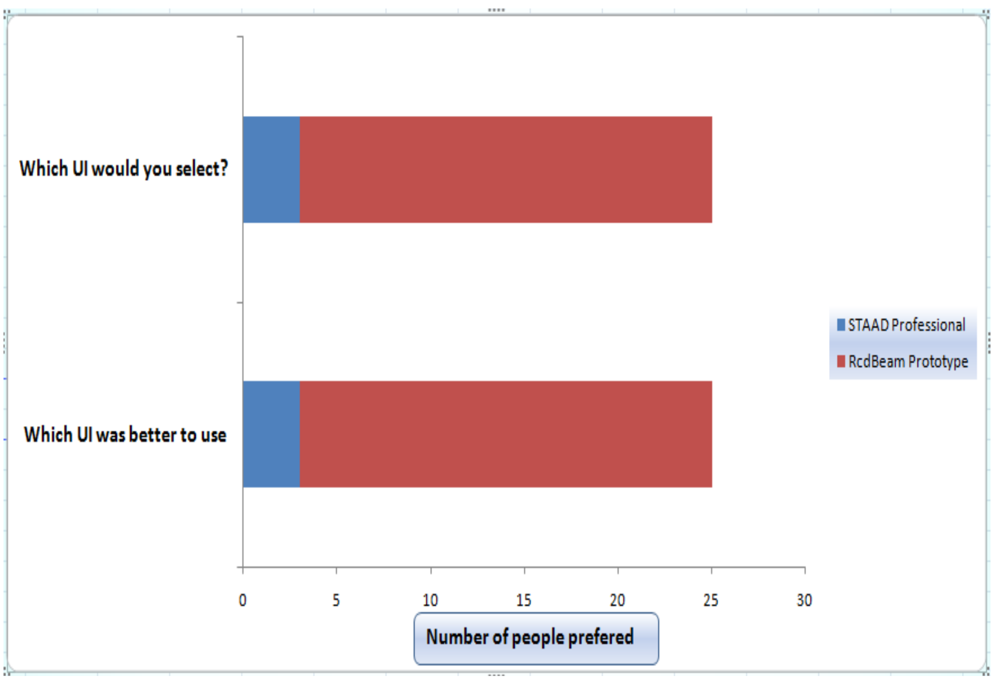

Figure 11. The preference choice of the participating users. Twenty-two users out of Twenty-five preferred the prototype RCDBeam. The remaining three would have chosen the STAAD Professional Application.

For the question “which UI is better to use” two of the twenty-five participants answered the STAAD Application whereas the remaining twenty-three choose the developed RCD Beam prototype. The numbers for the question “which UI would you choose yourself” were the same. In total, the users had to answer 27 questions, ten about RCD applications generally and seven that compared the two and tried to gain understanding about their preferences. The remaining questions that have not been mentioned so far were:

a. How important to you is the pleasure of using the interface (on scale from 0 to 5)?
b. How important for you are the features (on scale from 0 to 5)?
c. Which were the three best things about these applications?
d. Which were the worst things about the applications?
e. What would you wish to change or improve about these applications?

In general, the answers highlighted the role of the user interface over features, but such result is likely to be very culturally and demographically depending. That said, pleasurable UI averaged to 4.5 whereas the importance of features got an average of 3.4.

5.4 FINDINGS AND THE ANALYSIS OF THE RESULTS

The selections of participants were mainly civil/structural engineers and builder/architects who work in the construction industries. Also, the tests were 1 hour in length, which forced the study to concentrate on the immediate usability, and experience what the UI was able to offer. RCD Beam tool, by their nature, are designed to help RCD analysts analyze, design and visualize processes involved in RC beam design. Thus the likelihood of getting irrelevant feedback on the usability was to be taken into account. Short-term usability tests where users need to move from a fixed, familiar interface style to something they find completely new are more likely to cause resistance and annoyance when things do not work as they did before and after the specific tests. This also goes the other way around to hide possible problems the prototype UI might have over a prolonged usage. Factors such as the visual pleasantness and “wow”-factor can hide usability issues as well.

Also, most of the users were older than the approximated target group, which is a factor that should also be taken into account while analyzing the results. In interpreting the test results it should also be kept in mind that the RCD tool prototype was an initial one and was missing even many planned features such as a drawing environment in the absence of AutoCAD installation. Despite the request that the users would have been experienced users, the settings were difficult to find with the STAAD application. This was partly due to the fact that the software is very expensive and not usually available to individuals except companies/organizations who can afford it. The emotional contact between the test users and the content was understandably thin, as the users did not have practical means to have content they would have liked. The last two questions, however, were crafted to make the users give an evaluation (without justifications) of which of the applications they would have chosen themselves. This was deemed to be the most important finding on the test, and on that the prototype was the users preferred choice even if with a small margin. The technical immaturity was likely a factor that the margin was not larger than the results demonstrate. As some of the users were not highly technical in the use of computers, the comments and feedback were not always in relation to the software itself but their interaction with the computer system. This is to be expected, to some extent, but it also highlights that the user experience does not rely on the software or hardware but the combination of both. In optimal case, neither should be omitted when designing the products and both should support each other given the targeted users. The evaluation of the user experience was eventually shrunk to the two last questions ("Which UI was better to use" and "Which UI would you select"). Of these the latter was deemed more important as that would give the most meaningful answer about the preference of the users. The study did not answer to the questions on how the different elements affected the user experience or which of these elements were most meaningful ones. Overall, the prototype scored well against the established application. The scores on their own did not differ much between the compared clients, but they were consistently in favour of the prototype. Users preferred the prototype solution in eight of the ten questions. Qualitative comments also supported the general preference drawn from the measured scores. The 27-year-old female podcast users said that the current (STAAD Application) UI is ugly and boring, and thus would not want to use it. On the other hand, the 48-year-old female told the interviewer that even if she prefers the current UI the prototype could be better in the long run.

5.5 IMPROVEMENT IDEAS

We received good design feedback from participants suggesting how best to move towards redesign. For example, many users disliked the black background colour of the textbox for message alert. They wanted a white background with fore-colour in green indicating success while red fore-colour to indicate danger or failure. They also wanted to see all the messages sent to the message alert textbox to be appended for ease of review. Users expressed strong concerns about the desirability of entering the design moment directly as an option with 'area of steel calculated' automatically generated.

It is rare to encounter a clear-cut expression of preference, or the reverse, for a thoroughly explored innovative interface, and the outcome of an overall satisfaction questionnaire and briefing completed by participants is no exception. Responses to the questionnaire revealed no significant differences, though users preferred the new prototype tool, several areas of future work were identified. Inevitably users requested a long list of desirable features and these must be examined to see how they would affect users without jeopardizing ease of use for the novice. It was also recognized that studies must be carried out on how to incorporate our tool into hand-held devices like the newly introduced Windows Phone 7 using pens and touch-screens rather than the mice and keyboards that necessarily had to be employed in the reported studies.

What should be studied is what the user think of the method after all the other functionality is included to the controls. As the prototype was the preferred choice for the participants, further studies should be done to break down the user experience factors by altering the elements present in the interface. Such studies would hopefully clarify the relative importance of the functions and the looks that the prototype implemented.

6. CONCLUSIONS

- As a result from this paper a prototype implementation of RCD Beam tool client user interface was created and validated in user trials against a competing RCD product (STAAD Professional). The implementation offered the basic functionality that was deemed important to study user experiences and interface engineering over a short period of time and with RCD expert and analysts of which most had never carried RCD using RCD tools or software application before. Moreover, the paper studied the possibilities to offer additional value to the end user in subtle forms and prioritize the default information presented in the user interface. Most of the methods were not functional features per se, but were implemented to create a more pleasing experience. The user studies were made with a limited set of users to compare the prototype to a well established STAAD product and made to evaluate the pleasantness of each of the applications. The majority of the test users (twenty-two out of twenty-five) found the RCD prototype to be their preferred option if it was a fully working one. Main reasons for this seemed to be fast and responsive interface that presented relevant information in easily understandable manner. This was visible, for example, in the scores for easiness of navigation. The look of the application was very likely to affect the positive feedback as well.

- It could be speculated that a holistic user experience is a balance of at least usability, pleasure of use, aesthetics, responsiveness and predictability. Other factors are likely to come into play as well, but none of the aforementioned aspects suffices alone. Some level of usability is a requirement, but does not guarantee that the product is liked. Equally, an appealing visual look does not help long if the user does not understand the system. Thus, the design needs to take each aspect into account and find a solution that takes these into serious consideration.

- It is extremely difficult to say what aspects affected the positive end results when it comes to user experience. But what is clear is that it is not enough to think about application design with mere functional demands and requirements; how things appear and feel is an important factor to the actual end user. It is difficult to name a formal method of creating attractive and pleasant applications from the end-user perspective. Also, the proposed hypotheses about emotional functions (transferring emotional context and creating a connection via quirks) are not validated by these tests alone but they are not shown to be in direct conflict either. Further studies would be needed to refine the assumptions and to see how general those might be.

- The prototype has introduced the use of Service Oriented Application through the web service to check for serviceability in RCD. All the controls work on top of AutoCAD interface; communicate with AutoCAD through interoperability to visualize Bending Moment, Shear Force diagrams and Beam Detailing.

- The limitation of the RCD prototype tool is that the embedded RCD table advisor cannot read user's mind; users must generate data which will serve as a guide for picking reinforcement steel. Service must be available through a remote server or the internet in other to check for serviceability like deflection, minimum and maximum percentage of steel required in RC Beam. The major plus of the prototype tool is that it is almost platform independent unlike STAAD professional that is totally dependent on a particular platform. STAAD professional unlike RCD prototype automatically generate steel properties thus suppressing human perception and contribution to RCD. In future, our prototype tool will be redesigned to imbibe the emerging technology of cloud computing.

REFERENCES

Alben, L. (1996). "Quality of Experience: Defining the Criteria for Effective Interaction Design. Interactions", Journal of ACM Transactions on Computer-Human Interaction (TOCHI), Volume 3(3), May/June 1996, pages 11-15

Allen, B. (2000). Individual differences and the conundrums of user-centered design: Two experiments. J. Am. Soc. Information Sci., 51(6), 508–520.

Andrew, W. B., John, B. N. (2010). Computer Aided Concrete Design; A TLTP Phase 2 Consortium Project. http://www.shef.ac.uk/~compact.

Bill, M., John, B., Ray, H. (2007). Reinforced Concrete Design to Euro code 2. Palgrave Macmillan.

Butler, Keith A. (1996). Usability Engineering Turns 10. ACM Press, January 1996, 17 pages

Charles, E. R., James, C. S. (2003). Examples of the Design of Reinforced Concrete Buildings to BS 8110. Publish by E & FN Spon.

Chen, C., Yu, Y. (2000). Empirical studies of information visualization: a meta-analysis. Int. J. Human-Comp. Studies, 53(5), 851–866.

Combs, T. T. A., Bederson, B. B. (1999). Does zooming improve image browsing? *Proceedings of the fourth ACM Conference on Digital Libraries*, pages 130–137.

Donald, A. N. (2002). The Design of Everyday Things. First Doubleday / Currency Edition, New York

Donald, A. N. (2002). "Emotional Design: Why We Love (or Hate) Everyday Things". First Doubleday, New York.

Dumas, J. R., Redish, J. C. (1993). A Practical Guide to Usability Testing. Ablex Publishing Corporation Norwood NJ.

Ericsson, K. A., Simon, H. A. (1980). Verbal Reports as Data. Psychologic. Rev., 87, 215-251.

Ericsson, K. A., Simon, H. A. (1984). Protocol Analysis: Verbal Reports as Data. Cambridge, MA: Bradford Books / MIT Press.

Fady, R. (2008). Computer Analysis and Reinforced Concrete Design of Beams, Department of Civil Engineering, University of Nairobi, Kenya

Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly.

Few, S. (2009). *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press.

Folorunso, O., Yusuf, L. O., Okesola, J. O. (2010). “SOA-RTDBS: A Service Oriented Architecture (SOA) Supporting Real Time Database System”. Oriental J. Comp. Sci. Technol., 3(1), 171-184

Grady, H. (2000). Approaches to prototyping: Web site design: a case study in usability testing using paper prototypes, IEEE Educational Activities Department, September 2000, 7 pages

Hassenzahl, M., Tractinsky, N. (2006). User Experience – Research Agenda. Behaviour and Information Technology, Volume 25, No 2, March-April 2006, pages 91-97

Hertzum, M. (1999). User Testing in Industry: A case study of laboratory, Workshop and Field Tests. Proc.: 5th ERCIM WORKSHOP ON "USER INTERFACES FOR ALL" http://ui4all.ics.forth.gr/UI4ALL-99/Hertzum.pdf

Johnson, P. (1998). Usability and Mobility; Interactions on the move. In Proceedings of the First Workshop on Human-Computer Interaction with Mobile Devices, Glasgow, Scotland, GIST Technical Report G98-1.

Judith H., Robin B., Carol, B, Marcia K. (2007). Service Oriented Architecture for Dummies; Wiley Publishing Inc.

Kallio, T., Kekäläinen, A. (2004). Improving the Effectiveness of Mobile Application Design: User-Pairs Testing by Non-professionals. Proc. MobileHCI 2004 Conference, Glascow, UK Springer-Verlag 2004 315-319.

Kjeldskov, J., Skov, M. B., Stage, J. (2004b). Instant Data Analysis: Evaluating Usability in a Day. In Proceedings of NordiCHI 2004, Tampere, Finland. ACM, 233-240

Kjeldskov, J., Skov, M. B., Als, B. S., Høegh, R. T. (2004a). Is it Worth the Hassle? Exploring the Added Value of Evaluating the Usability of Context-Aware Mobile Systems in the Field . In Proceedings MobileHCI 2004 conference, Glasgow, UK. Springer- Verlag, 2004. 61-73.

Kjeldskov, J., Graham, C. (2003). A Review of Mobile HCI Research methods. In Proceedings of the 5th International Mobile HCI 2003 Conference, Udine, Italy, Sringer-Verlag.

Krafzig, D., Banke, K., Slama, D. (2004). *Enterprise SOA, Service Oriented Architecture Best practices*. Prentice Hall, November 2004. ISBN 0-13-146575-9

Larkin, J.H., Simon, H.A. (1987). Why a diagram is (sometimes) worth 10,000 words. Cognitive Sci., 11(1), 65–99.

Lim Youn-kyun, Pangam, A., Periyasami, S., Anjea, S. (2006). Comparative analysis of high- and low-fidelity prototypes for more valid usability evaluations of mobile devices. ACM Press, 2 pages.

Mäkelä, A., Fulton, Suri J. (2001): Supporting Users’ Creativity: Design to Induce Pleasurable Experiences. Proceedings of the International conference on Affective Human Factors Design, ASEAN Academic Press, 2001, pages 387 – 394

Mehrabian, A. (1981). Silent messages: implicit communication of emotions and attitudes, Albert Mehrabian, 2. Ed edn. Wadsworth, Belmont, Calif. 1981

Mike, R., Boris, L., Kevin, T. S., Marc, J. B. (2008). Applied SOA: Service Oriented Architecture and Design Strategies; Wiley Publishing, Inc. Indianapolis, Indiana

Mizutani, M. (2006). Emotional Communication. Media Lab, University of Art and Design Helsinki, March 2006, 51 pages.

Nielsen, J., Landauer, T. K. (1993). A Mathematical Model of Finding Usability Problems. In proceedings of INTERCHI ’93. 206-213.

Nielsen, J. (1993). Usability Engineering. Published by Morgan Kaufmann, San Francisco, 1993.

Nielsen, J. (2000). Designing Web Usability, New Riders, cop 2000, 419 pages.

OASIS. (2006). *Reference Model for service oriented architecture 1.0*, Committee specification 1, 2 August 2006. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm.

Papazoglou, P. M. (2003). *Service-oriented computing: Concepts, Characteristics and Directions*. In: Proceedings of the Fourth International Conference on Web Information Systems Engineering.

Papazoglou, P. M., van, den Heuvel W. (2007). *Service Oriented Architectures: Approaches, Technologies and research Issues*. The VLDB J.

Perrey, R., Lycett, M. (2003). *Service-Oriented Architecture.* Symposium on Applications and the Internet Workshops, 2003 Proc., pp. 116- 119

Qusay, F. H. (2009). Aspect of SOA: An Entry Point for Starters. Annals Computer Science Series 7th Tome 2nd Fasc. 2009.

Reponen, E., Pertti, H., Kristijan, M. (2006). Mobile Video Recording in Context, interactions, July & August.3 pages.

Robertson, G., Czerwinski, M., Larson, K., Robbins, D.C., Thiel, D., and Dantzich, M. van. (1998). Data Mountain: using spatial memory for document management. Proc.: 11th annual ACM Symp. on User Interface Software and Technology (UIST'98), pages 153–162.

Rod, S. (2010). Visual Basic 2010 Programmer’s Reference; Published by Wiley Publishing, Inc. 10475 Crosspoint Boulevard Indianapolis, IN 46256. www.wiley.com

Roto, V., Oulasvirta, A., Haikarainen, T., Lehmuskallio, H., Nyyssönen, T. (2004). Examining mobile phone use in the wild with quasi-experimentation. HIIT Technical Report 2004-1, August 13.

Roto, V. (2006). Web Browsing on Mobile Phones – Charasteristics of User Experience, Helsinki University of Technology, December 8th, 2006, 86 pages.

Rudd Jim, Stern Ken, and Isensee Scott (1996). Low vs. high-fidelity prototyping debate, ACP Press. Interactions, Volume 3 Issue 1, 10 pages.

Sebrechts, M.M., Vasilakis, J., Miller, M.S., Cugini, J.V. Laskowski, S.J. (1999). Visualization of Search Results: A Comparative Evaluation of Text, 2D and 3D Interfaces. Proc.: 22nd Ann. Int. ACM SIGIR Conf. on Research and development in information retrieval (SIGIR'99), pages 3–10.

Swan, R.C., Allan, J. (1998). Aspect Windows, 3-D Visualizations, and Indirect Comparisons of Information Retrieval Systems. Proc.: 21st Annual Int. ACM SIGIR Conf. on Research and development in information retrieval (SIGIR'98), pages 173–181.

Tamminen, S., Oulasvirta, A., Toiskallio, K., & Kankainen, A. (2004). Understanding mobile contexts. Special Issue of J. Personal and Ubiquitous Computing. 8, 135-143.

Tirupathi, R. Chandrupatla, and Ashok, D. Belegundu (2008). Introduction to Finite Elements in Engineering. Published by Pearson Education Inc.

Trewin, S. (2006). Physical Usability and the Mobile Web, ACM Press, May 2006, 4 pages.

Tsai, W.T., Xiao, B., Paul, R. A., Chen, Y. (2006). “Consumer-Centric Service-Oriented Architecture: A new Approach”, Proceeding of IEEE 2006 International Workshop on Collaborative Computing, Integration, and Assurance (WCCIA), April 2006, pp. 175-180.

Tufte, E. R. (1990). *Envisioning information*. Graphics Press Cheshire, Conn. (PO Box 430, Cheshire 06410).

Tufte, E. (1983). *The Visual Display of Quantitative Information*. Graphics Press, Chelshire, CT. UPA (Usability Professionals' Association) (2007). "Usability Body of Knowledge", http://www.usabilitybok.org/glossary, referenced at April 1st 2007

Yusuf, L.O., Folorunso, O., Akinwale, A.T., Adejumobi, A.I. (2009). Visualizing the behaviour of reinforced concrete beam structure under various types of loadings. African J. Mathemat. Comp. Sci. Res., 2(10), 202-217.

Yusuf, L.O., Folorunso, O., Akinwale, A.T., Adejumobi, A.I. (2010). Visualization RCD (reinforced concrete design) table advisor for decision support activity. African J. Mathematics Comp. Sci. Res., 3(7), 132-142.
