A FULL LIFE-CYCLE METHODOLOGY FOR STRUCTURED USE-CENTERED QUANTITATIVE USABILITY REQUIREMENTS SPECIFICATION AND USABILITY EVALUATION OF WEBSITES

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

Guoqiang Hu

Certificate of Approval:

Richard O. Chapman
Associate Professor
Computer Science and Engineering

Kai-Hsiung Chang, Chair
Professor
Computer Science and Engineering

Juan E. Gilbert
Associate Professor
Computer Science and Engineering

George T. Flowers
Dean
Graduate School

A FULL LIFE-CYCLE METHODOLOGY FOR STRUCTURED USE-CENTERED QUANTITATIVE USABILITY REQUIREMENTS SPECIFICATION AND USABILITY EVALUATION OF WEBSITES

Guoqiang Hu

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama
December 18, 2009

A FULL LIFE-CYCLE METHODOLOGY FOR STRUCTURED USE-CENTERED QUANTITATIVE USABILITY REQUIREMENTS SPECIFICATION AND USABILITY EVALUATION OF WEBSITES

Guoqiang Hu

Signature of Author

December 18, 2009

Date of Graduation

DISSERTATION ABSTRACT

A FULL LIFE-CYCLE METHODOLOGY FOR STRUCTURED USE-CENTERED QUANTITATIVE USABILITY REQUIREMENTS SPECIFICATION AND USABILITY EVALUATION OF WEBSITES

Guoqiang Hu

Doctor of Philosophy, December 18, 2009
(M.S., Peking University, 1993)
(B.S., Shenyang Institute of Technology, 1986)

199 Typed Pages

Directed by Kai-Hsiung Chang

World Wide Web has gained its dominant status in the cyber information and services delivery world in recent years. But how to specify website usability requirements and how to evaluate and improve website usability according to its usability requirements specification are still big issues to all the stakeholders. To help solve this problem, we propose a website usability requirements specification and usability evaluation methodology that features a *structured use-centered quantitative full life-cycle* method.

A validation experiment has been designed and conducted to prove the validity of the proposed methodology, *QUEST* (Quantitative Usability Equations SeT). Its principle is to prove that QUEST has stronger website usability evaluation capability than the most

typical existing usability evaluation methods. Apparently, if QUEST's website usability evaluation capability is established, then its usability metrics can be used to quantitatively specify upfront user usability requirements for websites.

In the validation experiment, 7 usability experts and 20 student subjects were recruited to perform 4 tasks on 2 open source calendar websites, WebCalendar 1.0.5 and VCalendar 1.5.3.1; 4 sets of usability data had been collected, which were corresponding to the following 4 usability evaluation methods respectively: expert usability review, traditional user usability testing, *SUS* (System Usability Scale), and QUEST.

According to the experiment results: both the expert usability review and the traditional user usability testing were inconclusive on which of the 2 target websites had better usability; although SUS rated the overall usability of WebCalendar 1.0.5 at 66.00 and VCalendar 1.5.3.1 at 61.75, it was subjective and vague on usability problems; in contrast, QUEST not only rated the overall usability of WebCalendar 1.0.5 at 56.59 and VCalendar 1.5.3.1 at 35.97, but also revealed where the usability problems were and how severe each usability problem was in a quantitative manner. In conclusion, it clearly can be stated that QUEST has stronger website usability evaluation capability than all other 3 most typical existing usability evaluation methods. So, the proposed methodology has been validated by the experiment results.

ACKNOWLEDGMENTS

This dissertation is whole-heartedly dedicated to the author’s always loving, always caring, and always supporting family.

Style manual or journal used Communications of the ACM

Computer software used Microsoft® Office Word 2003

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Today, Internet has reached almost every corner on earth and it connects all of us together [1][2]. On the Internet, World Wide Web (WWW) [3][4] has become one of the most powerful and influential Internet applications [5]. Now, just like the air we breathe, WWW is everywhere.

WWW has rapidly gained its dominant status in the cyber information and services delivery world by its simplicity, platform-independency, extensibility, flexibility, and versatility. It is hard to imagine the kind of information services or applications that cannot be built on the Web; and except physical objects, it is also hard to imagine the kind of objects that cannot be delivered through it. WWW has become not only an indispensable social mechanism of our society but also an essential daily necessity for most people. By its great impact to people's living, WWW has changed, to a great extent, the way people think about the computing technology. The importance of the WWW to the proper functioning of the human society is beyond any words can say [1][2][5][6].

WWW consists of tens of millions of Web sites or Web-based applications[1] [7] distributed all over the world. Because of WWW's significant value to all of us, how to specify website usability requirements and how to evaluate and improve website usability according to its usability requirements specification are big concerns to all the stakeholders. However, currently there exist no good ways to address this issue. To help solve this problem, we propose a website usability requirements specification and usability evaluation methodology that features a *structured use-centered quantitative full-life-cycle* method. Here, *use* refers to a real use of a designed task of a website by an end user; *use-centered* simply stresses the view[2] that because usability issues originate from use, usability study should be not only based on use but also focused on use, and that usability should be engineered for use and evaluated by use. In other words, usability study should be *from use, on use, for use,* and *by use*, thus be *use-centered*.

Our approach is: a system's usability is quantitatively defined in terms of its goal-tasks'[3] usabilities; in turn, a goal-task's usability is quantitatively defined in terms of its 5 major usability aspects; and further, each major usability aspect is quantitatively defined in terms of its basic use features. In this way, a structured and quantitative usability engineering framework for websites is set up.

---

[1] For convenience, Web sites or Web-based applications will be uniformly referred to as *websites* in this dissertation.

[2] When usability is concerned, in contrast to *user-centered*, the term *use-centered* is more appropriate and more accurate: first, usability problems occur during *uses* rather than on *users*; and further, *use-centered* takes into consideration the users, the task, and the interaction between them at the same time.

[3] A system can be divided into tasks. Because in an implemented system, each task is designed to achieve a certain goal and each goal is accomplished through a specific task, in this dissertation, the term *goal-task* is used to simultaneously represent a goal and the activities required to achieve the goal. Goal-task is a basic research object of this usability study.

The process of this methodology is: at system analysis stage, after goal-task analysis, each goal-task's user usability requirements can be assigned by quantitatively specifying the desired value for each of its major usability aspects' basic use features; At the same time, each goal-task's weight and use frequency in the target system can also be specified according to its relative importance and use frequency in the current system. Then, with the above quantitative specifications obtained, each goal-task's composite use features and the entire system's usability can be easily derived through their respective defining formulas. Finally, all the above information put together *as a package* forms the usability requirements specification for the entire system. It should be recognized that the user usability requirements have equal status with other traditional user requirements, such as user functional requirements. So, at all the other stages of the website's life-cycle, each time a review or testing is performed, the usability requirements specification should also be tested against to see if it has been satisfied just like functional requirements specification has always been. The only difference between them is the testing methods used, i.e., for the functional requirements specification, the testing method is the traditional software testing; but for the usability requirements specification, the testing method is usability testing by use.

Apparently, the user usability requirements specification should be agreed upon between the system analyst and the end user(s) (sometimes, the system procurer in lieu of the end users). The key point to be considered here is its economic, or *budgetary*,

implication, because as quality requirements, the higher the usability requirements are, the more expensive it will be for the target system to satisfy them.

This quantitative usability methodology is independent of, and therefore can be seamlessly integrated into, any engineering methodologies, processes, and techniques. For example, this methodology can be integrated into the Waterfall model as is illustrated in Figure 1.1.

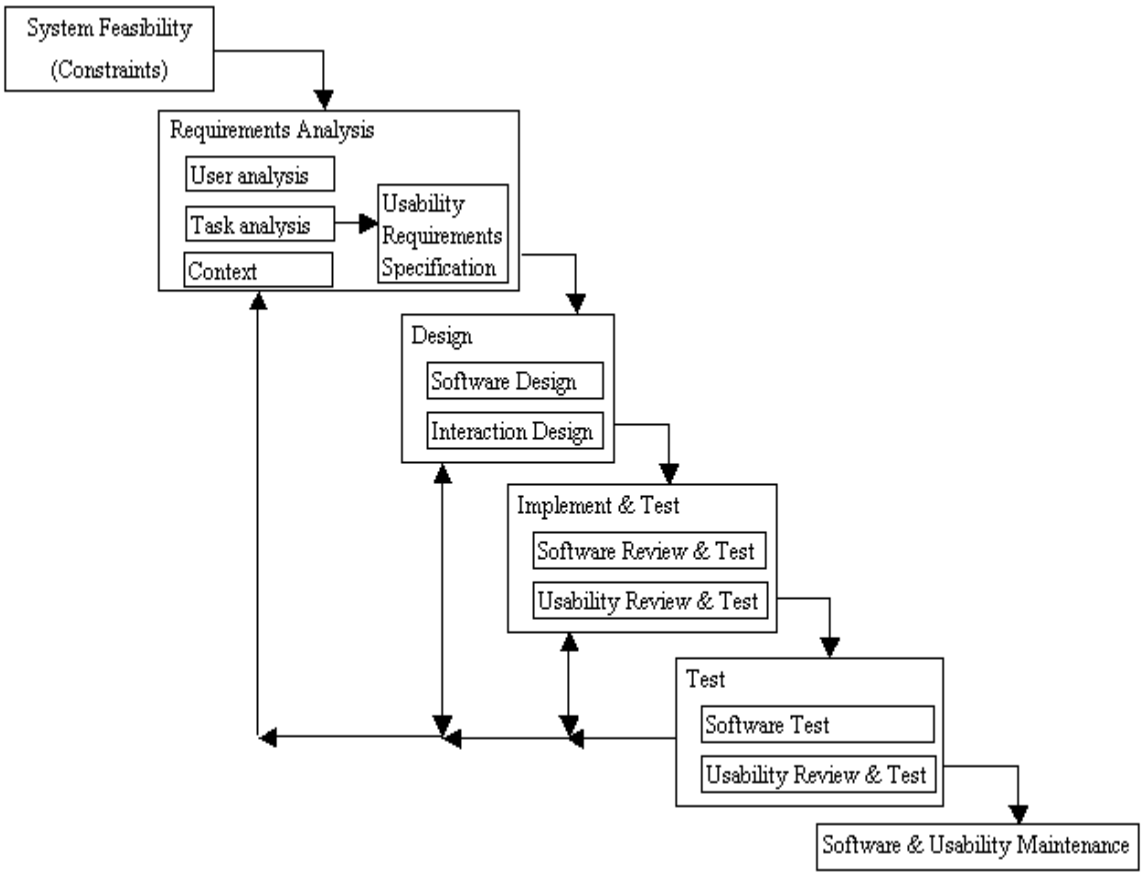


Figure 1.1 The methodology illustrated in Waterfall model

In order to prove the validity of the proposed methodology, a validation experiment has been designed and conducted. The principle of the validation experiment is to prove that the proposed methodology has stronger website usability evaluation capability than the following 3 most typical existing usability evaluation methods: expert usability evaluation, traditional usability testing, and SUS (*System Usability Scale*) [41]. Here, *website usability evaluation capability* contains the following 3 aspects: overall website usability evaluation, usability comparison between websites, and usability problem diagnosis for a website. Apparently, if the proposed methodology's website usability evaluation capability is established, then its usability metrics can be used to *quantitatively* specify *upfront* user usability requirements for websites.

The entire validation experiment was a double-blind and multi-control-group design. In the validation experiment, 7 usability experts and 20 student subjects were recruited to perform 4 tasks on 2 open source calendar websites, the WebCalendar 1.0.5 and the VCalendar 1.5.3.1, which were hosted locally. 4 sets of usability data had been collected, which were corresponding to the 4 usability evaluation methods respectively. According to the results of the validation experiment, it can be concluded that the proposed methodology has been validated. The details of the entire validation experiment are presented in Chapter 6.

Although the topic of this dissertation is focusing on how this methodology can be applied to website usability engineering process, the approach of defining a structured

and fully quantitative usability framework for websites can also be applied to any other human-tool interaction systems. The difference lies in the specific use features that have to be considered for a particular kind of human-tool interaction system. The advantage of this kind of quantitative usability framework is that no matter what particular kind of human-tool interaction system it is applied to, all the resulted usabilities are comparable with each other. In other words, the usability of a hammer can be compared with the usability of a website. Unfortunately, any further discussion of this expanded topic is beyond the scope of this dissertation.

The rest of this dissertation is organized as follows. Chapter 2 presents the literature review, compares the proposed methodology with related work, and highlights the potential contributions of the proposed methodology. Chapter 3 defines the concept of use feature, explains the mental model schism theory, identifies the existing problems that the proposed methodology intends to solve, and presents the principle of the proposed methodology. Chapter 4 introduces the architecture of World Wide Web, and points out some important features of websites that are critical for understanding our approach to website usability study. Chapter 5 presents the entire set of structured and fully quantitative website use feature definitions, and illustrates how to use these use features to specify *upfront* user usability requirements for websites. Chapter 6 introduces the design and setup of the validation experiment of the methodology, presents the 4 sets of experiment data that are corresponding to the 4 usability evaluation methods, compares

the 4 sets of experiment results and concludes the validation experiment of the methodology. Chapter 7 concludes the research, gives more discussion about the methodology, and points out future work. The complete set of traditional usability testing experiment data is given in Appendix A, and the complete set of *QUEST* experiment data is given in Appendix B.

CHAPTER 2

BACKGROUND

2.1 The history of research in usability

Usability is about how effectively, efficiently, and easily things can be used by human beings. Research in usability has a long history.

In its early stage under the terms like *Ergonomics* [8] and *Human Factors* [9], research in usability was mainly concerned with how to match the physical capabilities of humans and devices so that they could interact most effectively, efficiently, and safely.

After computer systems became widely used since the early 1980's, a new discipline or inter-discipline called *Human-Computer Interaction* (HCI) [10] emerged to specifically take on the issues related to the interaction between humans and computers. Compared to traditional Ergonomics and Human Factors, HCI stresses on how to match the mental and physical capabilities of humans and computers. The research scope of HCI covers the intersection of the disciplines such as *Human Cognition*, *Human Perception*, *Human Intelligence*, *Anthropometry*, *Biomechanics*/*Kinesiology*, *Sociology*, *Philosophy*, *Behavioral Science*, *Computer Science* and *Software Engineering*. Closely related to HCI, in 1986, the term *Usability Engineering* was coined to only name the

subset of the research in usability that specializes in usability of computer systems, especially software systems.

It should be noted that, with the time going and technologies advancing, all the terms mentioned above have taken on new meanings. Because the evolving history of these disciplines is beyond the concern of this dissertation, this chapter will only focus on usability engineering, or more specifically, website usability engineering.

The rest of this chapter is organized as follows. Section 2.2 briefly reviews the general accomplishments in usability engineering. Section 2.3 focuses on the achievements in website usability engineering. Section 2.4 contrasts our work with other related work and describes the potential contributions of the proposed methodology.

2.2 Usability engineering

In the early 1980's, software usability became a big concern in software engineering because people found out that there were many software products that were simply not very usable. Researchers [11][12][13][14][15] discovered that software usability problems were caused by designers who took a computer- and/or designer-centered view and were not considerate for their end users. So very soon, *user-friendly* [16][17][18] became a buzzword in the computer technology community. But in order to be more accurate and stress the shift of focus from computers and designers to end users, the term *user-friendly* was banned in favor of *user-centered* in

User Centered System Design by Norman and Draper (1986) [19], and this practice has been broadly accepted ever since [20].

Different definitions of usability for software systems have been given by Miller [21], Shackel [22][23][24][25], Bennet [26][27], Sheiderman [28][29][30], Nielsen [20], Bevan [31], Löwgren [32], Dix [33], Quesenbery [34][35], etc. In 1998, ISO 9241-11 [36] defined usability as: "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." It is the ISO 9241-11 usability definition that has been recognized as authoritative and has been widely adopted. Because our methodology is deeply related to ISO 9241-11 usability definition, its details will be further presented and evaluated in Chapter 3.

Based on the usability definitions mentioned above, a variety of usability metrics [20][37][38][39][40][41][42] have been suggested. Hornbæk provided a comprehensive review of current practices in measuring usability [84], which will be further introduced in 2.4.1.

There are many techniques [20][38][39][43][44][45][46][47] for usability evaluation, inspection, and testing. Usability evaluations carried out during a development cycle in order to improve the usability of the product under development are called formative usability evaluations; usability evaluations carried out at the end of a development cycle in order to show the overall usability of the evaluated product are

called summative usability evaluations. Also, usability evaluations "involving evaluating parts or aspects, either as a means to an overall evaluation or without the final synthesis" are called analytic evaluations [48]; usability evaluations aiming at "the allocation of a single score/grade/evaluation to the overall" usability of the evaluated product are called global evaluations [48].

It is very expensive to perform strict traditional usability evaluation and testing. To solve this problem, in 1989, Nielsen [49] proposed *discount usability engineering*, because it was found that 80% of usability problems could be detected with 4 to 5 participants [50][51]. Discount usability engineering is based on the following 4 techniques:

- User and task observation
- Scenarios
- Simplified thinking aloud
- Heuristic evaluation

In order to extend usability to as many user groups as possible, *universal usability* was proposed. It is believed that "universal usability will be met when affordable, useful, and usable technology accommodates the vast majority of the global population: this entails addressing challenges of technology variety, user diversity, and gaps in user knowledge in ways only beginning to be acknowledged by educational, corporate, and government agencies." [52]

With all the above research achievements in software usability being systematically put together [20][38][39][53][54][55][56][57][58], *Usability Engineering* as a discipline was formally established. In 1986, Good et al. [59] defined that:

> "Usability Engineering is a process, grounded in classical engineering, which amounts to specifying, quantitatively and in advance, what characteristics and in what amounts the final product to be engineered is to have. This process is followed by actually building the product, and demonstrating that it does indeed have the planned-for characteristics. Engineering is not the process of building a perfect system with infinite resources. Rather, engineering is the process of economically building a working system that fulfills a need. Without measurable usability specifications, there is no way to determine the usability needs of a product, or to measure whether or not the finished product fulfills those needs. If we cannot measure usability, we cannot have Usability Engineering. Usability Engineering has the following steps:
>
> 1. Define usability through metrics,
> 2. Set planned levels of usability,
> 3. Analyze the impact of design solutions,
> 4. Incorporate user-derived feedback, and
> 5. Iterate until the planned usability levels are achieved."

Because poor usability is costly and good usability can mean increased revenue, usability engineering is cost-justifiable [20][26][38][39][49][60]. *User-Centered Design* (UCD) [19][53][54][55][56][57][58][59][66] is currently the main methodology adopted in usability engineering to address the software usability issue.

2.3 Website usability engineering

Websites are mainly web-based software. Because websites have their own features (*See Chapter 4 for more details*) that separate them from traditional software and the number of existing and potential websites is huge, special efforts [39][61][62][63][64][65][66] have been made to specifically address the usability issues of websites.

Heuristic usability evaluations guidelines for websites [67][68][69][70] have been developed. [67] and [70] are two such examples.

In [67], Keevil collected a heuristic checklist that was organized into "usability categories or metrics". A designer or end user can choose from it the categories and items that are believed applicable to a target website, and then ask each of those chosen items as a question and answer "Yes" or "No" according to his/her experience of the target website. The total number of "Yes" divided by the total number of chosen items is the target website's "Usability Index" (in percentage).

Similarly, in [70], Nielsen suggested a list of 113 heuristic guidelines that are focusing on usability of website homepages. A designer or end user can choose from it

the heuristic guidelines that are believed applicable to a target website's homepage and count each one as "0, 1/2, 1" respectively for no-compliance, partial-compliance and full-compliance according to his/her experience of the target website's homepage, and then divide the final count by the total number of applicable guidelines: if "the usability compliance rate" is greater than 80%, the target website's homepage is in "good shape but may need a few minor fixes"; if between 50 to 80%, "bad enough and start a redesign"; if less than 50%, "abandon it and start over from scratch".

Automatic website usability evaluation tools [71] have also been developed. These tools can track a user's time, pages requested, errors occurred, response time, and traffic information, etc. They are most effective in navigation analysis, webpage level usability evaluation, and standards and guidelines review. The suggested webpage level usability metrics can be found in [72][73][74][75][76]. Chi et al [77][78] developed a simulating system to simulate a real user's navigating behavior based on information scent on the pages of a website, so the usability of a website can be evaluated without having to use a real user. This approach is skeptical, because usability is more of a user experience issue than just following links and counting the number of clicks. For the same reason, automatic website usability evaluation tools should only be used to facilitate usability evaluation but never to substitute user-based usability testing.

Websites inherently fit for *remote usability testing* [79][80][81][82][83]. World Wide Web Consortium (W3C)'s website [4] provides information about Federal and other

web accessibility standards, evaluation tools, filter and transform tools, repair tools, markup validator and other validators.

2.4 Related work and our methodology's potential contributions

2.4.1 Current practice in measuring usability

Whether or not having a complete, systematic, and reasonable set of quantitative metrics has long been considered as an indicator of an academic discipline's maturity. In this regard, usability engineering should be of no exception. But on this front, it has to be admitted that usability researchers have encountered big challenges. This fact can be clearly seen in [84]. In order to have a better understanding of the research findings in [84], it is necessary first to have some basic knowledge about the current usability defining frameworks on which the existing usability metrics are based.

There are presently three major usability defining frameworks from which most of the existing usability metrics have originated. The first one, which is also the most influential one, is the ISO 9241 standard for usability [36], which suggests the following three aspects of usability to be measured: *effectiveness* (which is further defined by *accuracy* and *completeness*), *efficiency*, and *satisfaction*; The second one is Shneiderman's usability definition [28][29][30], which recommends measuring *time to learn*, *speed of performance*, *rate of errors by users*, *retention over time*, and *satisfaction*; The third one is Nielsen's usability definition [20], which recommends measuring

learnability, *efficiency*, *memorability*, *rate of errors*, and *satisfaction*. Although each of the above three major usability defining frameworks claims that it defines usability, they differ in *what usability aspects* or *dimensions* usability consists of and *how* different usability metrics are categorized into corresponding usability aspects or dimensions.

With these definitions' differences being put aside, it's not difficult to find out that practices that strictly follow them will suffer in two aspects. Firstly, *direct* and *specific* measurements of the usability of *interaction process* will fall short. Because of this problem, in practice it is very hard to link each particular usability problem discovered to specific part of a particular interaction process. In other words, *the practicing process needs to be more formative*. Secondly, none of these definitions define what the overall usability of a target system is, and, to what extent and in which way each usability aspect affects the overall usability of the target system. Because of this problem, in practice, the overall usability of a target system over its life-cycle, the overall usability between different systems or different versions of the same system cannot be meaningfully compared. In other words, *the practicing process needs to be more summative*.

The research by Hornbæk in [84] chose ISO 9241 as its foundation. Hornbæk reviewed 180 usability studies that were published in core HCI journals and proceedings in recent years as to how the different usability measures or metrics were used in them. He critically concluded that measures of the quality of outcome of interaction were used in only 16% of the studies; measures of interaction process had not been given separate

attention; measures of usability over time were very rare; the measurement of satisfaction seemed to be in a state of disarray; and "despite more than 20 years of research into usability, current practice in measuring usability suggests that choosing usability measures is difficult". He further suggested that some of the above problems, for example, the lacking of measures that focus on *the quality of outcome of interaction* and *interaction process*, originated from the limitations of the ISO 9241 standard for usability, and thus, the ISO 9241 standard need to be improved.

2.4.2 Measuring usability in a single score

As stated in 2.4.1, a big problem with following the existing three major usability defining frameworks is that the overall usabilities of a target system over its life-cycle, the overall usabilities between different systems or different versions of the same system cannot be meaningfully compared.

In order to solve this problem, it is necessary to combine the different usability aspects into one single overall usability measure. This practice is called summative usability evaluation. Although there are few universal and convincing summative usability evaluation methods that can be used to evaluate and compare usability across systems, in this section, several existing summative usability evaluation methods will be briefly reviewed.

In [73], Babiker et al presented a metric for evaluating usability of hypertext systems. First, their hypertext usability metric was based on three attributes that were common in any hypertext system: access and navigation, orientation, and user interaction. Further, each of the three attributes was computed based on user performance time, key stroke time, and error rate. Finally, the overall usability — the metric — was computed through a weighted formula to combine the three attributes into a single measure.

As introduced in 2.3, Keevil [67] proposed a method to assess website usability, and Nielsen [70] proposed a method to assess the usability of a website's homepage. The two methods roughly assess usability based on their respective heuristic guidelines.

There are many questionnaire-based methods, such as SUMI [85][86], CSUQ [87][88], CUSI [89][90], MUMMS [91], PSSUQ [92][93], QUIS [94][95], SUS [41], WAMMI [96], etc., that claim to be able to assess the overall usability of a system by a single measure based on users' perception of the usability of the system. Some of these questionnaires are free, but others are commercial and require a license to use. A common problem with the usability questionnaire-based methods is that except providing a subjective global assessment of system usability, they cannot identify specific usability problems.

McGee [97] proposed a usability measurement method called Master Usability Scaling (MUS), which was based on Usability Magnitude Estimation (UME) [98] and Master Scaling [99]. UME is a subjective measure of usability based on users' perception

of usability. In practice, first, usability engineers provide the users an objective usability definition; then, according to this definition, the users make ratio usability estimates in terms of the usability of reference tasks; and finally, an averaging procedure is used to normalize the ratio usability estimates and form a single ratio scale of usability. In order for all the ratio scales of usability to be comparable across practices, the objective usability definition and the reference tasks used should be consistent among all practices.

Sauro et al [100] proposed a method to "simplify" all usability aspects into "a single, standardized, and summated usability metric (SUM)". In order to solve the problem that different usability aspects are currently measured on different scales, which makes it difficult to summate them into one single usability measure, the statistics unit *sigma* (σ) from Six Sigma is used as the universal unit for all scales. Now that all the different usability aspects are now expressed in sigma as standardized "quality level" percentages (Z-scores), the different usability aspects are deemed not only comparable with each other but also combinable into one single "summated usability metric" through an equal-weighted scheme. For the same reason, the SUM values of different systems are deemed to be comparable with each other [101].

In [102], Gupta and Gilbert proposed a Speech Usability Metric (SUM) to evaluate the usability of spoken language systems. The SUM metric is actually a weighted scheme to combine some usability aspects, for example, user satisfaction, accuracy, task completion time, etc., into a single usability measure.

2.4.3 Usability in User-Centered Design (UCD)

The accomplishments of the usability measurement and evaluation practices reviewed in the above two subsections are very limited. There are two reasons for this comment. First, most of these practices are only aiming at how to evaluate the usability of a single system, and their usability evaluation results normally are not comparable across random systems or even between different versions of the same system. Second, most of these practices have not attempted to address the issue of how to enable end users to specify *upfront* usability requirements for a system. From usability engineering's point of view, the latter is a bigger problem. In this subsection, how this problem is dealt with in UCD will be examined.

As stated before, UCD is currently the main methodology in usability engineering. UCD emphasizes users' center role in software engineering process and incorporates usability engineering activities into the traditional software life-cycle. UCD is said to be more of a philosophy or principle than just a methodology.

Indeed, in contrast to the usability measuring and evaluation practices introduced in the prior two subsections, UCD does try to base some of its usability evaluation on upfront specified usability requirements for a system. In dealing with the problem of user usability requirements specification, it defines the representative and frequently performed tasks (here, a task is defined as "clear, precise, repeatable instructions") of a system as *benchmark tasks*. Example benchmark tasks include common tasks (i.e., those

tasks that are 20% in number, but account for 80% usage in a system), and business- or mission-critical tasks. Once the benchmark tasks of a system are identified, they will be used as usability "measuring-instruments". For each benchmark task, its "interaction design" usability requirements are to be specified in terms of the following aspects:

- *Usability goal*: The high-level objectives for a user class in terms of usability and design of user interaction, for example, "walk-up-and-use" for new users, "power performance" for experts, and "avoiding errors";
- *Usability aspects*: The general usability characteristic to be measured, for example, initial performance, learnability, retainability, and initial impression;
- *Metrics*: The values to be measured, for example, "time to complete task", "number of errors", "frequency of help and documentation use", "time spent in errors and recovery", "number of repetitions of failed commands", "number of times user expresses frustration or satisfaction", and "number of commands, mouse-clicks, or other user actions to perform task";
- *A metric's baseline level*: The starting point to determine a metric's target level, normally coming from the level of user performance of the same or similar measuring-instrument (if available) in an existing system, or a prior version of the same system, or a competitor system, or even from trying out some users on early prototype;

- *A metric's target level*: The minimum acceptable level of user performance, usually an improvement over the metric's baseline level.

To say the least, the above approach to specifying user usability requirements for benchmark tasks is questionable. There are many reasons for this comment, for example:

- Why should the level of user performance of the same or similar measuring-instrument in an existing system be used as the starting point to determine a metric's target level?
- Is the usability of the existing system already so good that the system can be used as a model system?
- What is the exact relationship between the target system and the existing system?
- How much improvement a metric's target level should be made over its baseline level? And why?
- What exactly is the budgetary implication behind each metric's different levels?
- How thoroughly can the chosen metrics measure the usability of a benchmark task?
- What is the overall usability of a benchmark task?
- How much will a metric's particular improvement affect the overall usability of its respective benchmark task?

Because all the above questions have not been addressed appropriately in UCD, the quantitative usability goals set forth for the benchmark tasks can only be said to be an unfounded guesswork. In fact, UCD also admits that the bottom line of usability

requirements specification is that “this is not an exact science” [103][104][105]. But this argument should not be made as the justification for using some guesswork as purported usability requirements.

Actually, besides the above problems, there are some other problems with the UCD approach as well, for example:

- The specified usability requirements cannot be legitimately said that they are user usability requirements because real users normally do not understand them. In fact, in practice it is not real users who specify them.
- Because UCD directly adopts the major usability defining frameworks as its usability definition, UCD suffers the same problems as stated in subsection 2.4.1.
- If UCD directly adopts the summative usability evaluation techniques as have been introduced in subsection 2.4.2, inevitably the problems with those summative evaluation methods will still be present.
- Except the attempt to specify limited usability requirements for benchmark tasks, all other usability issues in a target system are tackled through iterations of sorts of usability reviews, evaluations, and testings by involving end users and/or usability experts. This is not to say that these techniques do not work, or they are not important. Instead, it is just to say that this approach will wrongfully subject the usability of a target system only to the good-will and/or good-luck of designers and usability experts rather than to a *contractual* user usability requirements specification

that is specified upfront by end users and has to be tested against at the end of a development project. In fact, we believe that, it is this kind of poor practices that have caused the situation that the end users have to grapple with many usability problems in many existing systems and this situation is totally unacceptable.

2.4.4 Potential contributions of the proposed methodology

The proposed methodology may solve the above problems. Its principles, details, and validation experiment will be presented in the following chapters. In this subsection, only its main features and potential contributions will be briefly described.

In the proposed methodology, the usability of a system is quantitatively defined in terms of the usabilities of the goal-tasks of the system. In turn, the usability of a goal-task is quantitatively defined in terms of the following 5 major usability aspects: *use interaction process interface and presentation aptness* (*presentation*, for short), *use interaction process aptness* (*interaction*, for short), *efficiency*, *satisfaction*, and *effectiveness*. Further, each major usability aspect is quantitatively defined in terms of its basic use features. In this way, as shown in Figure 2.1, a usability engineering framework is set up, and a structured and fully quantitative definition of usability is established.

In this framework, the usability of a system, the usability of a goal-task, and the 5 major usability aspects of the usability of a goal-task are all called *composite* or *derivative use features* of the system. They are derived, or built up, successively in

reverse order starting from the basic use features. Among the 5 major usability aspects, *presentation* and *interaction* focus on the quality of use interaction, with the former focusing on the quality of the interaction interface and presentation and the latter on the quality of the choreography of the interaction process; *efficiency* focuses on the quality of resource-consumption of use; *effectiveness* focuses on the quality of outcome of use; and *satisfaction* serves as a catch-up bag to capture users' feelings about the quality of all the other general usability facets that are hard to define and not captured by the other 4 major usability aspects, for example, the users' feelings about the quality of a content or the usefulness of a content, etc. Apparently, this framework (*See Chapters 3, 4, and 5 for details*) is clearer and more practical than the vague ISO 9241-11 usability definition.

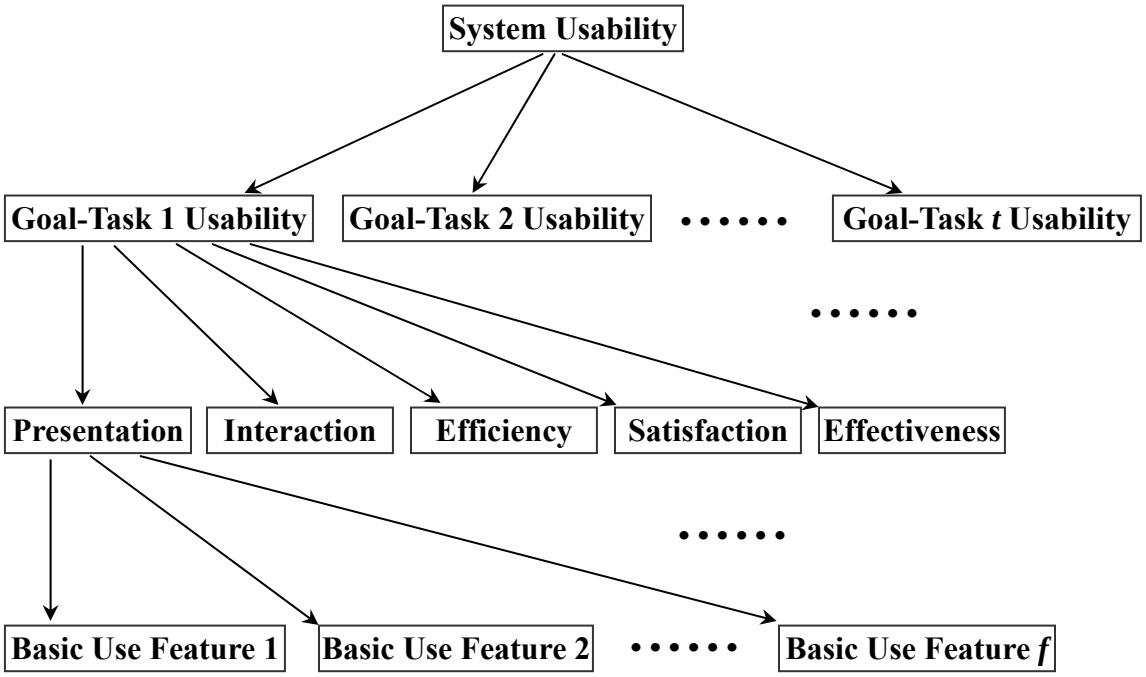


Figure 2.1 Structured and fully quantitative definition of usability

In this methodology, the value of each use feature is expressed as a ratio (in percentage) to measure the perfectness of the use feature (*100% = the best*, and *0% = the worst*), so the values of all use features are inherently and naturally normalized. They are directly comparable with each other not only for the full life-cycle of a product but also across any kinds of products. For example, the usability of a website and the usability of a hammer can be easily compared without any confusion. In other words, this methodology makes the usability of any products inherently comparable with no conversion ever needed. Further, because each basic use feature works like a usability problem probe probing directly into every aspect of use, the usability of a product is directly linked to the root of each usability problem of the product. In other words, this methodology also makes the usability of a product very diagnostic or analytical. All in all, it is fair to say that this methodology is not only both formative and summative but also both analytical and global.

This methodology is also discount usability engineering friendly and scalable. This is supported at least in the following two ways. First, the usability of a system can be estimated by measuring the usability of some selected tasks in the system and then scale the usability results up to the entire-system level. Second, a system's usability engineering practices can be done incrementally and over time. So, this methodology fits any project in terms of scale and budget situation.

The most important contribution of this methodology is that it makes it possible for end users to be able to specify *upfront*, natural, easy to understand, and *contractual* usability requirements for a target system *via* its basic use features. Because of this capability, user usability requirements specification for a system has not only become a reality but also gained equal status with other kinds of user requirements specification for the system. This guarantees that the desired user usability requirements will eventually be satisfied just like other kinds of user requirements have always been.

Apparently, the core ingredients that have made all the above potential contributions possible are the new concept of *use feature* and how the *perfectness* of a use feature in terms of usability is quantified or measured. Among all use features, it is especially worth noting that the new definition of the use feature *efficiency* in this methodology is unique. For details of the above, see Chapter 3.

CHAPTER 3

PRINCIPLE OF THE METHODOLOGY

3.1 Use features

It makes sense that whenever you begin to talk about the usability of a tool, first you must specify the context of its use, the goal for which it is to be used, and how you would expect or want it to be used to achieve the goal. The context of use defines the characteristics of the users and the organizational and physical environments of use. The goal of use defines the intended outcome of use. The "how you would expect or want it to be used" defines what features of the tool you expect or want you could make use of, i.e., the interface and capability *presentation* of the tool, and in what possible procedural orders you expect or want you could make use of those features, i.e., the *interaction* choreography or implementation of the tool.

For example, both humans and lions have hands or paws, but the human hands and lion paws are normally used in different contexts, for different goals, and with different presentations and interactions. Apparently, the usability of either the human hands or the lion paws is very good in their own contexts of use with their own presentations and interactions to meet their own goals, but probably not vice versa.

Any feature of a tool that is essential or significant for the tool's use is called a *use feature* of the tool. A use feature that does not consist of other use features is called a *basic use feature*. A use feature that consists of other use features is called a *composite* or *derivative use feature*. A tool can only be used through the use features it provides.

In order to understand the concept of use feature, let's examine some use feature examples of some familiar tools.

Figure 3.1 illustrates two hammers. Apparently, the two hammers are intended to be used in different contexts and for different goals, and they have different presentations and interactions. Definitely, for each hammer, its context of use, goals of use, presentation, and interaction are all its use features because all these features are essential for its use. But among these use features, the presentation and the interaction of each hammer are both composite use features because they both consist of other use features. For example, the presentation of a hammer consists of at least such component basic use features: the hardness of its hitting surface, the size of its hitting surface, the shape of its hitting surface, the weight of its head, the length of its handle, the shape of its handle, and the stiffness of its handle. It should be pointed out that not every feature of a hammer is a use feature of the hammer. For example, for some aesthetic effects, the hammer on the right in Figure 3.1 has some funny pictures on its head and also some color patterns on its handle, but because these features are not essential or significant for this hammer's use, they are not this hammer's use features.

Figure 3.1 Two hammers

Figure 3.2 shows the homepage of Auburn University TigerMail website. Like the 2 hammers, this homepage also has context of use, goals of use, presentation, and interaction use features, because all these features are essential for its use. For its presentation, at least the following component basic use features can be identified: the theme ratio (i.e., the ratio between the displayed space occupied by the theme of a page and the total displayed content space of a browser); the number of misleading or confusing items; the number of items that have bad readability; the number of distracting items; the number of items that have inappropriate layout or grouping; the number of items that have inconsistent appearances or properties; the number of necessary but missing methods; the number of links that cannot be easily identified to be links; the

number of links that do not follow visitation color-coding; the number of links that are broken; the quality of page help. Apparently, all these component use features are essential for the usability of the homepage's presentation.

Figure 3.2 The homepage of Auburn University TigerMail website

It should be pointed out that, when the usability of a tool is at concern, besides the 4 top level use features of the tool mentioned above (i.e., *context of use*, *goal of use*, *presentation*, and *interaction*), the other 3 top level use features of the tool, i.e., *effectiveness*, *efficiency*, and *satisfaction*, also have to be considered. Effectiveness is the

accuracy and completeness with which users can achieve specified goals by using the tool. Efficiency is the resources that have to be expended in relation to the accuracy and completeness with which users achieve specified goals. Satisfaction is users' feeling about the freedom from discomfort when using the tool and the degree of users' positive attitude toward the use of the tool. Among all these top level use features, while the context of use and the goal of use delimit the boundary of the discussion of the usability of the tool (i.e., the context of use can be considered as a pre-condition of use, and the goal of use can be regarded as an ideal post-condition of use), the rest of them form the body of the definition or evaluation of the usability of the tool.

3.2 Efficiency

In software usability studies, efficiency of a task is normally considered as the amount of time spent on the task by users. But in our opinion, this definition of efficiency is controversial. In order to enlighten this issue up and make it right, in this subsection, cases will be analyzed and our new definition of efficiency of use will be presented.

When measuring the efficiency of a task (or use of a task) is at concern, naturally, obtaining either the "absolute amount of time spent on the task by users" or the "speed" (i.e., the average achievement per unit of time with which users finish the task) seems to be the right way to go. Actually, this is exactly the case in most existing software usability studies, especially the former one.

Let's first take a look at how the "absolute amount of time spent on the task by users" approach fares. Let's assume each task as a straight route literally. In Case 1 on the left of Figure 3.3, let's assume the 2 users, User1 and User2, travel at the same speed *v*. User1 is supposed to travel through the AB route that has length L and User2 through the CD route that has length 10L. Apparently, if User1 takes time *t* from A to B, User2 needs time 10*t* from C to D. Then, which one is more efficient, User1 with time *t* or User2 with time 10*t*? Apparently in this case, the "absolute amount of time spent on the route" cannot be used to tell which one is more efficient, because User1 and User2 have traveled through 2 different routes with different lengths (i.e., 2 different situations) respectively. This approach sounds silly, but it has long been widely used to measure and compare task efficiency. In fact, it is not difficult to tell that both User1 and User2 have the same efficiency, because they have traveled at the same speed.

If the above scenario for Case 1 is not good enough to tell the truth, let's change the scenario a little bit: Let's assume everything else is the same except that User2 would travel at speed 10*v*. In this new scenario, apparently both User1 and User2 will take the same time *t* to reach their respective destinations. So, which one is more efficient, User1 with time *t* or User2 also with time *t*? In this new scenario, the "absolute amount of time spent on the route" approach still sounds silly. In fact, it is easy to tell that User2 is 10 times more efficient than User1.

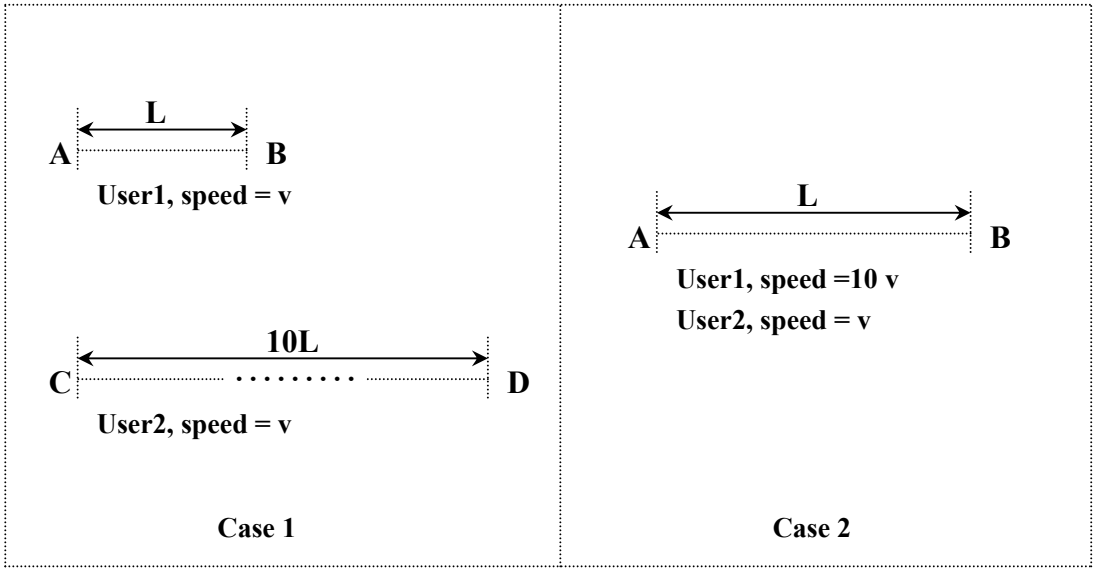


Figure 3.3 The "amount of time" or the "speed"?

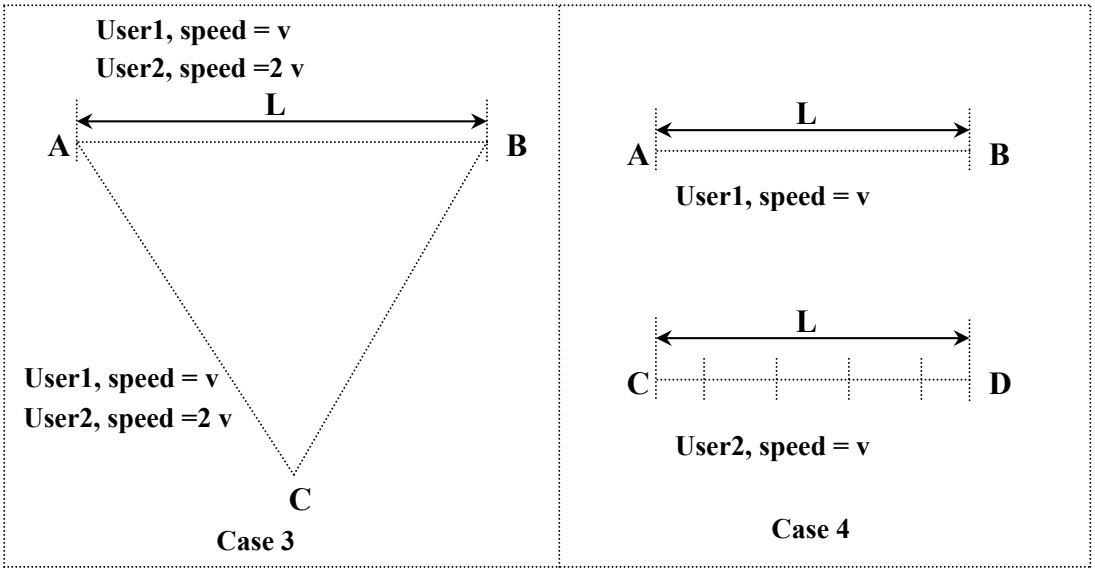


Figure 3.4 The efficiency of route

Now that the "absolute amount of time spent on the task by users" approach does not work, then measuring the "speed" might be the right way to go. Case 2 on the right of

Figure 3.3 illustrates how the "speed" approach might work. In Case 2, let's assume User1 and User2 travel at speeds 10*v* and *v* respectively, along the same route AB that has length L. If User1 takes time *t* from A to B, then User2 needs time 10*t*. So, which one is more efficient, User1 with time *t* or User2 with time 10*t*? Indeed, in this case, the "speed" approach seems to have worked perfectly both from the view of the "absolute amount of time spent on the route" and from the view of the "speed". The reason why the "absolute amount of time spent on the route" approach also seems to have worked in Case2 is because only the same single route is at concern here and there is no intention to compare it with any other routes.

Although the "speed" approach seems to have worked perfectly in Case 2, unfortunately, performing a task is not really the same thing as traveling along a route. It is really difficult to quantify the achievement, or the achievement per unit of time, of a task-performing. Perhaps this is the reason why this approach has rarely found use in existing software usability studies. But if examined further, it can be found that, even if the quantification of achievement of a task-performing were not an issue, the "speed" approach has actually measured the efficiencies of wrong targets, i.e., the efficiencies of users instead of the efficiency of route. Then, which one should have been at concern in the first place, the efficiencies of users or the efficiency of route? Certainly, it should have been the latter rather than the former. In this new light, suddenly it is not difficult to see that the efficiencies of users are not relevant any more, because the same route should

always have the same efficiency no matter who is riding on it. In other words, in Case 2, even though User1 is 10 times faster than User2, the efficiency of route AB is the same for both of them. From this case, we should recognize that it is always the usability of tools rather the ability of users that should be evaluated in a usability study.

So, we need to figure out how to measure the efficiency of route instead of trying to measure the efficiencies of users. Case 3 on the left of Figure 3.4 illustrates how the efficiency of a route could be measured. Let's assume there are 2 routs from A to B, one is AB with length L, and the other is ACB with length 2L. 2 users, User1 and User2, travel at speeds v and $2v$ respectively and each will travel along the 2 routs once at a time from A to B. If User1 takes time t via route AB, then s/he needs time $2t$ via route ACB. Apparently, User2 needs time $0.5t$ via route AB and needs time t via route ACB. Let's define efficiency of route as $\frac{T - T_w}{T}$. Here, T is the total amount of time spent, T_w is the amount of time wasted that has been imposed upon the users by the route. Then for both User1 and User2, the efficiency of route AB is 100% and the efficiency of route ACB is 50%. In fact, for whatever users, the efficiency of route ACB is always half of the efficiency of route AB, because route ACB always forces the users to travel double the length of route AB.

Actually, our new definition of efficiency of route can be applied to any cases or scenarios. If it has been applied to all the scenarios in Case 1 and Case 2 presented above, the efficiencies for both routes AB and CD in Case 1 and for route AB in Case 2 all can

be found to be 100%. Apparently, all the route efficiencies obtained this way are directly comparable with each other regardless the lengths or curvatures of the routes.

Case 4 on the right of Figure 3.4 illustrates a generic scenario showing how our new definition of efficiency of route works. In Case 4, User1 is supposed to travel from A to B via route AB and User2 is supposed to travel from C to D via route CD. Route AB can be considered as a straight route; route CD can be theoretically considered as a straight route but with many crossroads along the way. On route CD, all along it and at each of the crossroads, there is no sign telling any directions or giving any hints. Let's also assume that User2 has no idea that s/he can reach D by traveling the entire way straight forward from C. In other words, route CD is in fact a labyrinth. Although routes AB and CD have the same theoretical length L, the users' experiences travelling along them would be different. Straight route AB does not impose any difficulty on its users, so the users would not experience any wasted time, and its efficiency of route is 100%. In contrast, for route CD, because of its bad usability, users would experience much wasted time that is imposed on them. In order to find the efficiency of route CD, we need to use *Think-Aloud Protocol* [125] to help identify all the wasted times (*See 6.1.6 for a brief introduction*). The following kinds of wasted times along route CD can be expected: 1) at each crossroad, the time wasted on determining which direction to take next; 2) the time wasted on taking detours; 3) the time wasted on forming unnecessary loops. Because of such imposed wasted times, the efficiency of route CD is less than 100%.

Just as stated at the beginning, a *route* in the above discussion is actually a figurative representation of a *task* or *use* of a task. So if *route* is substituted with *task* or *use*, the definition of the efficiency of route presented above is actually our new definition of *efficiency of task* or *use*. The unique advantages of our new definition are that, first, it is true efficiency of task; second, if followed, all the resulted efficiencies of any tasks are guaranteed to be directly comparable with each other regardless the kind and size of a task.

Although identifying the wasted times of a task or use of a task seems daunting, on a high note, it is definitely doable. Keep in mind that all the tasks of any man-made tools are intentionally designed to be as efficient as possible, so how each task should be done or implemented should never be like a blackbox, or a labyrinth, or even a mystery. In other words, all the wasted times of use are identifiable and justifiable.

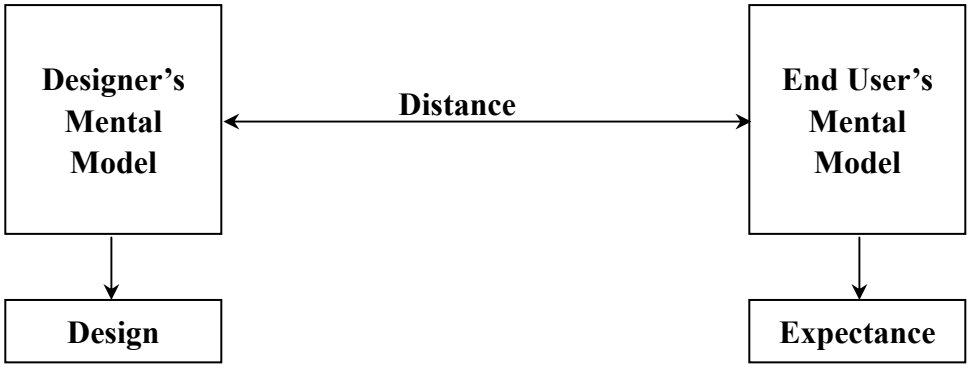
3.3 The origin of usability problems

Why exist there usability problems? In our opinion, usability problems originate from the mental model difference between designers and end users of a product. A mental model is simply a person's view of something experienced, its function and the person's expectance about it. Everybody forms a mental model about everything experienced, and for a variety of reasons, rarely two persons would form exactly the same mental model about one thing. The difference between two different mental models is called mental

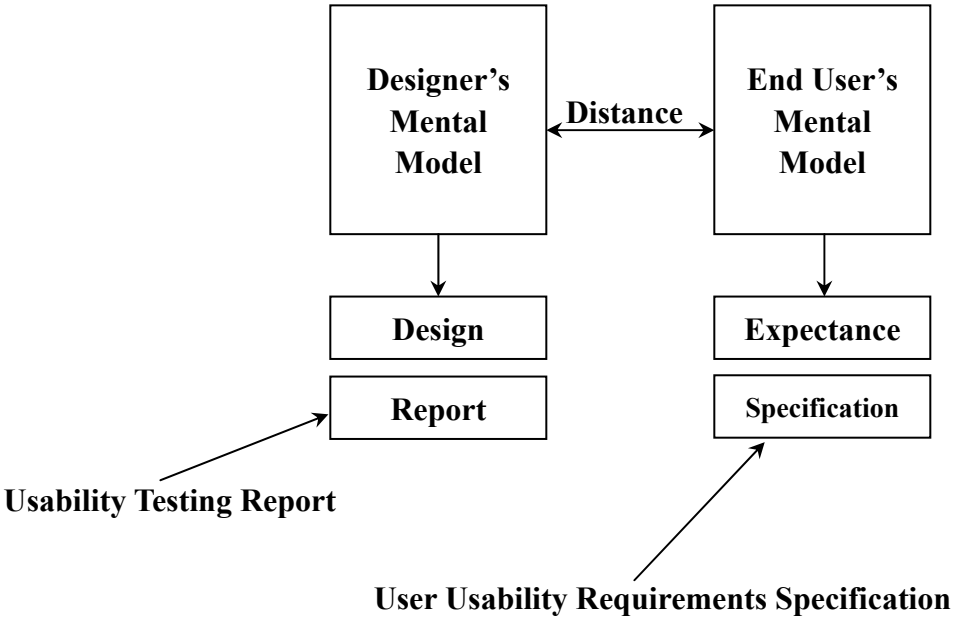
model schism. In this dissertation, only the mental model schism between the designers and the end users of a product is of our interest.

In their relationship, the role of a designer and the role of an end user of a product are not equal. A designer designs a product that is to be used by an end user; an end user has to use a product designed by a designer. The designer's mental model of the product, also called conceptual model, is a model of the product that the designer wants the end user to understand, and the product is simply a concrete embodiment of the designer's mental model of the product. The end user's mental model of the product is forced to match the designer's conceptual model in order for the end user to be able to understand and use the product. Unless the designer is also the end user, there would exist a mental model schism between the designer and the end user of the product. It is this mental model schism that has caused the product's usability problems.

Let's call the width of a mental model schism the *distance* between the two mental models at concern. Figure 3.5 illustrates both the relationship between the mental model of a designer and the mental model of an end user of a product and the change of the distance over the product's usability engineering process. It is fair to say that the bigger is the distance, the bigger are the usability problems. The ideal situation should be that the two mental models overlap as much as possible. Unfortunately, because the cognitive and psychological mechanism behind a mental model is still not well known, exactly what has caused the mental model schism is not clear. Therefore, the distance



The original mental models and the distance



The mental models and the distance after testing-correction and adaptation

Figure 3.5 Mental models' schism and the distance adjustment

between two mental models cannot be directly measured and shortened. Fortunately, the distance can be indirectly measured and shortened by testing end users using the product and then correcting the product's usability problems reported. For example, after

testing-correction, the distance as shown in the upper part of Figure 3.5 can be shortened to the distance as shown in the lower part.

In Figure 3.5, it is also shown that an end user's mental model can somewhat adapt to the designer's mental model. In other words, the end user learns to understand the designer's mental model. But this adaptation should not be expected to be much. This phenomenon simply reflects the fact that an end user's familiarity with a product can improve his/her perception of the usability of the product over time, but real and hard usability problems can not be compensated or eliminated just through the end user's familiarity with the product. For example, a fuzzy label for a button may be misleading or confusing at the beginning, but after a user understands its true meaning through trials and errors, the usability problem caused by the fuzzy label may become negligible to some degree for that particular user. In contrast, if a user is working on a long list, and every time after the user has performed some operation on an item that is a little far away from the beginning, the user is automatically brought back to the very beginning of the list (let's say to the very top of the first page of that list), sooner or later this imposed-upon usability problem may very well drive the user up a wall. The former kind of usability problems are soft-cored usability problems, which usually are tolerable and compensatable by end users; the latter kind of usability problems are hard-cored usability problems, which usually are intolerable and incompensatable by end users. But both soft-cored and hard-cored usability problems are real usability problems.

Meanwhile, it should be noted that the above mentioned end user's mental model adaptation phenomenon also reflects the fact that sometimes the end user even must make mental model transition toward a designer's mental model. An extreme but good example is that end users' old paper-based application mental model is forced to transition to modern computer-based application mental model when their old paper-based application is computerized. Because of the computerization of the old paper-based application, many originally non-existent or impossible concepts and operations in the old mental model now become existent and possible in the modern mental model. But this kind of examples should never be used to justify that designers can count on forcing end users' mental models to transition to solve real usability problems.

In order to eliminate, or at least alleviate, the usability problems of man-made tools, the following points need to be stressed:

- It should be the end users, rather than the designers, who have the center role and the final say on the usability of a designed product. As stated above, because the distance between the mental models of designers and end users cannot be directly measured and shortened, the distance can only be indirectly measured and shortened by testing end users using the product and then correcting the reported usability problems accordingly. In other words, the usability of a product can only be revealed by testing end users instead of just being calculated through some formulas. Testing is to measure the distance case by case; correcting is to shorten the distance case by case.

Because of this reason, end users must be guaranteed to have the center role and the final say in the software engineering process (certainly including the usability engineering process)[4] of a product.

- In order for end users to have the center role and the final say on the usability of a product, they also must have the first say on the usability of the product. In other words, an upfront contractual user usability requirements specification is necessary in making sure that end users will really have the final say on the usability of the product. In fact, if end users do not have the first say, designers can act like they have gotten a carte blanche from end users in the beginning and do not have to worry about being held accountable in the end. It should have long been realized that besides the mental model schism problem, the immunity or amnesty on the usability of products provided to designers by this kind of practices has been a major source of bad usability for many products today. Now, it is time for this loophole to be closed.

Actually, it is not difficult to see that the points emphasized above are consistent with the philosophy or principle advocated by User-Centered Design.

3.4 The solution

As stated in 3.3, in order to eliminate, or at least to alleviate, the usability problems of a designed product, an upfront contractual user usability requirements

[4] In our opinion, usability engineering should always be part of software engineering instead of being a stand-alone discipline as it is now, and each software engineer should also be a usability engineer or expert.

specification for the product is the solution. But, are user usability requirements valid user requirements? Intuitively, the answer is “Yes!”. Life experiences tell us that it should be considered wrong to begin designing and building a house without first knowing how the inhabitants would like to use it. Just imagine the difficulty and the mess that the inhabitants of the house may have to overcome to make things right after the fact (if it is ever possible). Unfortunately, such an answer cannot be found in the current textbooks of software engineering.

Although software engineering emphasizes on the importance of accurate acquisition of user requirements at the very beginning of any project, usability requirements have rarely been considered as valid user requirements that need to be collected from users at the beginning of a project and then tested against in the end. The void of methodology for dealing with usability requirements in regular engineering doctrine seems to have its reasons: first, since a product to be built does not exist, it is hard for its future users to specifically demand upfront how it should be used; second, usability issues seem to be subjective and they are hard to be described objectively and quantitatively. So, it seems impossible for usability requirements to be specified in such an objective and quantitative manner that they can be tested against to see if they have been met.

In fact, the predicament in dealing with usability requirements in software engineering has made “make it work first, then make it better” a practical guidance for

many practitioners. Most practitioners believe that after making a product work first, they can make it better later. But, can this "practical guidance" really work in reality? This doubt can be justified by the following reasoning. It is well known that accurate and complete user requirements are extremely important to the success of any project and it must be carefully dealt with in the very beginning, otherwise the undertakers of the project will be punished heavily later. This hard-learned lesson applies to all user requirements. So, if user usability requirements are supposed to be valid user requirements, the above practitioners are doomed to have a big trouble in the end! In our opinion, this is exactly the situation all the practitioners have been facing.

Covering up the inability of the existing engineering methodologies on usability issues would not make usability issues disappear. What we need is a methodology that can uncover the usability issues and make good usability not just an undetermined gift from the developers but a users' rights that is guaranteed via a contractual requirement that can be implemented, verified, and satisfied.

In fact, because of its contractual power, user functional requirements specification has been a successful controlling factor on the quality assurance of software products. Software functional issues have been solved pretty well through the relentless efforts in software engineering thus far. It is time for software engineering to take on the software usability issues. We believe that the only way out of this usability predicament is that end users are enabled to specify upfront contractual usability requirements in an

explicit and quantitative manner. Just like the role played in the "old" software engineering by functional requirements specification, in this new usability assurance campaign, an upfront contractual user usability requirements specification for any product is the solution.

3.5 Problems with the existing definitions of usability

After we have talked so much about usability, it is wise for us to take a break to re-examine the definition of usability before we proceed further. As mentioned in Chapter 2, different definitions of usability for software systems have been given by Miller [21], Shackel [22][23][24][25], Bennet [26][27], Sheiderman [28][29][30], Nielsen [20], Bevan [31], Löwgren [32], Dix [33], Quesenbery [34][35], etc. In 1998, ISO 9241-11 [36] gave out its own definition of usability. Now, it is the ISO 9241-11 definition of usability that has been recognized as authoritative and become widely adopted.

There are problems with the existing definitions of usability. In this dissertation we only focus on evaluating the problems of ISO 9241-11 definition of usability. It should be noted that the major conclusions about ISO 9241-11 definition of usability also apply to other existing definitions. The ISO 9241-11 defines usability as:

Context of use: characteristics of the users, tasks and the organizational and physical environments.

Goal: intended outcome.

Task: activities required to achieve a goal.

Effectiveness: the accuracy and completeness with which users achieve specified goals.

Efficiency: the resources expended in relation to the accuracy and completeness with which users achieve specified goals.

Satisfaction: freedom from discomfort, and positive attitude to the use of the product.

Usability: 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 pointed out below, in this definition, there exist ambiguities and vagueness that have caused usability problems for itself in practice. We think the ISO 9241-11 definition needs to be improved or extended in at least the following five major aspects.

Firstly, the ISO 9241-11 definition does not differentiate between the goal and task of designers and the goal and task of end users. According to the mental model schism theory presented in section 3.3, we believe that it is important to make a clear differentiation between the two in the definition of usability and the focus should be on the goal and task of the intended end users. The designers can also have goal and task in mind, but their goal and task should try to match as closely as possible those of the intended end users. There is no doubt that the bigger is the difference between the two, the more severe will the resulted usability problems be.

Secondly, the ISO 9241-11 definition does not specify how to measure the *effectiveness*, *efficiency* and *satisfaction*; also, it does not specify how to combine these measures into a single aggregate measure of usability for the entire system. Because of this problem, in practice, it is impossible to quantify usability. The big downside of the inability to quantify usability is that if you cannot measure it, then you cannot manage and control it. In other words, the usabilities of a system over its life-cycle, the usabilities between different systems or different versions of the same system cannot be meaningfully compared; and also, there is no way to determine to what extent and in which way each specific usability aspect affects the overall usability of a system.

Thirdly, the ISO 9241-11 definition defines *efficiency* as an absolute amount of "resources expended in relation to the accuracy and completeness with which users achieve specified goals". As discussed in 3.3, we believe that this is not a good way to define efficiency, because efficiency defined in this way is not comparable across tasks and it does not provide any insight into the quality of the amount of resources expended on a task by users. Here, let's examine this issue in detail. We can assume that for each absolute amount of resource expended, there are at least two portions: one portion that is rightfully expended in relation to the accuracy and completeness with which users achieve specified goals; the other that has been wasted but imposed upon users by an awkward design. Take time as an example. It does not make much sense to measure the absolute amount of time expended on a task as efficiency of that task. The reason is that

any work needs some time to finish, and depending on the complexity of the work, the time needed can be very long or very short. It does not matter how much time has to be expended, but it does matter how much time in the total is rightfully expended. Let's assume the total amount of time expended is T, the portion of T that is the rightfully expended is T_n (time necessary), the other that is wasted (because of mistakes or awkwardness imposed by design) is T_w (time wasted), then, $T_n = T - T_w$. As a measure of efficiency of time expended on a task, $\frac{T_n}{T}$ makes much more sense than T, because: first, it measures the efficiency of task; second, it is comparable across tasks, no matter how big or small a task is; and third, it provides us insight into the quality of total time spent on a task. So, we believe efficiency should be defined as a ratio between a part and the total instead of just an absolute total amount.

Fourthly, the ISO 9241-11 definition does not pay explicit, direct, and specific attention to measuring the usability of a goal-task's human-tool *interaction process* and its *interaction interface*. Actually, from Norman's "stages of action" model [111], which is illustrated in Figure 3.6, it is obvious that *the choreography of the interaction process* and *the presentation (including feedback presentation) of the interaction process* are two key components of a successful *interaction design*. So, if only *effectiveness*, *efficiency*, and *satisfaction* are to be measured as defined in the ISO 9241-11 definition, then the specific usability problems related to interaction process and interaction interface cannot be directly reflected in the usability evaluation. This will make the usability evaluation

too abstract and empty. We believe that the quality of interaction process and the quality of interaction interface need to be *directly* included in the definition of usability, because: first, they are the real sources of most usability problems; second, they determine the easiness of use; and third, they have much to do with users' cognitive feeling about a goal-task. Actually, from usability engineering process's point of view, the usability evaluation stage of a goal-task is also the right time and place to expose detailed usability problems related to the goal-task's interaction process and interaction interface. In fact, interaction design has so much to do with the usability of a product that some usability researchers began to call it user experience design [106][107][108][109].

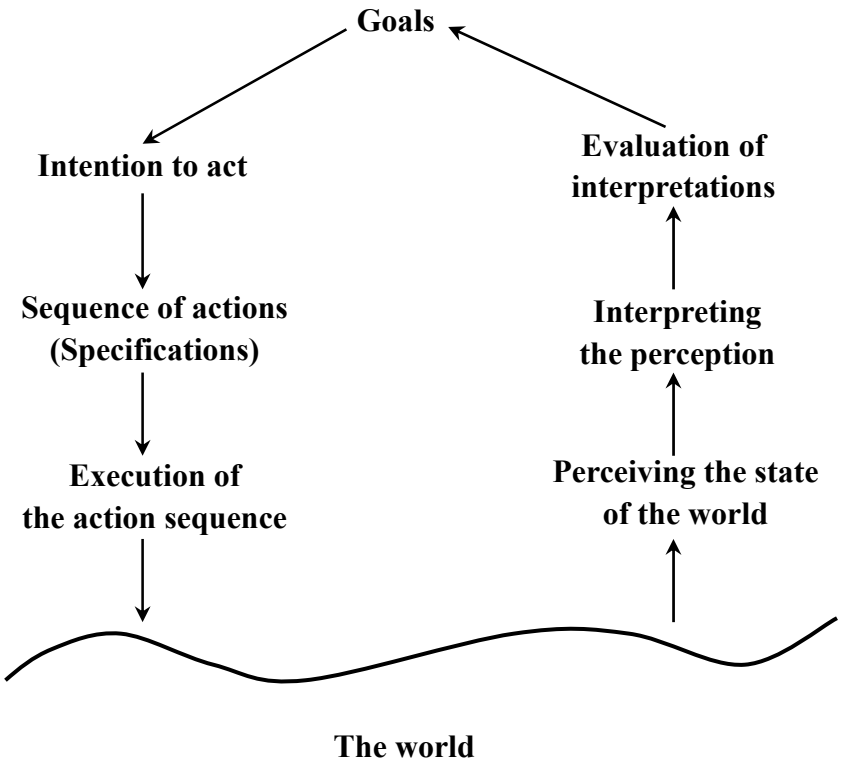


Figure 3.6 Norman's "stages of action" model

Lastly, the ISO 9241-11 definition does not state anything about the relationship between functional correctness and usability. We believe that, on one hand, usability is based on functional correctness of a system, because without functional correctness, a system is not usable at all; on the other hand, a functionally totally correct system can be rendered totally useless by improper usability design. So, in the engineering process of any system, the following 3 points are important: first, there should be usability requirements just like there have been functional requirements; second, usability requirements should be as important as functional requirements have been; third, usability requirements should also be tested against in the end to see if they have been satisfied just like functional requirements have always been. Meanwhile, functional correctness and usability are two totally different aspects of a system. Being functionally correct is a precondition of usability but not part of usability. In other words, functional correctness should be out of concern in usability study. So, for each system, functional requirements and usability requirements are apparently orthogonal to each other.

3.6 Principle of the methodology

In our opinion, any system consists of its goal-tasks, and so does its usability. In fact, as shown in Figure 2.1, we define the usability of a system as a usability hierarchy: The usability of a system consists of the usabilities of its goal-tasks; In turn, the usability of a goal-task consists of its 5 top level composite use features: *presentation*, *interaction*,

efficiency, *satisfaction*, and *effectiveness*; Further, each of the 5 top level composite use features consists of its corresponding component basic use features.

Here, it should be pointed out that the above *presentation* of a goal-task is actually the part of its presentation that only concerns the specific semantics of the goal-task. But, in any system that consists of more than one goal-task, the presentation of each goal-task must also conform to a set of system-level presentation consistency rules that have nothing to do with the specific semantics of any particular goal-task but are critical to the universal look-and-feel and usability of the entire system. In contrast, the part of the presentation of the goal-task that only concerns its conformance to the set of system-level presentation consistency rules is called *the aptness of use universal consistency* (*consistency*, for short). Because *consistency* and *presentation* are actually two facets of the presentation of a goal-task, they share their top level composite use feature status.

Also, it should be pointed out that, in any system, there is always a special system goal-task that is to be used by end users to locate each available end user goal-task in the system. This special system goal-task is called *the system navigation* (*navigation*, for short). Although system navigation is unique in many ways, it is still considered as just another end user goal-task in the system, because it is always the first goal-task that end users have to use when they use the system.

Now, it is time for us to consider how to choose the basic use features for each of the top level composite use features. We have already discussed the concept of use

feature in 3.1, but when we try to determine the appropriate basic use features for a composite use feature, the following two rules need to be considered.

The first rule: each basic use feature should represent the collective quality (in percentage) of all interface items of a goal-task in a corresponding usability aspect. Let's use the homepage of Auburn University TigerMail website in Figure 3.2 as an example to explain why this rule is necessary. Apparently, all the interface items on the homepage are composite use features. Take the "login" button for instance. It is a feature that users have to use when they try to login into their email accounts. According to the definition of use feature, the "login" button is a use feature of the homepage. At the same time, the "login" button has many usability aspects, such as: if it has proper layout or grouping; if it has proper labeling; if it has proper size; etc. Each of these usability aspects is significant for the use of the "login" button. According to the definition of composite use feature, the "login" button is also a composite use feature of the homepage. In fact, this observation applies to all the other interface items on the homepage[5]. Normally, there are many such interface items in any goal-task, and it does not make much sense to just consider each one of them individually. So instead, we only consider the collective quality (in percentage) of all interface items of a goal task in a corresponding usability aspect as an appropriate basic use feature. For example, the *presentation* of the "login into email account" goal-task can have such basic use features like: *percentage of interface items*

[5] Some of the interface items that are seemingly unrelated with the homepage are also the use features of the homepage in the sense that they are essential or significant in influencing or affecting the use of the homepage, for example, distracting end users' attention from their current goal-task, or messing up the theme of the homepage, etc.

that have improper layout or grouping; *percentage of interface items that have improper or misleading labels*; etc.

The second rule: each basic use feature should focus on one aspect of usability problems. Our purpose is to identify and evaluate usability problems. Just as already shown in the above examples, when determining basic use features, we only focus on usability problems.

It should be pointed out that the percentage measurement (as quality level of use feature) has the following advantages:

1. It can be used to measure the distance between two mental models. According to the mental model schism theory, usability problems are caused by the mismatches between designers' mental model and end users' mental model. Let's assume, in a goal-task, the total number of involved items is n, and because of the mismatches of the two mental models, m items present some aspect of usability problem to end users. Then, for that aspect of usability problem, its distance can be expressed as $\frac{m}{n}$, which means m among n items present that aspect of usability problem. In contrast, its usability can be expressed as $\frac{n-m}{n}$ or $1-\frac{m}{n}$, which means $n-m$ among n items do not present that aspect of usability problem. In other words, $1-\frac{m}{n}$ measures the quality level of the goal-task in that aspect of usability. Apparently, the total distance between two mental models of the goal-task can now be considered as the aggregation of the distances of all the basic use features of the

goal-task. Based on this observation, the usability of a goal-task is defined at the end of this section.

2. It can be used as a meaningful severity indicator without having to refer to the total amount involved. For example, it is reported that the fire accidents caused by rats account for 25% of total fire accidents. This report makes perfect sense by just using a percentage number rather than a total number to represent the severity of the fire accidents caused by rats. In fact, in this case, a total number, even if possible, makes much less sense than just a percentage number.
3. It can be used to compare the quality level of things both over time and across kinds. For example, when Dow Jones Index was at 100-point level, a 3-point change meant a 3% up or down from that level. Now, let's assume Dow Jones Index is at 10,000-point level, then a 3% move will mean a 300-point up or down. It does not make any sense to compare Dow Jones Index's daily moves or performances over time in absolute number of points. In contrast, its percentage moves compare meaningfully. Meanwhile, all the markets around the world are now known to be interrelated with each other. Because each market has its own absolute point level, the correlation between the markets can only be manifested by using their percentage changes on a particular day. For example, if China's Shanghai Stock Index made a 5% up move on a Friday, the Dow Jones Index would very probably make a more or less similar move the same day (considering the time difference). When it comes to

usability, the percentage measurements of use features reflect usability levels in such a normalized way that they can be directly compared without conversion. For example, a goal-task with 10 items can have 90% usability by making 9 of its 10 items match end users' expectations in all aspects of usability; while, another goal-task with 100 items can also reach the same usability by making 90 of its 100 items match. Although the efforts (costs) to achieve the same usability level are different, both of them can now be known as having the same level of usability regardless of their kinds and sizes. It should be pointed out that, just as an absolute point level of Dow Jones Index only makes sense when it signifies the scale of the entire market, an absolute item amount, i.e., the scale of the goal-task, only makes sense when it accounts for the total efforts needed to build the goal-task.

4. It is intuitive and easy to understand. For example, it is easy for both designers and users to understand the meaning of a usability requirement like "for goal-task *gt*1, no more than 10% of interface items can have misleading or confusing labels". Because a complete set of use features cover every usability aspect of a goal-task, users can easily specify upfront usability requirements in such a form: a desired quality level (in percentage) for a specific usability aspect of a particular goal-task. Hence, the upfront user usability requirements specification predicament mentioned before will not exist any more. The detailed definitions of all the use features for websites and how to use them to specify usability requirements are presented in Chapter 5.

5. It is independent of design and implementation. For example, different designs or implementations of a goal-task may very probably have different interface items or different numbers of interface items, but each design or implementation can be specified to meet the same level of usability (although definitely you can specify different ones if you really want to).

The following is the structured and fully quantitative definition of the usability framework presented above.

Let's assume P is the top level composite use feature *presentation* of a goal-task, P_1, P_2, · · ·, P_k are P's k component basic use features, and w_{P_1}, w_{P_2}, · · ·, w_{P_k} are these basic use features' weights respectively, $0 \le P_i \le 1$ and $0 \le w_{P_i} \le 1$ for i = 1 · · · k, and $\sum_{i=1}^{k} w_{P_i} = 1$, we define:

$$P = 1 - \sum_{i=1}^{k} w_{P_i} P_i \qquad (3\text{-}1)$$

Similarly, let's assume I is the top level composite use feature *interaction* of the goal-task, I_1, I_2, · · ·, I_h are I's h component basic use features, and w_{I_1}, w_{I_2}, · · ·, w_{I_k} are these basic use features' weights respectively, $0 \le I_i \le 1$ and $0 \le w_{I_i} \le 1$ for i = 1 · · · h, and $\sum_{i=1}^{h} w_{I_i} = 1$, we define:

$$I = 1 - \sum_{i=1}^{h} w_{I_i} I_i \qquad (3\text{-}2)$$

Similarly, let's assume E is the top level composite use feature *efficiency* of the goal-task, E_1, E_2, · · ·, E_q are E's q component basic use features, and w_{E_1}, w_{E_2}, · · ·, w_{E_q} are these basic use features' weights respectively, $0 \le E_i \le 1$ and $0 \le w_{E_i} \le 1$ for i = 1 · · · q, and $\sum_{i=1}^{q} w_{E_i} = 1$, we define:

$$E = 1 - \sum_{i=1}^{q} w_{E_i} E_i \qquad (3\text{-}3)$$

Similarly, let's assume S is the top level composite use feature *satisfaction* of the goal-task, S_1, S_2, · · ·, S_m are S's m component basic use features, and w_{S_1}, w_{S_2}, · · ·, w_{S_m} are these basic use features' weights respectively, $0 \le S_i \le 1$ and $0 \le w_{S_i} \le 1$ for i = 1 · · · m, and $\sum_{i=1}^{m} w_{S_i} = 1$, we define:

$$S = 1 - \sum_{i=1}^{m} w_{S_i} S_i \qquad (3\text{-}4)$$

Similarly, let's assume R (short for *Results*) is the top level composite use feature *effectiveness* of the goal-task, R_1, R_2, · · ·, R_n are R's n component basic use features, and w_{R_1}, w_{R_2}, · · ·, w_{R_n} are these basic use features' weights respectively, $0 \le R_i \le 1$ and $0 \le w_{R_i} \le 1$ for i = 1 · · · n, and $\sum_{i=1}^{n} w_{R_i} = 1$, we define:

$$R = 1 - \sum_{i=1}^{n} w_{R_i} R_i \qquad (3\text{-}5)$$

Similarly, let's assume C_{gt} is the top level composite use feature *consistency* of the goal-task, C_1, C_2, · · ·, C_v are C_{gt}'s *v* component basic use features, and w_{C_1}, w_{C_2}, · · ·, w_{C_v} are these basic use features' weights respectively, $0 \le C_i \le 1$ and $0 \le w_{C_v} \le 1$ for i = 1 · · · *v*, and $\sum_{i=1}^{v} w_{C_i} = 1$, we define:

$$C_{gt} = 1 - \sum_{i=1}^{v} w_{C_i} C_i \qquad (3\text{-}6)$$

Let's assume U_{gt} is the usability of the goal-task, w_P, w_I, w_E, and w_S are the weights of presentation (P), interaction (I), efficiency (E), and satisfaction (S) respectively, $0 \le w_P, w_I, w_E, w_S \le 1$ and $w_P + w_I + w_E + w_S = 1$, we define:

$$U_{gt} = (w_P P + w_I I + w_E E + w_S S)R \qquad (3\text{-}7)$$

(3-7) means:

1. U_{gt} will be 100% only if P, I, E, S, and R all are 100%;

2. If R<1, then R is a discount factor of U_{gt} (especially, if $R = 0$, then $U_{gt} = 0$).

Let's assume U is the usability of the system that consists of *t* goal-tasks. For the *t* goal-tasks, their respective usabilities are U_{gt_1}, U_{gt_2}, · · ·, U_{gt_t}, consistencies are C_{gt_1}, C_{gt_2}, · · ·, C_{gt_t}, and weights are w_{gt_1}, w_{gt_2}, · · ·, w_{gt_t}, with $0 \le w_{gt_i} \le 1$ for i = 1 · · · *t* and $\sum_{i=1}^{t} w_{gt_i} = 1$. Also let's assume the system navigation *nav* has usability U_{nav},

consistency C_{nav}, and weight w_{nav}. Assume w_{gt} is the weight for the combined usability of the *t* goal-tasks as a whole, $0 \le w_{gt} \le 1$, then $w_{nav} = 1 - w_{gt}$. Assume C is the consistency of the entire system. We define:

$$U = (w_{gt} \sum_{i=1}^{t} w_{gt_i} U_{gt_i} + w_{nav} U_{nav}) C \qquad (3\text{-}8)$$

And in (3-8),

$$C = w_{gt} \sum_{i=1}^{t} w_{gt_i} C_{gt_i} + w_{nav} C_{nav} \qquad (3\text{-}9)$$

(3-8) means that the overall comprehensive usability (U) of the system is a composite use feature that combines all the usabilities of its goal-tasks and navigation together, and then takes the consistency of the system into account as a discount factor[6].

Figure 3.7 illustrates the refined usability hierarchy.

3.7 More thoughts on the proposed methodology

As stated before, this structured and fully quantitative usability framework can be applied to any human-tool interaction systems. Their differences lie in the specific basic use features that have to be considered for a particular kind of human-tool interaction system. The advantage of this kind of quantitative usability framework is that no matter what kind of human-tool interaction system it is applied to, all the resulted usabilities are comparable with each other. In other words, the usability of a hammer can be easily

[6] The reason why consistency of system is used as a discount factor is because bad consistency severely affects the overall usability of any system, and it has no reason to exist at all.

compared with the usability of a website. Our concern in this dissertation is to apply it to the usability engineering of websites.

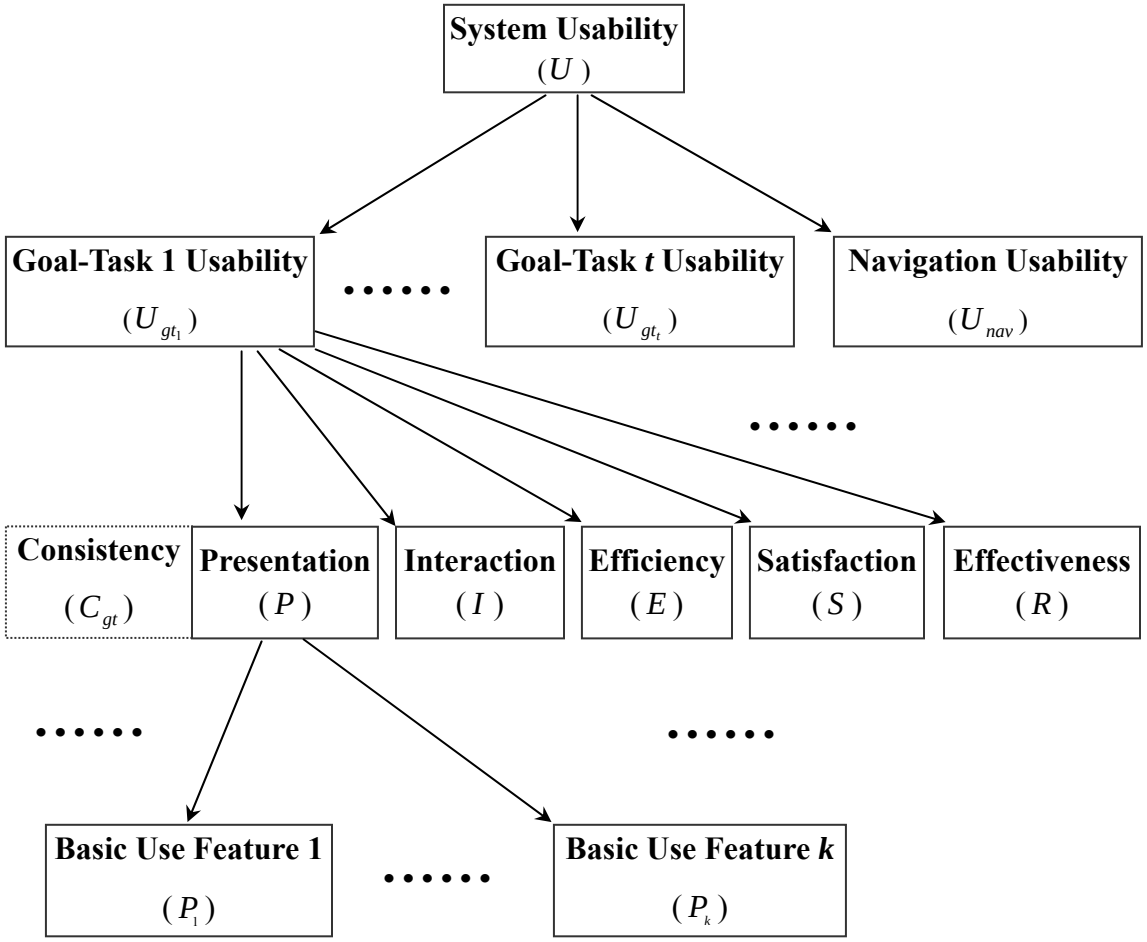


Figure 3.7 Usability hierarchy

As a first endeavor to provide a structured, fully quantitative, and full lifecycle usability engineering framework, this methodology is still at its infancy stage, so all aspects are open for improvement. Because the set of quantitative usability equations presented in 3.6 are subject to optimization and evolution according to their uses in

practice, in order to avoid any future confusion, we will give a version number to each set of quantitative usability equations. The above set of quantitative usability equations can be named *Q*uantitative *U*sability *E*quations *S*e*T* version 1.0 (*QUEST* v1.0). The quantitative usability value of a system should be stated along with the QUEST version number. The format can be like *Usability***:** *U* **(***Version#***)**. For example, a system with 95% usability using version 1.0 of QUEST can be noted as: Usability**:** 95 (QUEST v1.0), or Usability**:** 95% (QUEST v1.0).

A good analogy of this methodology is the methodology adopted for the evaluation of credit worthiness of people: a QUEST number is like a credit score; the structured and fully quantitative definition of usability is like the structured and fully quantitative definition of credit worthiness; the usability testing report is like the actual credit worthiness data collected. Like a credit score, although a sole quantitative usability value of a system is meaningful already, it cannot tell it all. The best way to publish the usability information of a system is to list the following contents in a structured way: the usability value of the system along with its all or at least the major use features; the listed use features' values; their respective allocated weights; and the usability problems associated with each listed use feature. This practice will serve well for the system's usability engineering purpose.

As mentioned before, this methodology is discount usability engineering friendly and scalable. One of the techniques is *goal-task grouping*, i.e., similar goal-tasks can be

grouped together so that only the group level usability evaluations and the group weights will appear in the system level QUEST. In each group, all or selected goal-tasks' usabilities will be evaluated, and a group's usability evaluation is the simple average of the usabilities of goal-tasks evaluated in the group. Another technique is *randomly sampling* or *selectively sampling*, i.e., only random or selected goal-tasks' usabilities will be evaluated. Their usability results would then be scaled up to the entire system level. The scaling up process can be done like this: let's say there are 10 goal-tasks in a system: *gt1* to *gt10* with their weights w_{gt1} to w_{gt10} respectively, and we only choose to usability test *gt1* and *gt2*. After usability testing, we get their usability evaluations: U_{gt_1} for *gt1*, and U_{gt_2} for *gt2*. We can then assume that the entire system just consists of these two goal-tasks, with their new weights of $\frac{w_{gt1}}{w_{gt1} + w_{gt2}}$ and $\frac{w_{gt2}}{w_{gt1} + w_{gt2}}$ respectively.

CHAPTER 4

SOME FEATURES OF WEBSITES

4.1 The general architecture of WWW

Generally speaking, WWW is a Client/Server Model-based application built upon Internet as illustrated in Figure 4.1.

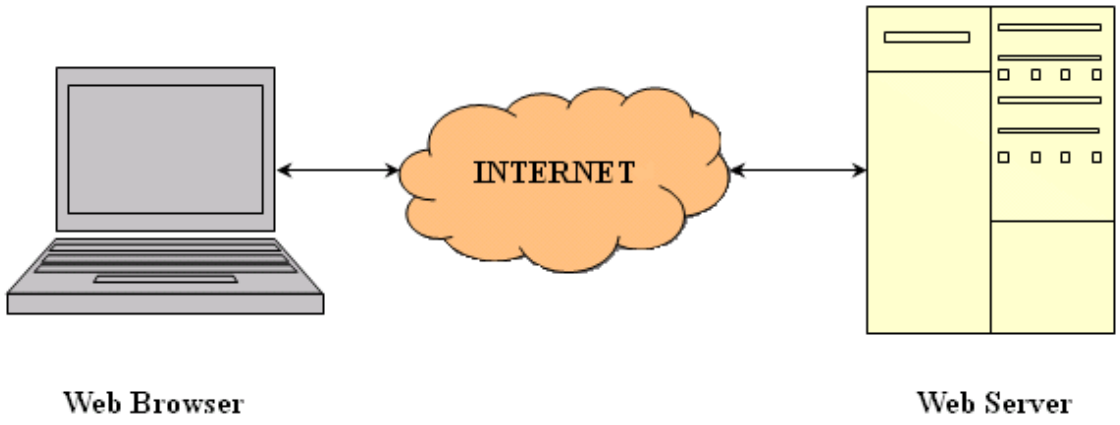


Figure 4.1 The general architecture of WWW

The basic information unit on the WWW is a hypertext document marked up in Hypertext Markup Language (HTML) [122], which is often simply called a webpage. Normally, a webpage contains the following information: content, page layout information, content presentation information, and hyperlinks. Among them, the

hyperlinks are what make the WWW as the Web we know. A hyperlink is described by a Universal Resource Locator (URL) [123], which, besides containing other kinds of information, denotes the address of other information resource on the Internet, such as the address of another webpage, the address of some multimedia information resource, or even just a place within the document itself. Through using URL as hyperlink, all the Internet resources become globally addressable and are contained within a universal addressing space. It is in this simple way that almost all the computerized information resources all over the world have been connected together and formed a worldwide web of information upon the Internet.

Normally, all the webpages are stored in, or upon request dynamically generated by, some website hosted on some web server located somewhere around the globe. Now, there are tens of millions of websites distributed all over the world. To use the web, a web user, via a web client, normally a web browser located in a local computer (*in this dissertation, we are only interested in web browsers as web clients*), connects to a web server and requests a webpage; the web server returns the webpage, and the web browser presents or displays it to the web user.

Over the years, many client side and server side web technologies have been developed, which keep the web technology evolving at a whizzing speed, overwhelming even many professional web application developers. Some technologies just come and go, but some stay over time. Among those useful extensions is the three-tiered or n-tiered

application architecture. From website usability's point of view, as to what web technologies to use, on the client side, apparently maximum cross-browser supportability should always overrule. In this spirit, we assume cross-browser supportability is beyond the scope of this dissertation. On the server side, regardless of the technologies used, webpage request response time should always be our top concern.

4.2 Some features of websites

Nowadays, websites have become the major means of information and services delivery over the Internet. Most websites have been built mainly for two purposes: 1. information publication and retrieval; 2. Web-based functional services (applications) delivery. Compared to traditional software, websites have distinctive features. Because these important unique features are critical to understanding our approach to website usability engineering, we will introduce them one by one.

4.2.1 Unification of functional services and contents

Generally speaking, no matter what its purposes are, any website is a nonlinear composition of functional service items and browsing items that are presented on webpages and linked together through hyperlinks. Here, a functional service item means a complete piece of functional service; a browsing item means a complete piece of content. Different from traditional applications that strictly differentiate between

commands (i.e., functions) and documents (i.e., data), websites do not distinguish between functional service items (i.e., implementations or presentations of commands or functions) and browsing items (i.e., implementations or presentations of documents or contents) at all. Both kinds of items are presented in the same way as a series of web pages. Because of this phenomenon, on the surface, a functional service item and a browsing item are not that different on a website. This distinctive feature of websites is called the *all-purpose composability*[7] of the World Wide Web, which makes websites extremely flexible and has proved to be a major beauty and strength of websites.

As a result, on the Web, both functional service items and browsing items can be given a unified term: *C*onceptually *I*ndependent *C*omposing *I*tem (*CICI*) (read as *kick*), which means they are conceptually independent, complete, and indivisible. Typical instances of CICI's are things like: a complete online article or book, a complete web-based transaction, etc. Abstractly, each CICI consists of a series of webpages that are put together for a purpose. On each of its webpages, besides its presentation, a CICI can be associated with methods, which are normally presented as links or buttons, that can be used by users to operate on it. In essence, a CICI is simply a designed goal-task of a website. In contrast to normal CICI's, the navigation system of a website is a unique designed goal-task that is solely for gluing the entire website together and providing a means of navigation between and beyond the CICI's of the website to end users.

[7] It should be kept in mind that abusing the "all-purpose composability" of a website can severely damage its usability.

4.2.2 Contentized navigation

The navigation system of a website is analogous to the menu system of traditional software. Because of the World Wide Web's "all-purpose composability", a website's navigation (organization) architecture can often be so *contentized* or expanded that the traditional clear distinctions between navigating items (menus) and data (real contents) become blurred or even disappeared. For example, each "menu" of a website can be a very descriptive or verbose webpage, which resembles or even mingles in the presentation of real content. Even so, the main purpose of a website's navigation is still to provide an efficient means of reaching the CICI's of the website to end users.

There are two flavors of navigation: fixed navigation and ephemeral navigation. Fixed navigation means each CICI of a website can be directly reached through the website's main navigation. Ephemeral navigation means some CICI's can only be reached through the links embedded in other CICI's. Ephemeral navigation by nature is context-dependent and easy to get lost. Because ephemeral navigation can cause severe usability problems, it should be avoided altogether or be replaced by short-cuts.

An extreme example of contentized navigation on the Web is the sitemap.

4.2.3 Extensive utilization of short-cuts

Because of the World Wide Web's "all-purpose composability" and the rich presentation space of each webpage, visualized short-cuts are extensively used on the

WWW. A short-cut is a redundant alternative navigation method that is provided outside the regular navigation and embedded in some webpage as a convenient way to efficiently reach some CICI on or off the current website. Compared to regular website navigation, short-cuts, on one hand, have the drawback of uncertainty, i.e., it is not guaranteed that a particular short-cut would be there when it is needed; on the other hand, they have the advantage of efficient navigation, i.e., they can extremely shorten the reaching distance of the referenced CICI's. If properly used, short-cuts can provide important alternative methods to efficiently navigate on the WWW. A good usage of short-cuts is to easily provide immediate cross-referencing between CICI's. But just as anything good, abusing short-cuts can also adversely affect the usability of a website.

Although short-cuts and ephemeral navigations look similar, it is important to understand their difference. Short-cuts are intended to provide pure convenience of reaching the referenced CICI's efficiently, and they are redundant alternative navigation methods with no intention to be part of the regular navigation of a website. In contrast, ephemeral navigations provide accesses to some CICI's in such an obscure way that the referenced CICI's are conceptually disconnected from the regular navigation of a website.

4.2.4 High dynamicity and unchanging usability expectance

Websites are extremely dynamic. Some websites can be updated many times a day. The user populations of websites can also be very dynamic: the kinds of users of a

website are simply unpredictable; a specific user may only be interested in a specific small portion or topic of a website; and, some users may only visit a specific website once for their lifetime. However, walk-up-and-use for everybody is a default and unchanging usability expectance for almost all websites.

CHAPTER 5

WEBSITE USE FEATURES

5.1 General terms

Designed Goal (G_d): Designed goal G_d is the outcome of a designed goal-task that is intended for end users to achieve by the designers.

End Users' Goal (G_u): End users' goal G_u is the outcome of a designed goal-task that is anticipated by end users.

Designed Goal-Task: A designed goal-task is a procedural sequence of steps and actions designed by the designers to be taken by end users to achieve the designed goal.

End Users' Goal-Task: An end users' goal-task is a procedural sequence of steps and actions anticipated by end users to take to achieve the end users' goal.

Use: Use is an improvised real execution of a designed goal-task by an end user, and it is a human-tool interaction process that consists of a sequence of use steps and actions taken by the end user to achieve the end user's goal.

Use Feature: A use feature of a goal-task is any feature of the goal-task that is essential or significant for the use of the goal-task. A goal-task can only be used through its use features.

Basic Use Feature: A basic use feature is a use feature that does not consist of other use features.

Composite or *Derivative Use Feature*: A composite or derivative use feature is a use feature that consists of other use features. While each component use feature of a composite use feature measures the perfectness of a particular usability aspect of the composite use feature, the composite use feature measures the comprehensive perfectness of all its component use features in the usability aspect represented by itself.

Distance Of A Use Feature: The distance of a use feature is the distance between the actual value of a use feature in a designed goal-task and the anticipated ideal value of the use feature by end users, and it is expressed as a ratio (in percentage) to measure the *imperfectness* of the use feature in terms of the use feature itself (*100% = the worst*, and *0% = the best*).

Result Of Use (R_{set}): Result of use R_{set} is a use feature that signifies the set of items achieved through a use.

Designed Context Of Use (C_d): Designed context of use C_d is a use feature that signifies the set of quantified or enumerable ranges of characteristics of the end users, the designed goal-task, and the organizational and physical environments that are specified as restrictions of use by the designers. For example, the designed context of use of a (bank account) balance transfer goal-task can be specified as:

C_d = { users { range of gender: **male/female**;

range of age: **≥ 18**;

range of language: **English**;

range of level of expertise: **all levels**;

range of permission: **registered in the system** };

computers { range of hardware configuration: **all**;

range of internet connection: **all**;

range of operating system: **all**;

range of browser: **any**, **with 128-bit cipher strength** }}

Actual Context Of Use (C_a): Actual context of use C_a is a use feature that signifies the set of actual values in a use for those characteristics that are in or at least sufficiently implied in the designed context of use.

Satisfied Context Of Use: When all the actual values of the characteristics in an actual context of use are within the ranges of the corresponding characteristics in the designed context of use, this actual context of use is called a satisfied context of use. If a characteristic in the designed context of use is not applicable in a use, that characteristic can be regarded as satisfied.

5.2 Website goal-task use features

5.2.1 Presentation and its basic use features

The presentation composite use feature of a goal-task measures the comprehensive aptness (in percentage) of all the interfaces and presentations involved in the use. We define the following 9 basic use features for it, each of them measures its *imperfectness* in one particular usability aspect.

P_1, *Confusing-Misleading Interface Items Ratio*: P_1 is defined as P_{1_f}, the number of confusing, misleading, or too-constrictive interface items involved in the use, divided by P_{1_b}, the total number of interface items involved in the use.

Note: A confusing interface item means end users cannot understand it by its label. A misleading interface item means end users misunderstand it by its label. A too-constrictive interface item means it is an input interface item that has a shorter than reasonable input length. The interface items are counted according to the following rules (*unless noted otherwise, these rules apply to other basic use features where interface-item-counting is involved*):

Rule 1: Nested interface items, such as radio buttons, selection lists, etc., should be counted by *nested computation method*, i.e., a whole nested interface item is counted as 1. For example, let's assume a selection list has 10 member items, and among them, one is

confusing or misleading, then the whole selection list should be counted as 1/10.

Rule 2: An interface item and its label are two separate interface items. For example, an input field labeled "First name" should be counted separately from its label.

Rule 3: Interface items on repeated pages should be counted only once.

P_2, *Inappropriate Theme-Ratio Pages Ratio*: Each page should have a theme. The ratio between the displayed space occupied by the theme and the total displayed content space of a browser is the page's theme-ratio. P_2 is defined as P_{2_f}, the number of pages involved in the use whose theme-ratio is less than 65%, divided by P_{2_b}, the total number of pages involved in the use.

Note: Repeated pages should be counted only once (*unless noted otherwise, this rule applies to other basic use features where page-counting is involved*).

P_3, *Methods-Insufficient Pages Ratio*: Each page should provide sufficient necessary methods to end users. For example, in a list of submitted banking transfers, each transfer should have methods to view, edit, or delete it. P_3 is defined as P_{3_f}, the number of pages involved in the use that have insufficient methods, divided by P_{3_b}, the total number of pages involved in the use.

P_4, *Memory-Exacting Pages Ratio*: P_4 is defined as P_{4_f}, the number of pages involved in the use that force end users to accurately remember facts from previous pages in order to finish the actions on the current page, divided by P_{4_b}, the total number of pages involved in the use.

Note: Repeated pages should be counted accumulatively.

P_5, *Distracting Pages Ratio*: P_5 is defined as P_{5_f}, the number of pages involved in the use that have severely distracting extra features, divided by P_{5_b}, the total number of pages involved in the use.

P_6, *Inappropriate Layout or Item-Grouping Pages Ratio*: P_6 is defined as P_{6_f}, the number of pages involved in the use that have inappropriate layout or item-grouping, divided by P_{6_b}, the total number of pages involved in the use.

P_7, *Bad Feedback Pages Ratio*: P_7 is defined as P_{7_f}, the number of pages involved in the use that do not present appropriate feedback to actions performed on the previous page, divided by P_{7_b}, the total number of pages involved in the use.

Note: Repeated pages should be counted accumulatively.

P_8, *No/Bad Page Level Help Pages Ratio*: Each page should provide page level help methods. P_8 is defined as P_{8_f}, the number of pages involved in the use that have no/bad page level help, divided by P_{8_b}, the total number of pages involved in the use.

P_9, *Bad Readability Pages Ratio*: P_9 is defined as P_{9_f}, the number of pages involved in the use that have bad readability, divided by P_{9_b}, the total number of pages involved in the use.

Let's assume the 9 basic use features have equal weights. Then, according to formula (3-1), the aptness of presentation of a goal-task should be:

$$P = 1 - \sum_{i=1}^{9} \tfrac{1}{9} P_i \qquad (5\text{-}1)$$

Figure 5.1 illustrates the relationship between presentation and its basic use features.

As an example, to specify user usability requirement for the presentation of goal-task *gt*1, end users can simply demand that:

gt1's confusing-misleading interface items ratio should be less than 5%;

gt1's inappropriate theme-ratio pages ratio should be less than 5%;

gt1's methods-insufficient pages ratio should be no more than 0%;

gt1's memory-exacting pages ratio should be less than 5%;

gt1's distracting pages ratio should be less than 5%;

gt1's inappropriate layout or item-grouping pages ratio should be less than 5%;

gt1's bad feedback pages ratio should be no more than 0%;

gt1's no/bad page level help pages ratio should be less than 5%;

gt1's bad readability pages ratio should be no more than 0%.

Then, according to formula (5-1), we get:

$$P = 1 - \frac{1}{9}(5\% + 5\% + 0\% + 5\% + 5\% + 5\% + 0\% + 5\% + 0\%) = 96.67\%$$

So, 96.67% is the user usability requirement for the presentation of *gt*1.

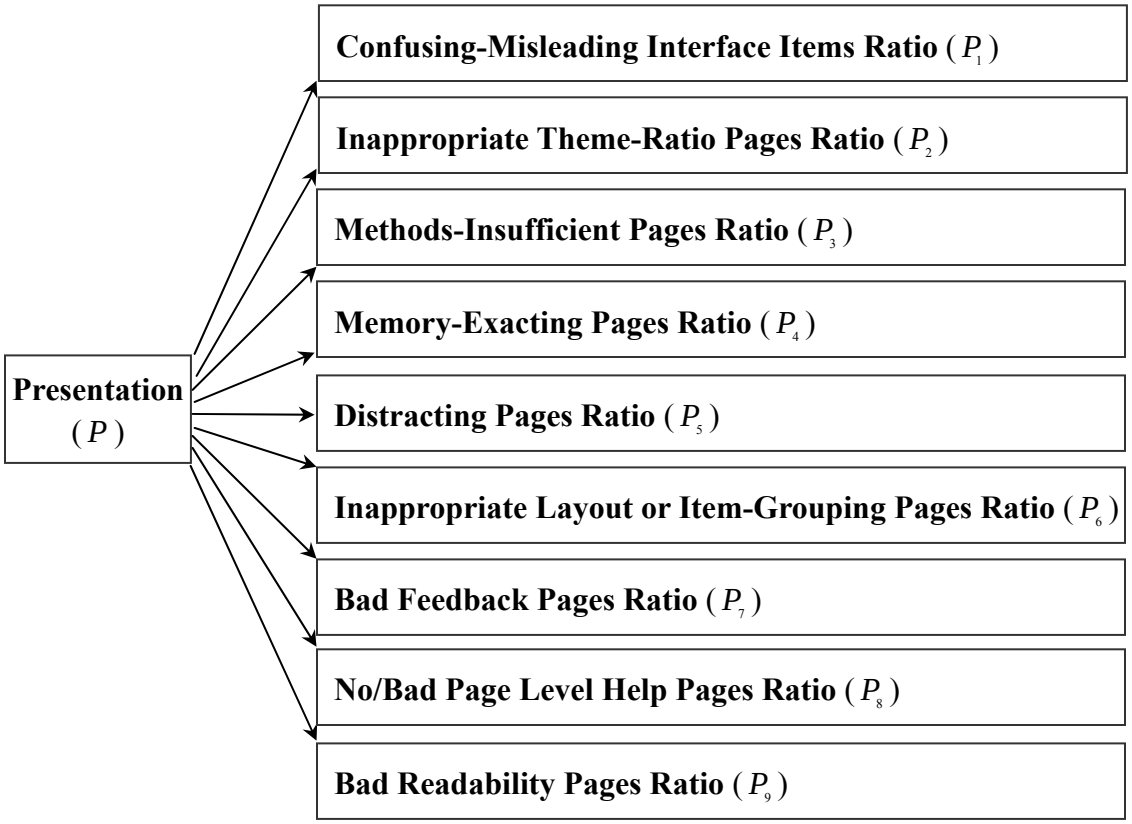


Figure 5.1 Goal-task presentation and its basic use features

5.2.2 Interaction and its basic use features

The interaction composite use feature of a goal-task measures the comprehensive aptness (in percentage) of all the interactions involved in the use. We define the following 4 basic use features for it, each of them measures its *imperfectness* in one particular usability aspect.

I_1, *Mistake-Error Intolerant Actions Ratio*: I_1 is defined as I_{1_f}, the number of actions involved in the use that cannot be corrected, undone, or cancelled, divided by I_{1_b}, the total number of possible actions involved in the use.

Note: An action means an input action or a command method. An action that cannot be corrected, undone, or cancelled means that the action has already caused a failure. In other words, in order to accomplish the goal-task, the goal-task has to be started all over again. Repeated actions should only be counted once.

I_2, *Mistake-Error Actions Ratio*: I_2 is defined as I_{2_f}, the number of actions involved in the use that have caused mistakes or errors DUE TO the design, divided by I_{2_b}, the total number of actual actions involved in the use.

Note: Actions are counted according to the following rules:

Rule 1: Mistake-error actions due to user's own reason should not be counted.

Rule 2: Repeated actions should be counted accumulatively.

I_3, *Imposed-Upon Awkward Actions Ratio*: I_3 is defined as I_{3_f}, the number of actions involved in the use that are unnecessary, unreasonable, awkwardly designed, divided by I_{3_b}, the total number of actual actions involved in the use.

Note: An action that is unnecessary, unreasonable, awkwardly designed means that the action is out of place or order, not straightforward, not logical, not necessary, but is forced upon the user by the design. Repeated actions should be counted accumulatively.

I_4, *Unsuccessful Users Ratio*: I_4 is defined as I_{4_f}, the number of users who cannot finish the goal-task, divided by I_{4_b}, the total number of users who have tried to accomplish the goal-task.

Let's assume the 4 basic use features have equal weights. Then, according to formula (3-2), the aptness of interaction of the goal-task should be:

$$I = 1 - \sum_{i=1}^{4} \frac{1}{4} I_i \qquad (5\text{-}2)$$

Figure 5.2 illustrates the relationship between interaction and its basic use features.

As an example, to specify user usability requirement for the interaction of goal-task *gt*1, end users can simply demand that:

gt1's mistake-error intolerant actions ratio should be no more than 0%;

gt1's mistake-error actions ratio should be no more than 0%;

gt1's imposed-upon awkward actions ratio should be no more than 0%;

gt1's unsuccessful users ratio should be less than 1%.

Then, according to formula (5-2), we get:

$$I = 1 - \tfrac{1}{4}(0\% + 0\% + 0\% + 1\%) = 99.75\%$$

So, 99.75% is the user usability requirement for the interaction of *gt*1.

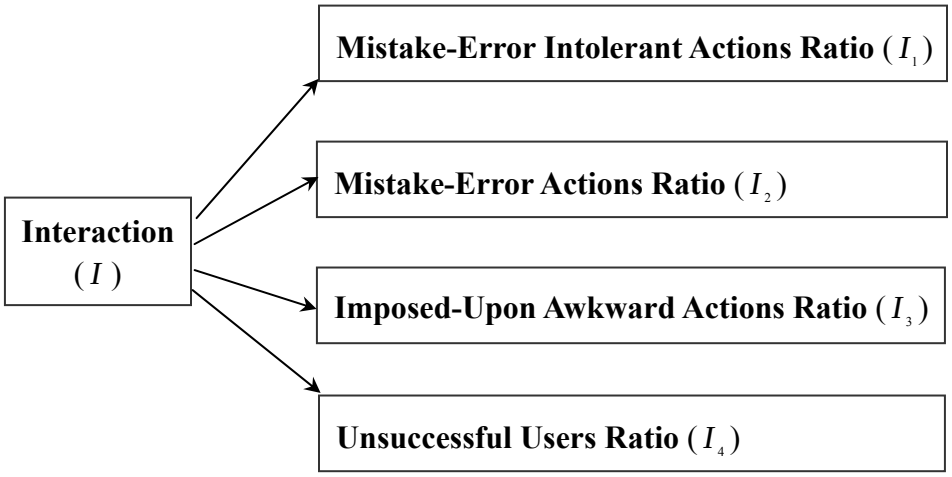


Figure 5.2 Goal-task interaction and its basic use features

5.2.3 Efficiency

For website goal-task efficiency, currently we only consider the time that is spent on a goal-task by a user, so efficiency is a basic use feature by itself. As explained in 3.2, in contrast to the old ways, we define the efficiency of a goal-task, E, as the ratio (in percentage) between the amount of time expended on the goal-task that is perceived necessary and the total amount of time expended on the goal-task. Let's assume the actual

total amount of time expended on a goal-task is T, the amount of time wasted that is imposed upon the user by the design is T_w, then:

$$E = \frac{T - T_w}{T} \qquad (5\text{-}3)$$

It should be noted that the amount of time wasted on a goal-task that is due to users' personal reasons should be excluded from both parts of the above ratio. In order to identify the amount of time wasted that is imposed upon users by design, Think-Aloud Protocol should be used.

As an example, to specify user usability requirement for the efficiency of goal-task *gt*1, end users can simply demand that:

gt1's efficiency should be at least 95%.

5.2.4 Effectiveness and its basic use features

The effectiveness composite use feature of a goal-task measures the comprehensive completeness and accuracy (in percentage) with which users achieve their goals through the use. Theoretically[8], we define the following 2 basic use features for it, each of them measures its *perfectness* in one particular usability aspect.

[8] In practice, the value of effectiveness can be obtained by questionnaires from end users tested. End users can assess the effectiveness of a goal-task based on their accomplishments of uses, and then the average of their assessments can be used as the value of the effectiveness of the goal-task as if it were computed in the way introduced in this section. In fact, this dissertation takes this practical approach in assessing the effectiveness of a goal-task.

R_1, *Result Completeness*: For each item x in an end user's goal G_u, assign a weight to it according to its relative importance among all the expected items in G_u, and the sum of the weights for all the items in G_u equals 1.

Apply the same weight of each item in G_u to its corresponding item in the result R_{set} of a use: only those items that are present both in G_u and R_{set} get their weights, other items in R_{set} get 0 as their weights.

Then, R_1 equals the sum of weights of all the items in R_{set}.

R_2, *Result Accuracy*: For each item that is present in both G_u and R_{set}, if its value in R_{set} is less than its value in G_u, divide its value in R_{set} by its value in G_u, then the result is the accuracy of this item; otherwise, its accuracy is 1.

Then, R_2 equals the weighted sum of the accuracies for all the items in R_{set} that are present in both G_u and R_{set}.

Note: The weight used for each item's accuracy is the same as the weight allocated to that item in the definition of R_1.

Let's assume the 2 basic use features have equal weights. Because both of them are effectiveness's positive basic use features, differently from formula (3-5), we define the effectiveness of a goal-task, R, as:

$$R = \sum_{i=1}^{2} \tfrac{1}{2} R_i \qquad (5\text{-}4)$$

Figure 5.3 illustrates the relationship between effectiveness and its basic use features.

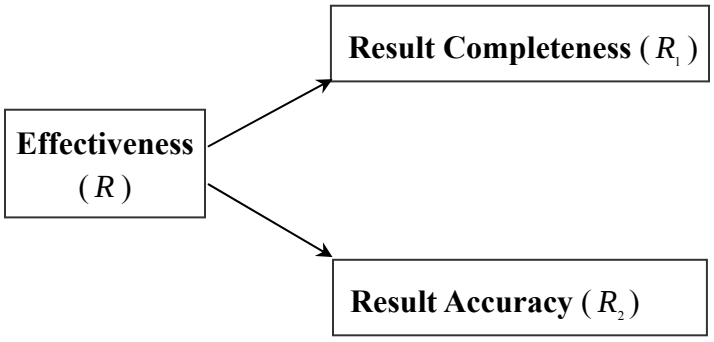


Figure 5.3 Goal-task effectiveness and its basic use features

In practice, the value of effectiveness can be obtained by questionnaires from end users. End users can assess the effectiveness of a goal-task according to their accomplishments of uses. The average of the assessments can then be used as the value of the effectiveness of the goal-task. In this approach, effectiveness is regarded as a basic use feature by itself.

We take the practical approach. As an example, to specify user usability requirement for the effectiveness of goal-task *gt*1, end users can simply demand that:

gt1's effectiveness should be 100%.

5.2.5 Satisfaction

Satisfaction measures the comprehensive degree (in percentage) of users' general feelings of freedom from discomfort in the use and positive attitude toward the use. As one of the top 5 major usability aspects of a goal-task, it serves as a catch-up bag to capture users' feelings about the quality of all the other general usability facets that are hard to define and not captured by the other 4 major usability aspects, for example, the users' feelings about the quality of a content or the usefulness of a content, etc.

In practice, satisfaction is regarded as a basic use feature by itself and obtained from end users through questionnaires. As an example, to specify user usability requirement for the satisfaction of goal-task $gt1$, end users can simply demand that:

gt1's satisfaction should be no less than 90%.

5.2.6 Usability of a goal-task

Usability of a goal-task (U_{gt}) is a composite use feature that measures the comprehensive quality (in percentage) of the goal-task under a satisfied context of use in the following 5 usability aspects: presentation (P), interaction (I), efficiency (E), effectiveness (R), and satisfaction (S).

Let's assume P, I, E, and S have equal weights. Then, according to formula (3-7), the usability of a goal-task should be:

$$U_{gt} = (\tfrac{1}{4}P + \tfrac{1}{4}I + \tfrac{1}{4}E + \tfrac{1}{4}S)R \qquad (5\text{-}5)$$

As an example, using the user usability requirements for P, I, E, S, and R of goal-task *gt*1 (*see the user usability requirements specification examples in sections 5.2.1 ~ 5.2.5 for details*) in formula (5-5), we get:

$$U_{gt} = \frac{1}{4}(96.67\% + 99.75\% + 95\% + 90\%)100\% = 95.36\%$$

So, 95.36% is the user usability requirement for the usability of *gt*1.

5.3 Website navigation use features

The navigation system of a website is analogous to the menu system of traditional software. Although it is unique in many ways, it is just a designed goal-task that is solely for gluing the entire website together and providing a means of reaching the CICI's of the website to end users. Structurally, it is a single-entrance multi-exit functionality. Figure 5.4 illustrates the relationship between the navigation and the normal goal-tasks on a website. In Figure 5.4, the inner nodes are "sub-menus", and the leaf-nodes are normal goal-tasks. Conceptually, Figure 5.4 can be transformed into Figure 5.5 to demonstrate the simplified relationship between the navigation and each goal-task.

Because navigation is the first goal-task that end users have to use when they use a website, its usability is important. Since navigation is just another goal-task, we can still use formula (3-7) to evaluate its usability. But because it is also unique when compared to other normal goal-tasks, the use features defined above for normal goal-tasks must be customized to fit this unique goal-task's special situation.

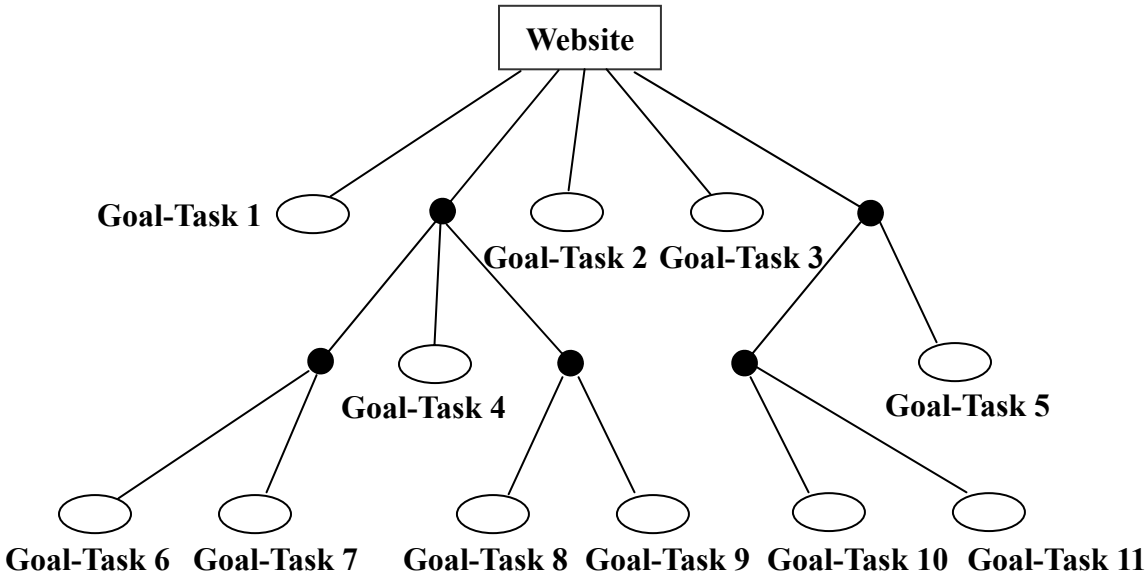


Figure 5.4 Navigation and goal-tasks

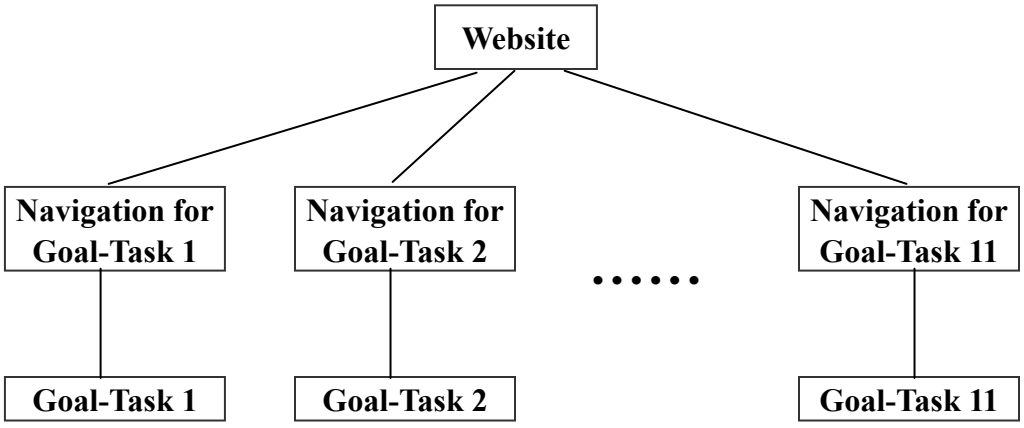


Figure 5.5 Conceptually-simplified navigation and goal-tasks

5.3.1 Presentation and its basic use features

The presentation composite use feature of navigation measures the comprehensive aptness (in percentage) of all the interfaces and presentations in the navigation system. We define the following 5 basic use features on a per goal-task basis for it, each of them measures its *imperfectness* in one particular usability aspect on a per goal-task basis.

P_1^{gt}, *Confusing-Misleading Navigation Methods Ratio*: P_1^{gt} is defined as $P_{1_f}^{gt}$, the number of confusing, misleading, or illegible navigation methods on all the navigation pages involved in the navigation process leading to the desired goal-task, divided by $P_{1_b}^{gt}$, the total number of navigation methods on all the navigation pages involved in the navigation process leading to the desired goal-task.

Note: Navigation methods on repeated pages should be counted only once.

P_2^{gt}, *Inappropriate Theme-Ratio Pages Ratio*: P_2^{gt} is defined as $P_{2_f}^{gt}$, the number of navigation pages involved in the navigation process leading to the desired goal-task whose theme-ratio is less than 65%, divided by $P_{2_b}^{gt}$, the total number of navigation pages involved in the navigation process leading to the desired goal-task.

P_3^{gt}, *Distracting Pages Ratio*: P_3^{gt} is defined as $P_{3_f}^{gt}$, the number of navigation pages involved in the navigation process leading to the desired goal-task that have

severely distracting extra features, divided by $P_{3_b}^{gt}$, the total number of navigation pages involved in the navigation process leading to the desired goal-task.

P_4^{gt}, *Inappropriate Layout or Item-Grouping Pages Ratio*: P_4^{gt} is defined as $P_{4_f}^{gt}$, the number of navigation pages involved in the navigation process leading to the desired goal-task that have inappropriate layout or item-grouping, divided by $P_{4_b}^{gt}$, the total number of navigation pages involved in the navigation process leading to the desired goal-task.

P_5^{gt}, *No/Bad Page Level Help Pages Ratio*: P_5^{gt} is defined as $P_{5_f}^{gt}$, the number of navigation pages involved in the navigation process leading to the desired goal-task that have no/bad page level help, divided by $P_{5_b}^{gt}$, the total number of navigation pages involved in the navigation process leading to the desired goal-task.

Let's assume the 5 basic use features have equal weights. Then, according to formula (3-1), the aptness of presentation of navigation in locating the desired goal-task, P^{gt}, should be:

$$P^{gt} = 1 - \sum_{i=1}^{5} \frac{1}{5} P_i^{gt} \qquad (5\text{-}6)$$

Figure 5.6 illustrates the relationship between presentation and its basic use features.

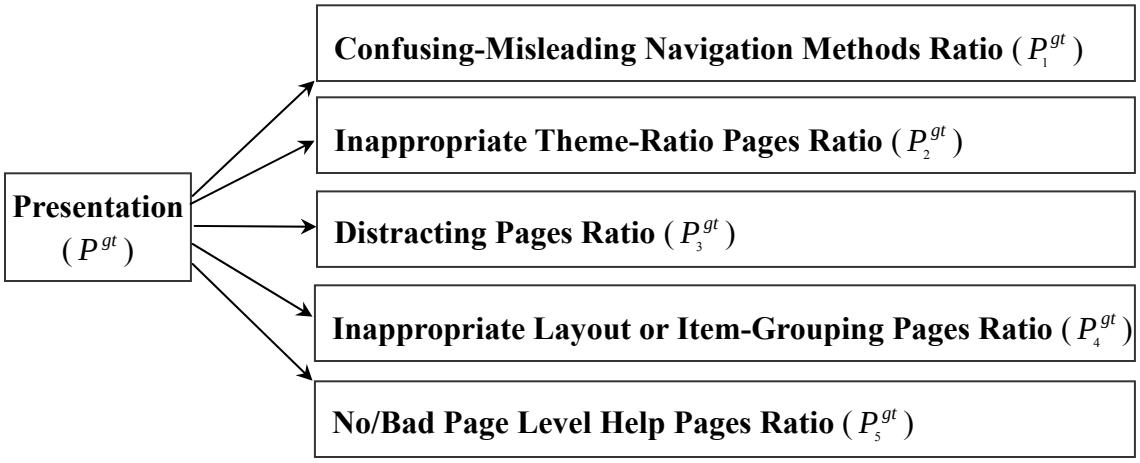


Figure 5.6 Navigation presentation and its basic use features

As an example, to specify user usability requirement for the presentation of navigation in locating goal-task *gt*1, end users can simply demand that:

The confusing-misleading navigation methods ratio in locating gt1 should be 0%;

The inappropriate theme-ratio pages ratio in locating gt1 should be less than 5%;

The distracting pages ratio in locating gt1 should be less than 5%;

The inappropriate layout or item-grouping pages ratio in locating gt1 should be less than 5%;

The no/bad page level help pages ratio in locating gt1 should be less than 5%.

Then, according to formula (5-6), we get:

$$P^{gt} = 1 - \tfrac{1}{5}(0\% + 5\% + 5\% + 5\% + 5\%) = 96\%$$

So, 96% is the user usability requirement for the presentation of navigation in locating *gt*1.

Let's assume (*this assumption holds for the rest of this Chapter*) a website consists of t goal-tasks, w_{gt_1}, w_{gt_2}, · · ·, w_{gt_t} are their weights respectively,

$0 \le w_{gt_i} \le 1$ for i = 1 · · · t, and $\sum_{i=1}^{t} w_{gt_i} = 1$. Assume P^{gt_1}, P^{gt_2}, · · ·, P^{gt_t} are respectively the presentations of navigation in locating these goal-tasks. We define the presentation of the entire navigation system, P_{nav}, as:

$$P_{nav} = \sum_{i=1}^{t} w_{gt_i} P^{gt_i} \qquad (5\text{-}7)$$

If we assume the example website only has 1 goal-task $gt1$, then its weight is 100%. According to formula (5-7), we get:

$$P_{nav} = 100\% \cdot 96\% = 96\%$$

So, 96% is the user usability requirement for the presentation of entire navigation system.

5.3.2 Interaction and its basic use feature

The interaction composite use feature of navigation measures the comprehensive aptness (in percentage) of all the interactions in the navigation system. We only define the following 1 basic use feature on a per goal-task basis for it.

I_1^{gt}, *Unsuccessful Users Ratio*: I_1^{gt} is defined as $I_{1_f}^{gt}$, the number of users who cannot locate the desired goal-task, divided by $I_{1_b}^{gt}$, the total number of users who have tried to locate the desired goal-task.

Apparently the weight for I_1^{gt} is 100%. According to formula (3-2), the aptness of interaction of navigation in locating the desired goal-task, I^{gt}, should be:

$$I^{gt} = 1 - I_1^{gt} \qquad (5\text{-}8)$$

As an example, to specify user usability requirement for the interaction of navigation in locating goal-task *gt*1, end users can simply demand that:

The unsuccessful users ratio in locating gt1 should be 0%.

Then, according to formula (5-8), we get:

$$I^{gt} = 1 - 0\% = 100\%$$

So, 100% is the user usability requirement for the interaction of navigation in locating *gt*1.

Let's assume I^{gt_1}, I^{gt_2}, $\cdots$, I^{gt_t} are respectively the interactions of navigation in locating the *t* goal-tasks. We define the interaction of entire navigation system, I_{nav}, as:

$$I_{nav} = \sum_{i=1}^{t} w_{gt_i} I^{gt_i} \qquad (5\text{-}9)$$

Because the example website only has 1 goal-task *gt*1 (i.e., *gt*1's weight is 100%), according to formula (5-9), we get:

$$I_{nav} = 100\% \cdot 100\% = 100\%$$

So, 100% is the user usability requirement for the interaction of entire navigation system.

5.3.3 Efficiency

Instead of time, efficiency of navigation is better considered in terms of human *physical effort* needed to reach a desired CICI through the navigation architecture of a website. Specifically, the human physical effort means how many levels an end user has to click through the navigation architecture in order to reach the desired CICI. If we name the top level of a navigation architecture as level 1, then we can define the *reaching*

distance of a particular CICI as the level at which the CICI can be located. In other words, a CICI's reaching distance is simply the least number of mouse clicks for the CICI to be reached.

Let's assume a CICI i has an access probability p_i, its reaching distance is d_i, and the total number of reachable CICI's is n, we define the *average probability reaching distance*, D_{ap}, as:

$$D_{ap} = \sum_{i=1}^{n} d_i p_i \qquad (5\text{-}10)$$

In order to have the best efficiency, a website needs to have an optimal average probability reaching distance.

Besides D_{ap}, another factor that can affect the efficiency of navigation is the breadth of a navigation architecture. Breadth, $W_{\max}$, is normally defined as the maximum number of navigation items at the same level of any branch of the navigation architecture.

It is believed that a navigation architecture is most efficient when $D_{ap} = 1$, and any navigation architecture with $D_{ap} \geq 5$ should be avoided [65][112][113][114][115] [116]. It is also believed that $W_{\max}$ has much less effects on the efficiency of navigation than D_{ap} [117][118][119] [120], but it is normally suggested that $W_{\max}$ should not be more than nine[9] [121]. In other words, an efficient navigation architecture should be shallow and wide, but not too wide.

[9] Sometimes, this limitation is not practical on the WWW. Actually, in extreme situations, the number of items on one level of the navigation architecture of some websites can easily run up to the order of thousands or even millions, for example, the topic lists on some forum websites, or the search result lists of web search engines.

According to the above discussion, we define the efficiency of navigation, E_{nav}, as:

$$E_{nav} = 1 - (90\% \cdot \frac{d_e}{4} + 10\% \cdot \frac{w_e}{9}) \qquad (5\text{-}11)$$

In (5-11), $\frac{d_e}{4}$ is the inefficiency caused by D_{ap}, and $\frac{w_e}{9}$ is the inefficiency caused by $W_{\max}$, and,

$$d_e = \begin{cases} D_{ap} - 1; & if D_{ap} < 5; \\ 4; & if D_{ap} \geq 5; \end{cases} \qquad (5\text{-}12)$$

$$w_e = \begin{cases} 0; & if W_{\max} \leq 7; \\ W_{\max} - 7; & if \ 7 < W < 16; \\ 9; & if W_{\max} \geq 16; \end{cases} \qquad (5\text{-}13)$$

As defined, the efficiency of navigation is a basic use feature for the entire navigation system. As an example, to specify user usability requirement for the efficiency of navigation of a website, end users can simply demand that:

The efficiency of the navigation system should be more than 80%.

But, because the example website only has 1 goal-task *gt*1, *gt*1's reaching distance is 1, its use probability is 100%, the breadth of the website navigation architecture is $W_{\max} = 1$, and according to formula (5-10), the average probability reaching distance of the website is $D_{ap} = 100\% \cdot 1 = 1$. Then, according to formula (5-11), we get the example

website's actual efficiency $E_{nav} = 1 - (90\% \cdot \frac{1-1}{4} + 10\% \cdot \frac{0}{9}) = 100\%$. So, the actual efficiency of navigation of the example website is much better than the above user usability requirement for it.

5.3.4 Effectiveness and its basic use feature

The effectiveness of navigation, R_{nav}, is defined as the *reachability* of all the CICI's on a website. We only define 1 basic use feature on a per goal-task basis for it, and this basic use feature happens to be the same single basic use feature that has been defined for the interaction of navigation in 5.3.2. So,

$$R_{nav} = I_{nav} \qquad (5\text{-}14)$$

5.3.5 Satisfaction

Satisfaction of navigation is a use feature of the navigation system that measures the comprehensive degree (in percentage) of users' general feelings of freedom from discomfort in the navigation and positive attitude toward the navigation system. In practice, it is regarded as a basic use feature by itself and is obtained from end users through questionnaires on a per goal-task basis.

As an example, to specify user usability requirement for the satisfaction of navigation in locating goal-task *gt*1, end users can simply demand that:

The satisfaction of navigation in locating gt1 should be no less than 90%.

Let's assume S^{gt_1}, S^{gt_2}, · · ·, and S^{gt_t} are respectively the satisfactions of navigation in locating the *t* goal-task. We define the satisfaction of entire navigation system, S_{nav}, as:

$$S_{nav} = \sum_{i=1}^{t} w_{gt_i} S^{gt_i} \qquad (5\text{-}15)$$

Because the example website only has 1 goal-task *gt*1, according to formula (5-15), we get:

$$S_{nav} = 100\% \cdot 90\% = 90\%$$

So, 90% is the user usability requirement for the satisfaction of entire navigation system.

5.3.6 Usability of navigation system

Usability of navigation system (U_{nav}) is a composite use feature that measures the comprehensive quality (in percentage) of the navigation system under a satisfied context of use in the following 5 usability aspects: presentation (P_{nav}), interaction (I_{nav}), efficiency (E_{nav}), effectiveness (R_{nav}), and satisfaction (S_{nav}).

Let's assume P_{nav}, I_{nav}, E_{nav}, and S_{nav} have equal weights. Then, according to formula (3-7), we get:

$$U_{nav} = (\tfrac{1}{4} P_{nav} + \tfrac{1}{4} I_{nav} + \tfrac{1}{4} E_{nav} + \tfrac{1}{4} S_{nav}) R_{nav} \qquad (5\text{-}16)$$

As an example, using the user usability requirements for P_{nav}, I_{nav}, E_{nav}, S_{nav}, and R_{nav} (*see the user usability requirements specification examples in sections 5.3.1 ~ 5.3.5 for details*) in formula (5-16), we get:

$$U_{nav} = \frac{1}{4}(96\% + 100\% + 80\% + 90\%)100\% = 91.50\%$$

So, 91.50% is the user usability requirement for the usability of the navigation system.

5.4 Website universal consistency use features

As stated before, on any website that consists of more than one goal-task, the presentation of each single goal-task should also conform to a set of website level presentation consistency rules that have nothing to do with the specific semantics of any particular goal-task. These presentation consistency rules, called *presentation universal consistency conventions*, are critical to the universal look-and-feel and usability of entire website. The usability aspect that focuses on these rules' conformation is called the aptness of use universal consistency (*consistency*, for short). As shown in Figure 3.7, consistency shares its top level composite use feature status with presentation. We define the following 6 consistency conventions:

Default Set Of Global Methods Convention: Except pop-up windows, any page on a website must not only display the default set of global methods but also do it consistently. The default set of global methods include: top level navigation methods, homing method, sitemap method, institution information method, security terms method, privacy terms method, etc.

Link Indication & Color-Coding Convention: Without any extra effort, users must be able to tell not only if a link is a link but also if the link has been visited.

Time-Sensitive Content Timestamping Convention: Without any extra effort, users must be able to tell all the necessary timing information of any time-sensitive content.

Page Request Response Time Convention: Excluding any network and users' local machine configuration factors, users' page requests must be responded within a tolerable time limit.

Broken Link Convention: There should be no broken links.

Presentation Consistency Convention: In cases of multiple occurrences of an individual interface item or a group of interface items, except the necessary presentation variations that can be justified, no presentation variation in any shape or form should occur.

5.4.1 Goal-task consistency and its basic use features

The consistency composite use feature of a goal-task measures the comprehensive universal consistency (in percentage) of all the interfaces and presentations involved in

the use. We define the following 6 basic use features for it, each of them measures its *imperfectness* in one particular usability aspect.

C_1, *Default Set Of Global Methods Convention Violation Pages Ratio*: C_1 is defined as C_{1_f}, the number of pages involved in the use that have violated the default set of global methods convention, divided by C_{1_b}, the total number of pages involved in the use.

C_2, *Link Indication & Color-Coding Convention Violation Links Ratio*: C_2 is defined as C_{2_f}, the number of links involved in the use that have violated the link indication & color-coding convention, divided by C_{2_b}, the total number of links involved in the use.

Note: Links on repeated pages should be counted only once (*unless noted otherwise, this rule applies to other basic use features where link-counting is involved*).

C_3, *Time-Sensitive Content Timestamping Convention Violation Content-Items Ratio*: C_3 is defined as C_{3_f}, the number of content-items involved in the use that have violated the time-sensitive content timestamping convention, divided by C_{3_b}, the total number of content-items involved in the use. Because each content-item either follows or violates this convention, C_3 will be either 0% or 100%.

C_4, *Page Request Response Time Convention Violation Pages Ratio*: C_4 is defined as C_{4_f}, the number of pages involved in the use that have violated the page request response time convention, divided by C_{4_b}, the total number of pages involved in the use.

C_5, *Broken Link Convention Violation Links Ratio*: C_5 is defined as C_{5_f}, the number of links involved in the use that have violated the broken link convention, divided by C_{5_b}, the total number of links involved in the use.

C_6, *Presentation Consistency Convention Violation Interface Items Ratio*: C_6 is defined as C_{6_f}, the number of interface items involved in the use that have violated the presentation consistency convention, divided by C_{6_b}, the total number of interface items involved in the use.

Let's assume the 6 basic use features have equal weights. Then, according to formula (3-6), the aptness of consistency of a goal-task should be:

$$C_{gt} = 1 - \sum_{i=1}^{6} \frac{1}{6} C_i \qquad (5\text{-}17)$$

Figure 5.7 illustrates the relationship between consistency and its basic use features.

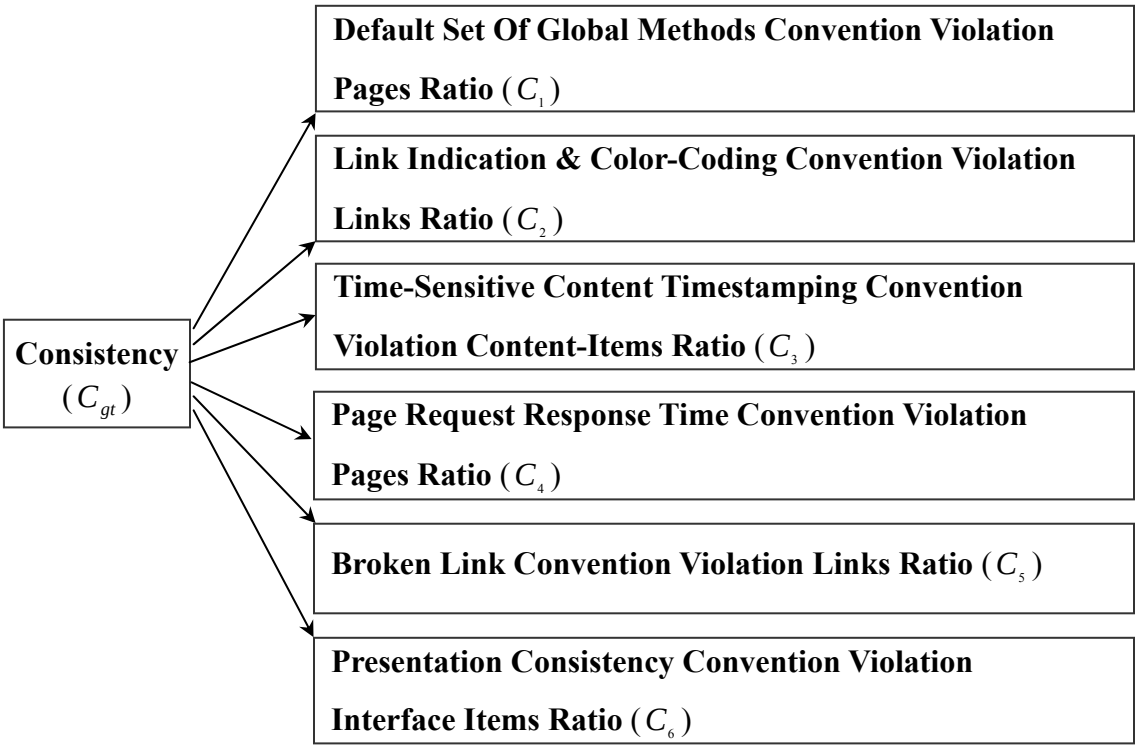


Figure 5.7 Goal-task consistency and its basic use features

As an example, to specify user usability requirement for the consistency of goal-task *gt*1, end users can simply demand that:

gt1's default set of global methods convention violation pages ratio should be less than 5%;

gt1's link indication & color-coding convention violation links ratio should be less than 5%;

gt1's time-sensitive content timestamping convention violation content-items ratio should be 0%;

gt1's page request response time convention violation pages ratio should be less than 5%;

gt1's broken link convention violation links ratio should be less than 5%;

gt1's presentation consistency convention violation interface items ratio should be less than 5%.

Then, according to formula (5-17), we get:

$$C_{gt} = 1 - \tfrac{1}{6}(5\% + 5\% + 0\% + 5\% + 5\% + 5\%) = 95.83\%$$

So, 95.83% is the user usability requirement for the consistency of *gt*1.

5.4.2 Navigation consistency and its basic use features

The consistency composite use feature of navigation measures the comprehensive universal consistency (in percentage) of all the interfaces and presentations in the navigation system. We define the following 5 basic use features on a per goal-task basis for it, each of them measures its imperfectness in one particular usability aspect on a per goal-task basis. The definitions in this section are similar to the ones defined for goal-task in 5.4.1.

C_1^{gt}, *Default Set Of Global Methods Convention Violation Pages Ratio*: C_1^{gt} is defined as $C_{1_f}^{gt}$, the number of navigation pages involved in the navigation process leading to the desired goal-task that have violated the default set of global methods convention, divided by $C_{1_b}^{gt}$, the total number of navigation pages involved in the navigation process leading to the desired goal-task.

C_2^{gt}, *Link Indication & Color-Coding Convention Violation Links Ratio*: C_2^{gt} is defined as $C_{2_f}^{gt}$, the number of links on all the navigation pages involved in the navigation process leading to the desired goal-task that have violated the link indication &

color-coding convention, divided by $C_{2_b}^{gt}$, the total number of links on all the navigation pages involved in the navigation process leading to the desired goal-task.

C_3^{gt}, *Page Request Response Time Convention Violation Pages Ratio*: C_3^{gt} is defined as $C_{3_f}^{gt}$, the number of navigation pages involved in the navigation process leading to the desired goal-task that have violated the page request response time convention, divided by $C_{3_b}^{gt}$, the total number of navigation pages involved in the navigation process leading to the desired goal-task.

C_4^{gt}, *Broken Link Convention Violation Links Ratio*: C_4^{gt} is defined as $C_{4_f}^{gt}$, the number of links on all the navigation pages involved in the navigation process leading to the desired goal-task that have violated the broken link convention, divided by $C_{4_b}^{gt}$, the total number of links on all the navigation pages involved in the navigation process leading to the desired goal-task.

C_5^{gt}, *Presentation Consistency Convention Violation Interface Items Ratio*: C_5^{gt} is defined as $C_{5_f}^{gt}$, the number of interface items on all the navigation pages involved in the navigation process leading to the desired goal-task that have violated the presentation consistency convention, divided by $C_{5_b}^{gt}$, the total number of interface

items on all the navigation pages involved in the navigation process leading to the desired goal-task.

Let's assume the 5 basic use features have equal weights. Then, according to formula (3-6), the consistency of navigation in locating the particular goal-task should be:

$$C_{nav}^{gt} = 1 - \sum_{i=1}^{5} \frac{1}{5} C_i^{gt} \qquad (5\text{-}18)$$

Figure 5.8 illustrates the relationship between consistency and its basic use features.

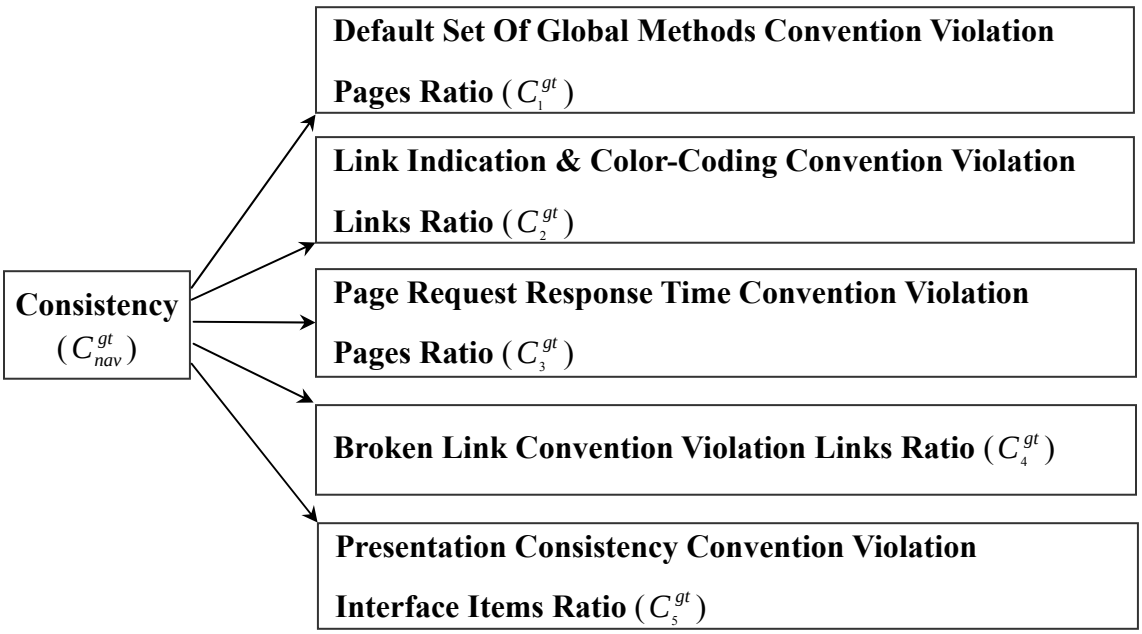


Figure 5.8 Navigation consistency and its basic use features

As an example, to specify user usability requirement for the consistency of navigation in locating goal-task *gt*1, end users can simply demand that:

The default set of global methods convention violation pages ratio in locating gt1 should be less than 5%;

The link indication & color-coding convention violation links ratio in locating gt1 should be less than 5%;

The page request response time convention violation pages ratio in locating gt1 should be less than 5%;

The broken link convention violation links ratio in locating gt1 should be less than 5%;

The presentation consistency convention violation interface items ratio in locating gt1 should be less than 5%.

Then, according to formula (5-18), $C_{nav}^{gt} = 1 - \frac{1}{5}(5\% + 5\% + 5\% + 5\% + 5\%) = 95\%$. So, 95% is the user usability requirement for the consistency of navigation in locating *gt*1.

Let's assume $C_{nav}^{gt_1}$, $C_{nav}^{gt_2}$, · · ·, and $C_{nav}^{gt_t}$ are respectively the consistencies of navigation in locating the *t* goal-task. We define the consistency of the entire navigation system, C_{nav}, as:

$$C_{nav} = \sum_{i=1}^{t} w_{gt_i} C_{nav}^{gt_i} \qquad (5\text{-}19)$$

Because the example website only has 1 goal-task *gt*1 (i.e., *gt*1's weight is 100%), according to formula (5-19), we get:

$$C_{nav} = 100\% \cdot 95\% = 95\%$$

So, 95% is the user usability requirement for the consistency of entire navigation system.

5.4.3 Website consistency

As defined in formula (3-9), the consistency of a website, C, is a composite use feature that combines all the consistencies of the goal-tasks and navigation together, and it measures the comprehensive universal consistency (in percentage) of entire website.

Let's assume the weight for the combined goal-tasks of the example website be $w_{gt} = 90\%$, then the weight for the navigation system be $w_{nav} = 1 - w_{gt} = 1 - 90\% = 10\%$ (*this assumption holds for the rest of this Chapter*). Because the example website only has 1 goal-task *gt*1, then *gt*1's weight is 100%. Using these weights and the user usability requirements for C_{gt} and C_{nav} (*see the user usability requirements specification examples in sections 5.4.1 and 5.4.2 for details*) in formula (3-9), we get:

$$C = 90\%(100\% \cdot 95.83\%) + 10\% \cdot 95\% = 95.75\%$$

So, 95.75% is the user usability requirement for the consistency of entire website.

5.5 Website usability

As defined in formula (3-8), the overall usability of a website (U) is a composite use feature that combines all the usabilities of the goal-tasks and navigation system together, and then takes the consistency of the website into account as a discount factor. It measures the comprehensive usability (in percentage) of entire website.

Using the user usability requirements for U_{gt}, U_{nav}, and C (*see the user usability requirements specification examples in sections 5.2.6, 5.3.6, and 5.4.3 for details*) in formula (3-8), we get:

$$U = (90\%(100\% \cdot 95.36\%) + 10\% \cdot 91.5\%)95.75\% = 90.94\%$$

So, 90.94% is the user usability requirement for the overall usability of the entire example website.

5.6 User usability requirements specification

As shown throughout this Chapter, this methodology supports upfront, explicit and specific, quantitative user usability requirements specification. In fact, if all the weights, use probabilities, basic use features, and derivative use features for all the goal-tasks including the navigation system are put together in a structured way, a simple, easy, straightforward, and upfront quantitative user usability requirements specification for a website is perfectly done.

The beauty of this methodology is that not only can the derivative use features give us full quantitative sense about every aspect, including the overall aspect, of the usability of a website, but also can each individual basic use feature independently work to its fullest quantitative degree to make sure that the usability of the website will be achieved in the end.

CHAPTER 6

VALIDATION EXPERIMENT

6.1 Introduction

6.1.1 Design

The principle of the validation experiment is to prove that the proposed methodology, QUEST, has stronger website usability evaluation capability than the following 3 most typical existing usability evaluation methods: expert usability evaluation, traditional usability testing, and SUS (*System Usability Scale*). Here, *website usability evaluation capability* contains the following 3 aspects: overall website usability evaluation, usability comparison between websites, and usability problem diagnosis for a website. If the proposed methodology's website usability evaluation capability is established, then its usability metrics can be used to *quantitatively* specify *upfront* user usability requirements for websites.

The entire validation experiment was a double-blind and multi-control groups design. First, 7 usability experts (*2~3 usability experts* is normally recommended) were selected by one of my dissertation committee members to form an expert group. Each group member was asked to independently evaluate the usabilities of 2 target websites as

usability expert. An independent third party was chosen to act as the liaison between the expert group and the committee member, but all experts had to send their expert usability review reports directly to the committee member. This expert group was regarded as *Control Group A*. Second, 2 identical groups would be formed to do user usability testing on the 2 target websites respectively. The 2 user usability testing groups together were conceptually considered as a new group that was regarded, at the same time, as *Control Group B* and *Control Group C*. This is because 3 usability evaluation methods, i.e., traditional user usability testing, SUS, and QUEST, would be used to evaluate the usabilities of the 2 target websites through the same user usability testings.

In order to eliminate possible biases, the 2 user usability testing groups would be formed with restrictions. First, all subjects should have appropriate computer systems and web skills. Second, all subjects should have no previous experience with the 2 target websites. Third, because Think-Aloud Protocol would be used to collect usability data, all subjects should have good oral English capability. Fourth, an equal number of qualified subjects would be randomly assigned to one of the 2 groups. Fifth, a group would perform the usability testing on only one of the 2 target websites, and a subject of a group would perform each of the required tasks only once. Sixth, each subject would perform the usability testing in the same format.

6.1.2 Target websites and test tasks

The 2 open source web calendar websites, WebCalendar 1.0.5 and VCalendar 1.5.3.1, were selected to be the target websites of the validation experiment. They were locally hosted respectively at:

http://spider.eng.auburn.edu/huguoqi/webcalendar/login.php, and

http://spider.eng.auburn.edu/huguoqi/vcalendar/index.php.

We have chosen them as target websites mainly for the following 3 reasons:

- Both are open source software, there are no special limitations on how they can be used.
- We only have limited resources to conduct the experiment. The sizes of both calendar websites are especially appropriate.
- Everybody has enough knowledge about web-based or electronic calendars. No special training is needed for the qualified subjects.

For a brief introduction, the following is quoted from WebCalendar's official website: "*WebCalendar is an Open Source web-based calendar/scheduling system written in PHP. WebCalendar has been under development since 2000 and continues to evolve. After years of development, testing, and user feedback from around the world, WebCalendar is a very stable and feature-rich product that compares very favorably with the best commercial calendars. WebCalendar can be configured as a single-user calendar, a multi-user calendar for groups of users, or as an event calendar viewable by visitors.*

MySQL, PostgreSQL, Oracle, DB2, Interbase, MS SQL Server, or ODBC is required."[126]

Also, for a brief introduction, the following is quoted from VCalendar's official website: "*VCalendar (Virtual Calendar) is an Open Source web calendar application with related tools, for posting and maintaining events and schedules online, in calendar format. This is an excellent and free solution for use by online Web communities and any commercial and non-commercial organizations. Unlike any other online calendars, VCalendar comes with source code in multiple programming languages: PHP, ASP and ASP.NET (C# and VB.NET); with potential for adding more technologies in the future. VCalendar features: Annual, monthly, weekly and daily calendar views; Multiple categories for classifying calendar events; Recurring and all-day events; Role-based user permissions and calendar configuration.*"[127]

The following are the 4 required test tasks:

Task 1:

Goal description: Add a calendar entry for the following event:

> A free Yoga Workshop will be held every Wednesday, from 12:00 p.m. to 2:00 p.m. in Foy Ballroom from October 3, 2007 to October 31, 2007. Participants should wear comfortable clothing for this event and bring a yoga mat or towel.
>
> Participants: Faculty, Staff, and Students with valid AU ID.

Task 2:

Goal description: Copy the Friday, August 31, 2007 "CSD Game Day Barbecue and Social" event calendar entry to Friday, November 23, 2007.

Note: Before testing, the content of the calendar entry for the Friday, August 31, 2007 "CSD Game Day Barbecue and Social" event should have already been pre-setup according to the following event information:

> *Computer Science Department's Game Day Barbecue and Social will be held on Friday, August 31, 2007 from 4:00 p.m. to 8:00 p.m. at Dunstan's west lawn. All CSD Faculty, Staff, and Students are welcome.*

Task 3:

Goal description: Edit the calendar entry for the Monday, July 21, 2008 "Fine Art Juried Student Exhibition" event:

Please change the duration of the event to: from 8:00 a.m. to 4:00 p.m., from Monday, July 7, 2008 through Friday, July 11, 2008.

Please change the location of the event to: Foy 217

Note: Before testing, the content of the calendar entry for the Monday, July 21, 2008 "Fine Art Juried Student Exhibition" event should have already been pre-setup according to the following event information:

Department of Fine Arts presents the 2008 Fine Art Juried Student Exhibition on Monday, July 21, 2008 from 9:30 a.m. to 10:00 p.m. in 101 Biggin Hall. All events are free and open to the public.

Task 4:

Goal description: Delete *only* the Wednesday, September 26, 2007 "Wireless Seminar Series" event calendar entry.

Note: Before testing, the content of the calendar entry for the Wednesday, September 26, 2007 "Wireless Seminar Series" event should have already been pre-setup according to the following event information:

Computer Science Department Fall 2007 Wireless Seminar Series will be held in Brown Hall 224 every day, from 3:00 p.m. to 5:00 p.m., from Monday, September 24, 2007 through Friday, September 28, 2007.

It should be noted that Task 2, i.e., the copy event functionality, was not directly supported by VCalendar 1.5.3.1. In other words, the subjects had to come up with their own ways to make up this task on the fly. It was purposely left out of the test task list originally. But it was later decided to be included because we wanted to see what would happen. Because of this reason, whenever possible, the usability evaluations of the 2

target websites would be considered in 2 cases: one case was when Task 2 was considered as a test task; the other was when Task 2 was excluded.

6.1.3 Expert usability evaluation

Expert usability evaluation [128] is also called expert usability inspection. It is a widely used usability evaluation method that employs several experts to independently evaluate the usability of a system and identify usability problems. This is done by walking through the system in the context of tasks and at the same time assessing the usability of the system against a set of principles. These principles are also called heuristics. Normally, 2 to 3 usability experts are needed in an expert usability evaluation project. Compared to user-based usability testing, expert-based usability evaluation is much quicker and cheaper. The result of an expert usability evaluation is usually a usability report that prioritizes a list of specific usability problems found.

After discussion, we decided to recruit 7 usability experts, but each expert was free to choose his or her own heuristics to avoid any limitation on the experts. Each expert would be requested to submit an expert usability evaluation report that should answer in detail the following 3 questions:

1. What are the usability problems you have found on each website?

2. Which website's overall usability is better in your expert opinion?

3. Why do you think one website's overall usability is better than the other's?

In order to give the experts more chances to walk through the 2 target websites, no task pre-setups were provided. The experts were instructed to pre-setup the necessary events according to the instructions.

6.1.4 Traditional user usability testing

Traditional user usability testing uses typical test subjects that are supposedly coming from the target user population of a system to perform specific tasks. While test subjects are performing the tasks, their performance data are collected. After the tasks are completed, test subjects are often asked to provide their opinions about the system through a survey or interview, so that more usability data can be collected. The entire usability testing can be video- and audio-recorded, and concurrent vocal protocols can be used to gain insights into the thinking processes of the test subjects so that the comprehension and cognition problems faced by the test subjects can be addressed.

The traditional user performance usability metrics normally are:

Task Completion Time: The amount of time that user takes to successfully complete a task.

Number of Incomplete Tasks: The number of tasks that user does not complete in the allotted time, or give up.

Error Rate: The number of errors on the way to task completion.

Error Time: The amount of time that user deals with error.

Success Ratio: The number of users that can successfully complete the task divided by the total number of users.

Help Time: The amount of time that user uses help.

Help Frequency: The number of times that user uses help.

Compared to other usability evaluation methods, user usability testing is the most expensive and time-consuming usability evaluation method. But, it is also the most essential, important, and irreplaceable usability evaluation method. It is only through user usability testing that real usability data from real users performing real tasks can be collected.

It is believed that 80% of usability problems could be detected with 4 to 5 participants [50]. We decided that 10 subjects would be recruited for each user usability testing group. In fact, our experiences in this experiment had further confirmed the belief.

6.1.5 SUS

SUS, i.e., System Usability Scale [41], is a reliable, low-cost usability scale that can be used for global usability assessments of systems. It was developed at Digital Equipment Corporation in 1986 in its pursuit for a usability measurement scale that can be used to compare usability across systems. Specifically, SUS is a simple 10-item 5-point Likert scale which gives a quantitative global view of subjective assessments of the usability of a system, with a score range of 0 to 100. Scores for individual items in a

SUS are deemed not meaningful on their own. SUS is generally administered after the subjects have used the system but before any other discussion takes place. SUS has long been used in many research projects and industrial usability evaluations, and it has proved to be a valuable, robust, and reliable usability evaluation tool [124].

It should be noted that the actual SUS questionnaire used in this experiment was adapted from its original form by replacing “system” with “website”.

6.1.6 Think-Aloud Protocol

Think-Aloud Protocol is a method that is used in usability testing to gather usability data that is otherwise kept in the experiment participants’ minds. Think-Aloud Protocol states that experiment participants are expected to say whatever they are looking at, thinking, doing, and feeling, as they go about their task so that the processes of thinking, task-performing, and problem-solving, and the nature of the difficulties encountered can be fully revealed. When Think-Aloud Protocol is used, the experiment sessions are often video- and audio-recorded.

In order to make this experiment a successful one, we had earnestly encouraged the qualified experiment participants to practice at home (for about 10 to 30 minutes on any website they like) their thinking-aloud skills before they came for the experiment. In the end, we were deeply impressed by the smoothness of the experiment in regard of the thinking-aloud by the experiment participants.

6.1.7 Pilot study

Another Ph.D. candidate also participated in the administration of this user usability testing. After the testing lab was set up, we took turns to run through the whole testing in the real testing environment at least twice to refine all the testing instruments, including the etiquette to greet each subject and the wording of the briefing, to make sure that every subject would be treated in exactly the same clear and appropriate way, all the testing equipment would work as expected, and each testing would take place in the same correct format.

6.1.8 Setup

After the pilot study was concluded, we still had not received any response from the potential volunteer subjects, which were undergraduate and graduate students from classes in Computer Science and Software Engineering Department, Auburn University. But soon after, we began to receive response emails from the students, and the subject screening and accepting process kept going on till the pre-determined number of qualified subjects were tested on both of the target websites. All the qualified subjects were assigned to one of the two user usability testing groups according to the receiving order of their response emails and the progresses of the 2 groups. Therefore, the testing group assignment was a random process. Each qualified subject had no idea about other subjects, the website, and the tasks to be tested.

After each subject was welcomed into the lab, the subject was asked to read and sign the Informed Consent form, which was part of the 06-104 EP 0706 Research Protocol that had been approved by Auburn University Institutional Review Board. The subject would then be briefed about the purpose, the procedure, and the format of the usability testing. During the briefing, the subject was told that the website was not developed by us so that the subject did not have to worry if bad testing results would embarrass us, and the subject was also told to keep in mind that it was the website usability rather than the subject that was being evaluated so that the subject should simply perform the tasks as normally and truthfully as a normal real user would.

Each subject would be expected to perform the 4 tasks on one of the two target websites using Think-Aloud Protocol. The tasks were numbered and they were supposed to be performed in the order as they were numbered. For each task, the subject would be given the goal description of the task on a piece of paper. The subject were expected to read carefully and understand fully the goal description of the task first (this was the only time the subject was encouraged to ask any questions about the goal description, because it was a usability test rather than a reading comprehension test), then independently perform the task using the Think-Aloud Protocol beginning by reading aloud the task number and the task description (to sound-mark the beginning of the task, and at the same time also to think-aloud what the subject was supposed to accomplish through the task).

In order to finish the task, the subject could seek any help (during which, the subject still needed to think-aloud) from the website if there was any on it, but the subject was not supposed to seek any help from the administrators. The subject should do the best to try to finish the task. If the subject could not figure out how to finish it, the subject could also give up. The completion or giving up of a task should also be sound-marked by saying "Finished" or "Give up" respectively. Each subject could perform a particular task only once. In order to make sure that all the subjects could have the same starting point for every task, for each task, a subject was logged into the account by the administrator after the subject said "I am ready to go" and was logged out after the subject said "Finished" or "Give up". The entire experiment process (with timing information) would be audio- and video-recorded in order to capture the experiment data.

Right after the completion of all tasks, the subject would be served with a SUS questionnaire to fill out.

After the subject had turned in the filled out SUS questionnaire, in a free style retrospective testing (or post-test interview), we would go over the recorded audio-video tape again on a "page by page and task by task" basis, to clarify things up, and at the same time orally answer a series of questions that were based on the website goal-task basic use features defined by the proposed methodology. This clarification process would also include re-examining the pages and/or re-enacting the task on the real website as necessary. The entire clarification process were also audio- and video-recorded.

It should be noted that, at the end of the post-test interview for each goal-task, the effectiveness and the user satisfaction of the goal-task, and the user satisfaction for the navigation involved in locating the goal-task would be acquired from the subject through the following oral questionnaire:

Effectiveness of the goal-task:

Assume you have taken the goal description as your own goal for performing this task. Before you perform the task, in particular what did you expect the result(s) of the task would be?

After you perform the task, what particular part(s) of your goal that have not been completed as you expected? (For each, please give an exact description: what and how?)

To what extent, this task has completed your task goal?

If 0 = not at all, 100 = fully completed as expected,

Please give your estimation:_____.

Satisfaction of the goal-task (S):

Is this task useful (S_1)?

If 0 = not at all; 10 = very useful,

Please choose: 0 1 2 3 4 5 6 7 8 9 10

Did you feel any discomfort when performing this task (S_2)?

If 0 = not uncomfortable at all; 10 = very uncomfortable,

Please choose: 0 1 2 3 4 5 6 7 8 9 10

How do you rate the quality of this task (S_3)?

If 0 = no quality at all; 10 = perfect quality,

Please choose: 0 1 2 3 4 5 6 7 8 9 10

Let's assume S_1, S_2, and S_3 have equal weights, we define:

$$S = \frac{1}{3} \cdot \frac{S_1}{10} + \frac{1}{3} \cdot \frac{10 - S_2}{10} + \frac{1}{3} \cdot \frac{S_3}{10} \qquad (6\text{-}1)$$

Satisfaction of the navigation involved in locating the goal-task (S^{gt}):

Was it easy to locate the task you were looking for (S_1^{gt})?

If 0 = very difficult; 10 = very easy,

Please choose: 0 1 2 3 4 5 6 7 8 9 10

Do you like the way provided by the website to locate the task (S_2^{gt})?

If 0 = not at all; 10 = like it very much,

Please choose: 0 1 2 3 4 5 6 7 8 9 10

Let's assume S_1^{gt}, and S_2^{gt} have equal weights, we define:

$$S^{gt} = \frac{1}{2} \cdot \frac{S_1^{gt}}{10} + \frac{1}{2} \cdot \frac{S_2^{gt}}{10} \quad (6\text{-}2)$$

6.2 Expert usability evaluation results

6.2.1 Expert usability evaluation reports

Of the 7 usability experts in the expert group, 1 did not turn in the usability report; 4 only performed the first task, so their usability reports cannot be used because of insufficient data; 2 performed all 4 tasks and their usability reports were accepted. The 2 accepted usability reports are presented respectively in Tables 6.1 and 6.2.

6.2.2 Discussion and sub-conclusion

According to the 2 usability reports, one expert preferred to use WebCalendar 1.0.5 over VCalendar 1.5.3.1, because WebCalendar 1.0.5 was more successful in task completions, even though the expert recognized that VCalendar 1.5.3.1 had more appealing interfaces and layouts. In contrast, the other expert preferred to use VCalendar 1.5.3.1 over WebCalendar 1.0.5, because VCalendar 1.5.3.1 was more aesthetically pleasing and handled dates better, even though the expert realized that VCalendar 1.5.3.1 was not as good as WebCalendar 1.0.5 in task completions and it had many other usability problems. Both experts had based their decisions on partial findings, because it can be easily seen that both experts had identified other usability problems on the target

websites, but none of those had been taken into account in the decision-makings. If we examine the 2 usability reports in detail, we can see that, although most of the usability problems identified are true usability problems, these experts had found not only different usability problems but also different numbers of usability problems. In fact, this phenomenon is common in expert usability evaluations.

Table 6.1 Expert usability evaluation report 1

Usability Problems:

WebCalendar 1.0.5:

- I could not figure out what to put in the frequency box;
- I could not figure out how to automatically copy an event;
- Instead of entering the duration to determine the end time of an event, should allow a user to enter the actual end time.

VCalendar 1.5.3.1:

- Task #1 was already in the calendar. It seems like a user's public event would only be on their calendar;
- I could not figure out the significance of the category list box;
- Task #2, I could not figure out how to automatically copy an event;
- The term 'AM' should be at the end of the time listing;
- The search for the date can benefit from a suggested format, so that you can type it in. The search for July 21, 2008 took too long;
- The recurrence checkbox seemed to disappear, so task #3 could not be changed completely;
- On task #4, there wasn't an option to delete just one day, as opposed to all occurrences.

Preference: WebCalendar 1.0.5

- VCalendar 1.5.3.1 had a more appealing interface and layout than WebCalendar 1.0.5, but I was more successful in my task completion using WebCalendar 1.0.5. In essence, VCalendar 1.5.3.1 looks better, but WebCalendar 1.0.5 gets the job done more efficiently.

Table 6.2 Expert usability evaluation report 2

Usability Problems:

WebCalendar 1.0.5:

- Alternating colors within time slots would be better, easier to see. Also, maybe lighter colors;
- I wouldn't recommend the Duration option. I prefer to see from 8 am to 4 pm and not have to think how long it is, possibly making an unnecessary mistake;
- Frequency tag is not clear as to what it exactly refers to. Explanation is provided, but does not stay on page long enough to finish reading it. Explanation also says "…the default 1…", yet it has a 0 in the box to begin with. The Frequency option is just confusing for all repeating tasks;
- Didn't see any option to share an entry with other groups or people. If there was some way to share that entry (Faculty, Staff, and Students with valid AU ID), I didn't see it;
- Why not show times before 8 am and after 5 pm on the calendar?
- When event starts at 9:30, the yellow event box shows it starting at 9:00. It should have at least 15 min splits for the yellow box;
- I hope the task wanted the location changed in the description, because if there was a location option, I didn't see it anywhere.

VCalendar 1.5.3.1:

- Not as simple to add event, could not see "+" very well, took more time to find it;
- Kind of annoying having to unselect "All Day Event" every time a new event is added. That should be an option, but not selected to begin with.

Preference: VCalendar 1.5.3.1

- To input the date when creating an even was a whole lot better and easier on WebCalendar 1.0.5, the user doesn't have to guess the date format;
- There was no way to directly copy an event in VCalendar 1.5.3.1 (at least none that I saw), and there was a way in WebCalendar 1.0.5;
- VCalendar 1.5.3.1 is better in selecting months and years when seeing the calendar of events (on the bottom left of the page). It spans many many years with drop down option, where WebCalendar 1.0.5 only spans a couple of years;
- Once a one day event has been added to VCalendar 1.5.3.1, it doesn't allow user to go back and make it a reoccurring event;
- Would probably use VCalendar 1.5.3.1 over WebCalendar 1.0.5. The major reasons are that it's more aesthetically pleasing and I like how it handled dates better.

Another phenomenon is that report 1 said that “I could not figure out how to automatically copy an event” in WebCalendar 1.0.5, but report 2 said that “There was no way to directly copy an event in VCalendar 1.5.3.1 (at least none that I saw), and there was a way in WebCalendar 1.0.5”. It can be verified that report 2 was correct on this issue.

According to the above observations, we can reach the following sub-conclusion:

1. Expert usability evaluation can identify usability problems, but it can also identify false usability problems;
2. Different experts may find different usability problems and also different numbers of usability problems;
3. Expert usability evaluation cannot find a quantitative overall usability value of a website;
4. Expert usability evaluation cannot reliably compare the overall usabilities of different websites. The final ranking of the 2 target websites by their overall usabilities through this method was, at the best, not conclusive. Although increasing the number of usability experts might improve the evaluation results, it would not overcome the limitations of this method.

In fact, it should be pointed out that the above sub-conclusion had only confirmed what had already been known about this method [38][44][128]. In other words, there was no new discovery here.

6.3 Traditional user usability testing results

6.3.1 User performance data

The complete set of user performance data is given in Appendix A.

6.3.2 Discussion and sub-conclusion

In the beginning, it seemed that all the traditional user performance metrics were well defined and made full sense. But in the end, it was not the case. Let's examine some of these metrics through the user performance data collected. We begin by taking a look at the *task completion time*. According to its definition, task completion time includes the time between the beginning and the end of a task-performing. The task completion times in Tables A.1 to A.8 reflect this definition, but the problem is that we cannot gain much insight into how the task completion times had been spent.

Some people may argue that a task completion time can be divided into task performing time, error correction time, and help time. Then the task completion time minus the sum of error correction time and help time is the task performing time. Now that we had them all, nothing was missing.

But if we check the data in the tables, we cannot find much help time and error correction time. The reason is that most web users simply learn things on the fly by trial and error, so the traditional help time is more reflected as the time spent on learning from trials and errors than the help time as originally defined. But, why cannot we find much

error correction time in the tables either? The reason is that web users normally could have many choices. If one choice does not work, they do not have to correct the old "error", instead they simply try another choice to get around it. In other words, not all errors need to be corrected when users try to get on track again. It should be pointed out that most often some trials are not really errors at all, but they should not be counted as task performing time either. The time spent this way should better be treated as wasted time rather than error correction time unless there was an error that had been corrected.

In summary, although there was much time that was spent on learning, this learning time should not be counted as error correction time, help time, or task performing time. Instead, it should be counted as wasted time (a good website does not force users to learn a lot unnecessarily). In this light, the error correction time and help time appear to not have their usability probing power at all.

Besides the above problems with the performance metrics, there are at least the following 4 problems with the traditional user usability testing method:

1. Although each performance metric appears to be comparable for a particular goal-task across the 2 target websites, they are inherently not comparable with each other. They were measuring two different goal-tasks that have the same name.
2. Different performance metrics are simply measuring different things on different scales. There is no convincing way to combine these metrics together to form a single usability score so that the overall usability can be compared.

3. Although usability problems can be revealed through user usability testing, each of the performance metrics does not have the capability to explicitly identify specific usability problems associated with it.
4. It is hard to use these performance metrics to reasonably specify usability requirements for a website upfront except using them to evaluate usability after the website has already been built.

6.4 SUS results

6.4.1 SUS data

The SUS data for usability testing on WebCalendar 1.0.5 is presented in Table 6.3. The SUS data for usability testing on VCalendar 1.5.3.1 is presented in Table 6.4.

Table 6.3 WebCalendar 1.0.5 usability testing SUS data

Subject Code	SUS Scores	Q.1	Q.2	Q.3	Q.4	Q.5	Q.6	Q.7	Q.8	Q.9	Q.10
2007100201	87.50	4	1	4	1	4	1	5	1	5	3
2007100301	72.50	1	1	4	1	3	1	2	1	4	1
2007100401	55.00	3	3	3	2	3	3	4	3	2	2
2007100803	47.50	2	3	3	1	3	2	2	4	2	3
2007100901	37.50	1	3	3	3	2	5	3	4	2	1
2007100902	95.00	5	2	5	1	4	1	5	1	5	1
2007101101	57.50	3	4	3	2	3	2	3	2	3	2
2007101201	40.00	2	4	3	2	2	3	3	4	3	4
2007101302	82.50	4	2	4	1	4	1	5	2	4	2
2007101701	85.00	4	2	5	1	4	3	4	1	5	1
Average:	66.00										

Table 6.4 VCalendar 1.5.3.1 usability testing SUS data

Subject Code	SUS Scores	Q.1	Q.2	Q.3	Q.4	Q.5	Q.6	Q.7	Q.8	Q.9	Q.10
2007100302	100.00	5	1	5	1	5	1	5	1	5	1
2007100402	55.00	2	2	3	1	4	2	2	4	2	2
2007100701	47.50	1	3	3	1	1	4	3	3	3	1
2007100801	62.50	3	3	4	1	3	4	3	3	4	1
2007100802	32.50	1	3	2	3	1	1	2	4	2	4
2007100903	72.50	2	3	3	1	4	2	5	2	4	1
2007101001	70.00	2	3	4	1	3	2	4	2	4	1
2007101202	70.00	3	2	4	1	3	2	4	3	4	2
2007101301	27.50	1	4	2	3	3	3	2	4	1	4
2007101702	80.00	2	1	4	1	3	1	5	2	4	1
Average:	61.75										

6.4.2 Discussion and sub-conclusion

As promised by SUS, we got the overall subjective usability scores for both the target websites, with WebCalendar 1.0.5 rated at 66.00 and VCalendar 1.5.3.1 at 61.75. But except these overall subjective usability scores, we got nothing else. Although we can compare the usabilities of the 2 websites by their SUS scores, we do not know exactly why the websites got those scores and how we are supposed to improve the usabilities of the websites. Certainly, we cannot use a SUS score to specify upfront user usability requirement to ensure the desired usability, either.

In fact, the subjectivity of SUS is evidenced by the fact that almost for each SUS question, on both target websites, the users' answers ran almost evenly on respective scale. It is also evidenced by the fact that subject 2007100302 on VCalendar 1.5.3.1 gave each perfect score for that website, but from our observation of the subject's task-performings, the actual situation did not warrant the perfect ratings.

So, our sub-conclusion is that, by using SUS method on a website, in the end we will get a seemingly clear SUS score, but at the same time we will also get a perplexity around that SUS score. Admittedly, it is better than nothing.

6.5 QUEST results

6.5.1 QUEST data

As mentioned before, we consider the QUEST usability evaluations of the 2 target websites in 2 cases: in Case 1, Task 2 is included in the test task list; and in Case 2, Task 2 is excluded. In both cases, we consider all the tasks have equal weights and equal use probabilities. So: for Case 1, $w_{gt_1} = w_{gt_2} = w_{gt_3} = w_{gt_4} = \frac{1}{4}$ and $p_1 = p_2 = p_3 = p_4 = \frac{1}{4}$; and for Case 2, $w_{gt_1} = w_{gt_3} = w_{gt_4} = \frac{1}{3}$ and $p_1 = p_3 = p_4 = \frac{1}{3}$.

If we assume the weight for the combined usability of all tasks is 80%, then $w_{gt} = 80\%$ and $w_{nav} = 20\%$.

The complete set of QUEST raw experiment data is given in Appendix B.

6.5.1.1 Goal-task usability

6.5.1.1.1 WebCalendar 1.0.5 goal-task usability

We can derive the composite use features for the tasks via the following steps:

1. According to formula (5-1) and the goal-task presentation basic use features data in Tables B.1 ~ B.9, we get the goal-task presentations.

2. According to formula (5-2) and the goal-task interaction basic use features data in Tables B.10 ~ B.13, we get the goal-task interactions.
3. Directly from Tables B.14, B.15, and B.16 respectively, we get the goal-task efficiencies, effectivenesses, and satisfactions.
4. According to formula (5-17) and the goal-task consistency basic use features data in Tables B.17 ~ B.22, we get the goal-task consistencies.
5. According to formula (5-5) and the data obtained in steps 1~3, we get the goal-task usabilities.

The results of the above calculations are shown in Table 6.5.

Table 6.5 Composites for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Composite Use Features	Task 1	Task 2	Task 3	Task 4
Presentation (P)	72.29%	75.37%	74.21%	77.75%
Interaction (I)	93.16%	92.50%	87.32%	87.50%
Efficiency (E)	55.34%	62.35%	55.75%	65.13%
Satisfaction (S)	72.00%	78.00%	79.33%	86.00%
Effectiveness (R)	83.00%	80.00%	77.00%	70.00%
Usability (U_{gt})	60.75%	61.65%	57.10%	55.37%
Consistency (C_{gt})	93.08%	92.24%	92.00%	91.17%

6.5.1.1.2 VCalendar 1.5.3.1 goal-task usability

Similar to 6.5.1.1.1, the composite use features for the tasks on VCalendar 1.5.3.1 can be derived. The results are shown in Table 6.6.

Table 6.6 Composites for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Composite Use Features	Task 1	Task 2	Task 3	Task 4
Presentation (P)	71. 11%	62. 05%	60. 79%	53. 67%
Interaction (I)	86. 77%	79. 82%	71. 32%	84. 27%
Efficiency (E)	45. 19%	14. 19%	0. 00%	37. 95%
Satisfaction (S)	74. 67%	22. 67%	35. 67%	86. 33%
Effectiveness (R)	78. 00%	51. 00%	0. 00%	100. 00%
Usability (U_{gt})	54. 16%	22. 79%	0. 00%	65. 55%
Consistency (C_{gt})	88. 72%	87. 19%	89. 02%	85. 42%

6.5.1.1.3 Some comments

The composite use features in Tables 6.5 and 6.6 tell us a lot about the different usability aspects of the tasks on both websites. Tables 6.7 and 6.8 illustrate the average and the biggest difference of each composite use feature across the tasks on each website respectively in Case 1 and Case 2.

Table 6.7 Comparisons of usability aspects on both websites (Case 1)

Composite Use Features	WebCalendar 1.0.5		VCalendar 1.5.3.1	
	Average	Biggest Difference	Average	Biggest Difference
Presentation (P)	74.91%	5.46%	61.91%	17.44%
Interaction (I)	90.12%	5.84%	80.55%	15.45%
Efficiency (E)	59.64%	9.79%	24.33%	45.19%
Satisfaction (S)	78.83%	14.00%	54.84%	63.66%
Effectiveness (R)	77.50%	13.00%	57.25%	100.00%
Usability (U_{gt})	58.72%	6.28%	35.63%	65.55%
Consistency (C_{gt})	92.12%	1.91%	87.59%	3.60%

Table 6.8 Comparisons of usability aspects on both websites (Case 2)

Composite Use Features	WebCalendar 1.0.5		VCalendar 1.5.3.1	
	Average	Biggest Difference	Average	Biggest Difference
Presentation (P)	74.75%	5.46%	61.86%	17.44%
Interaction (I)	89.33%	5.84%	80.79%	15.45%
Efficiency (E)	58.74%	9.79%	27.71%	45.19%
Satisfaction (S)	79.11%	14.00%	65.56%	50.66%
Effectiveness (R)	76.67%	13.00%	59.33%	100.00%
Usability (U_{gt})	57.74%	5.38%	39.90%	65.55%
Consistency (C_{gt})	92.08%	1.91%	87.72%	3.60%

From Tables 6.7 and 6.8, it can be seen that, in both Case 1 and Case 2, for all composite usability aspects, the averages for tasks on WebCalendar 1.0.5 were better than that on VCalendar 1.5.3.1.

From Table 6.6, it can be seen that: because Task 2 was not directly supported on VCalendar 1.5.3.1, the efficiency of the made-up Task 2 was only 14.19%; and due to improper design, the effectiveness of Task 3 on VCalendar 1.5.3.1 was 0%, i.e., no subject had been able to successfully finish this task.

6.5.1.2 Navigation usability

6.5.1.2.1 WebCalendar 1.0.5 navigation usability

According to formula (5-6) and the navigation presentation basic use features data in Tables B.23 ~ B.27, we get the navigation presentations in locating a goal-task. The results are shown in Table 6.9.

Table 6.9 P^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

P^{gt_1}	P^{gt_2}	P^{gt_3}	P^{gt_4}
59. 38%	59. 41%	59. 54%	59. 49%

So, according to formula (5-7), we get the presentation of entire navigation system:

For Case 1: $P_{nav} = 59.45\%$;

For Case 2: $P_{nav} = 59.47\%$.

According to formula (5-8) and the navigation interaction basic use feature data in Table B.28, we get the navigation interactions in locating a goal-task. The results are shown in Table 6.10.

Table 6.10 I^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

I^{gt_1}	I^{gt_2}	I^{gt_3}	I^{gt_4}
100. 00%	100. 00%	100. 00%	90. 00%

So, according to formula (5-9), we get the interaction of entire navigation system:

For Case 1: $I_{nav} = 97.50\%$;

For Case 2: $I_{nav} = 96.67\%$.

According to formula (5-14), we get the effectiveness of entire navigation system:

For Case 1: $R_{nav} = I_{nav} = 97.50\%$;

For Case 2: $R_{nav} = I_{nav} = 96.67\%$.

According to the navigation architecture of WebCalendar 1.0.5, the reaching distances for Task 1, 2, 3, and 4 are 1, 3, 3, and 3 respectively; the breadth of its navigation $W_{\max} = 31$. Then, according to formula (5-10), we get the average probability reaching distance of the website:

For Case 1: $D_{ap} = 2.5000$;

For Case 2: $D_{ap} = 2.3333$.

So, according to formulas (5-12), (5-13), and (5-11), we get the efficiency of entire navigation system:

For Case 1: $E_{nav} = 56.25\%$;

For Case 2: $E_{nav} = 60.00\%$.

According to formula (5-15) and the navigation satisfactions data in Table B.29, we get the navigation satisfaction of entire navigation system:

For Case 1: $S_{nav} = 85.75\%$;

For Case 2: $S_{nav} = 85.83\%$.

According to formula (5-16), we get the usability of entire navigation system:

For Case 1: $U_{nav} = 72.87\%$;

For Case 2: $U_{nav} = 72.98\%$.

According to formula (5-18) and the navigation consistency basic use features data in Tables B.30 ~ B.34, we get the navigation consistencies in locating a goal-task. The results are shown in Table 6.11.

Table 6.11 C_{nav}^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

$C_{nav}^{gt_1}$	$C_{nav}^{gt_2}$	$C_{nav}^{gt_3}$	$C_{nav}^{gt_4}$
95.11%	92.61%	86.13%	91.09%

So, according to formula (5-19), we get the consistency of entire navigation system:

For Case 1: C_{nav} = 91.23%;

For Case 2: C_{nav} = 90.77%.

Table 6.12 shows all composite use features of WebCalendar 1.0.5 navigation system.

Table 6.12 Composites for WebCalendar 1.0.5 navigation system

Composite Use Features	Case 1	Case 2
Presentation (P_{nav})	59.45%	59.47%
Interaction (I_{nav})	97.50%	96.67%
Efficiency (E_{nav})	56.25%	60.00%
Satisfaction (S_{nav})	85.75%	85.83%
Effectiveness (R_{nav})	97.50%	96.67%
Usability (U_{nav})	72.87%	72.98%
Consistency (C_{nav})	91.23%	90.77%

From Table 6.12, it can be seen that for all composite usability aspects, there was no substantial difference between Case 1 and Case 2.

6.5.1.2.2 VCalendar 1.5.3.1 navigation usability

Similar to 6.5.1.2.1, the composite use features of VCalendar 1.5.3.1 navigation system can be derived. The results are shown in Table 6.13.

Table 6.13 Composites for VCalendar 1.5.3.1 navigation system

Composite Use Features	Case 1	Case 2
Presentation (P_{nav})	48.57%	48.25%
Interaction (I_{nav})	97.50%	100.00%
Efficiency (E_{nav})	39.38%	52.50%
Satisfaction (S_{nav})	68.88%	69.83%
Effectiveness (R_{nav})	97.50%	100.00%
Usability (U_{nav})	61.99%	67.64%
Consistency (C_{nav})	89.41%	89.10%

From Table 6.13, it can be seen that except the navigation efficiency in Case 2 was better than that in Case 1, for all other composite usability aspects, there was no substantial difference between Case 1 and Case 2. Majorly because of better efficiency, the navigation usability in Case 2 was better than that in Case 1.

6.5.1.3 Website consistency

According to formula (3-9), we get the consistency of WebCalendar 1.0.5:

For Case 1: $C = 91.95\%$;

For Case 2: $C = 91.82\%$.

Similarly, we get the consistency of VCalendar 1.5.3.1:

For Case 1: $C = 87.95\%$;

For Case 2: $C = 88.00\%$.

6.5.1.4 Website usability

According to formula (3-8), we got the usability of WebCalendar 1.0.5:

For Case 1: $U = 56.59\%$;

For Case 2: $U = 55.82\%$.

Similarly, we got the usability of VCalendar 1.5.3.1:

For Case 1: $U = 35.97\%$;

For Case 2: $U = 40.00\%$.

From the above results, it can be seen that on WebCalendar 1.0.5, there was no substantial difference in usabilities between Case 1 and Case 2, but on VCalendar 1.5.3.1, the usability in Case 2 was better than that in Case 1. Nonetheless, in both cases, even

though the overall usabilities of both websites were not good, the usability of WebCalendar 1.0.5 was better than that of VCalendar 1.5.3.1.

6.5.2 Discussion and sub-conclusion

As shown above, QUEST has the following usability evaluation capabilities:

1. It is fully quantitative and all the results are comparable. For example, the usability of Task 1 on WebCalendar 1.0.5 can compare not only with the usability of Task 1 on VCalendar 1.5.3.1, but also with the usability of Task 2, or 3, or 4 on either WebCalendar 1.0.5 or VCalendar 1.5.3.1. The overall website usabilities are also comparable. For example, in both cases, we know that the usabilities of both target websites were not good, but the usability of WebCalendar 1.0.5 was better than the usability of VCalendar 1.5.3.1.
2. Its metrics are diagnostic and meaningful. For example, respectively from P_1, P_2, P_3 of Task 1 on WebCalendar 1.0.5 (*see Tables B.1 to B.3 in Appendix B for details*), we know that 25.92% of its interface items are confusing or misleading, 0% of its pages have inappropriate theme ratio, and 5.67% of its pages did not have sufficient necessary methods.
3. Its metrics can be used to specify upfront quantitative user usability requirements for websites.

6.6 Conclusions and discussion

Based on the discussions and sub-conclusions about the 4 usability evaluation methods, the website usability evaluation capabilities of the 4 methods are summarized in Table 6.14.

Table 6.14 Capability comparisons between the 4 methods

Capability / Methods	Overall Website Usability Evaluation	Usability Comparison Between Websites	Usability Problem Diagnosis
Expert Usability Evaluation	Inconclusive	Inconclusive	Specific
Traditional Usability Testing	Inconclusive	Inconclusive	Specific
SUS	WebCalendar 1.0.5: 66.00 VCalendar 1.5.3.1: 61.75	WebCalendar 1.0.5: 66.00 VCalendar 1.5.3.1: 61.75	Vague, subjective
QUEST	Case 1 { WebCalendar 1.0.5 : 56.59; VCalendar 1.5.3.1 : 35.97 } Case 2 { WebCalendar 1.0.5 : 55.82; VCalendar 1.5.3.1 : 40.00 }	Case 1 { WebCalendar 1.0.5 : 56.59; VCalendar 1.5.3.1 : 35.97 } Case 2 { WebCalendar 1.0.5 : 55.82; VCalendar 1.5.3.1 : 40.00 }	Specific, direct, quantitative

From Table 6.14, it can be seen that:

- For the *overall website usability evaluation*, both expert usability evaluation and traditional usability testing were inconclusive, but both SUS and QUEST produced their quantitative evaluations.
- For the *usability comparison between websites*, both expert usability evaluation and traditional usability testing were inconclusive, but both SUS and QUEST produced

similar rankings for the 2 target websites. It should be mentioned that the difference in scores is noticeably greater for QUEST. We think this is because QUEST is more accurate in differentiating usabilities than SUS.

- For the *usability problem diagnosis*, both expert usability evaluation and traditional usability testing identified specific usability problems, SUS could not identify specific usability problems, but each of the QUEST metrics identified direct and specific usability problems in a quantitative manner.

Therefore, it can be clearly concluded that QUEST has stronger website usability evaluation capability than all other 3 most typical existing usability evaluation methods. According to the principle of this validation experiment, the proposed methodology has been validated.

In fact, it is worth noting that, besides the above conclusion, we are also very impressed (even surprised) by the following four findings through this experiment:

1. Think-Aloud Protocol worked extremely well.
2. With easy test tasks, many subjects tried hard but still failed to finish their tasks.
3. The SUS scores were distributed almost evenly on almost all the SUS scales. One perfect score could not be vindicated by the actual task-performings.
4. According to the official websites of both target websites, their designers must have sincerely believed that the target websites have been designed with good usability. But through this experiment, it is clear that both target websites are not as usable as

their designers might have wished. This phenomenon indicates how severe the mental model schism might be in the real world and also how important it is to reveal the distance of mental model schism via QUEST user usability testing.

CHAPTER 7

CONCLUSION AND FUTURE WORK

The work of this dissertation was based on the achievements of almost 30 years of research in usability engineering. As pointed out by Hornbæk in [84], "despite more than 20 years of research into usability, current practice in measuring usability suggests that choosing usability measures is difficult". Our motivation was to provide a full lifecycle and fully quantitative methodology to change this situation. In this dissertation, we focused on website quantitative usability engineering.

Through this work, we have achieved the following main accomplishments:

- A structured and fully quantitative usability definition framework has been established. It is more complete, clearer, and more usable than the vague and not so usable ISO 9241-11 usability definition.
- A new concept — use feature — is provided. In this methodology, use feature is the core of quantitative usability measurement. Expressed as quality levels in percentages, use features are used to measure the distances of mental-model schism in respective usability aspects. Based on this concept, a whole set of new quantitative usability metrics for websites is proposed. Among all use features, it is especially

worth noting that the new definition of the use feature *efficiency* in this methodology is unique.

- Usabilities are comparable across products without conversion.
- End users are now able to easily specify upfront quantitative usability requirements for websites though the metrics of this methodology. This guarantees that the desired user usability requirements will eventually be satisfied just like other kinds of user requirements have always been.

As stated before, this endeavor to provide a structured fully quantitative and full lifecycle website usability engineering framework is still at its infancy stage. The details of this methodology, for example, the weighting schemes, basic use features for each major usability aspect, and the formations of QUEST, need to be polished in practice.

Besides websites, how to apply this methodology to other kinds of human-tool interaction systems needs to be explored. We wish this dissertation would mark the coming of age of fully quantitative usability engineering.

Just as CASE (*C*omputer *A*ided *S*oftware *E*ngineering) tools to software engineering, it would be helpful to have *C*omputer *A*ided *U*sability *S*pecification and *E*valuation (*CAUSE*) tools that are oriented toward this methodology.

Between each percentage change of a particular usability aspect and its corresponding percentage impact on the budget of a project, there might be a relationship similar to the one illustrated in Figure 7.1. But, the exact relationship between them while

taking the scale of the project into account needs to be investigated. Last, but not the least, the overall economical impact of this methodology to the usability engineering practices also needs to be studied.

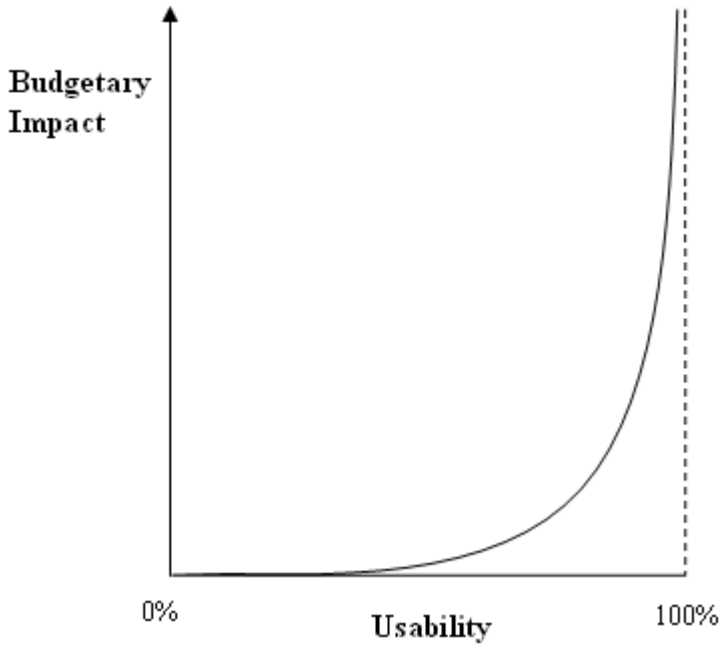


Figure 7.1 Possible relationship between usability and its budgetary impact

So, how should we conquer the usability issues? In terms of our terms, the answer is “QUEST for the CAUSE!”.

BIBLIOGRAPHY

[1] Telegeography/PriMetrica in D.C. Market Research Report: "Global Internet Geography 2005", Primetrica, Inc., 2005.

[2] The “Internet World Stats —— Usage and Population Statistics” website. URL: http://www.internetworldstats.com/stats.htm

[3] Berners-Lee, Tim. "Weaving the Web: the original design and ultimate destiny of the World Wide Web by its inventor", 1st edition, HarperCollins, 1999.

[4] The World Wide Web Consortium (W3C) website: http://www.w3.org/

[5] CNN.com & the Lemelson-MIT Program. "Top 25: Innovations —— The Internet, ranked No. 1, changed the world", URL: http://www.cnn.com/2005/TECH/01/03/cnn25.top25.innovations/index.html.

[6] Edited by Albarran, Alan B.; Goff, David H. "Understanding the Web: social, political, and economic dimensions of the Internet", 1st edition, Iowa State University Press, 2000; ISBN: 081382527X.

[7] Netcraft.com: "October 2005 Web Server Survey", URL: http://news.netcraft.com/archives/web_server_survey.html

[8] Mital, Anil; Kumar, S.; Kilbom, A. "Ergonomics Guidelines and Problem Solving", 1st edition, Elsevier Science & Technology Books, 2000. ISBN: 0080436439.

[9] Oborne, D. "Ergonomics and Human Factors", 1st edition, New York Univ Pr, 1995. ISBN: 0814761887.

[10] Dix, A. J.; Finlay, J.; Beale, R.; Abowd, G. "Human Computer Interaction", 3rd edition, Prentice Hall, 2003. ISBN: 0130461091.

[11] Golden, D. "A plea for friendly software", ACM SIGSOFT Software Engineering Notes, Volume 5 Issue 4, pp. 4~5, October 1980.

[12] Dwyer, B. "A user-friendly algorithm", Communications of the ACM, Volume 24 Issue 9, pp. 556~561, September 1981.

[13] Root, R. W.; Draper, S. W. "Questionnaires as a Software Evaluation Tool", Proceedings of CHI ’83, Boston, pp. 83~87.

[14] Borland, R. E. "'Those silly bastards': A report on some users' views of documentation", Proceedings of the 2nd annual international conference on Systems documentation, pp. 11~15, April 1983.

[15] Sherman, B. "The New Revolution: The Impact of Computers on Society", John Wiley, 1985.

[16] Wade, J. "Practical guidelines for a user-friendly interface", ACM SIGAPL APL Quote Quad, Proceedings of the international conference on APL APL '84, Volume 14 Issue 4, pp. 365~371, June 1984.

[17] Salton, G. "Some characteristics of future information systems", ACM SIGIR Forum, Volume 18 Issue 2-4, pp. 28~39, September 1985.

[18] Stevens, G. C. "User-friendly computer systems? A critical examination of the concept", Behavior & Information Technology, Volume 2 Issue 1, pp. 3~16, 1983.

[19] Norman, D.; Draper, S. "User Centered System Design: New Perspectives on Human-Computer Interaction", Lawrence Erlbaum Associates, 1986.

[20] Nielsen, J. "Usability Engineering", Boston: Academic Press, c1993.

[21] Miller, R. B. "Human ease of use criteria and their tradeoffs", IBM Technical Report TR 00.2185. Poughkeepsie, NY: IBM Corporation, 1971.

[22] Shackel, B. "The concept of usability", Proceedings of IBM Software and Information Usability Symposium, September 15-18, 1981, Poughkeepsie, New York, USA, 1-30. Poughkeepsie, NY: IBM Corporation.

[23] Shackel, B. "The concept of usability", In Bennett, J. L., Case, D., Sandelin, J., & Smith, M. (Eds.), Visual display terminals: Usability issues and health concerns, pp. 45~87, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[24] Shackel, B. "Ergonomics in design for usability", In Harrison, M. D. & Monk, A. F., People and computers: Designing for usability, pp. 44~64. Proceedings of HCI 86. Cambridge, UK: Cambridge University Press, 1986.

[25] Shackel, B. "Human factors and usability", in Preece and Keller (Eds.), Human Computer Interaction, pp. 27~41. Prentice Hall, Hemel Hempstead, 1990.

[26] Bennett, J. L. "The commercial impact of usability in interactive systems", In Shackel, B. (Ed.), Man/computer communication: Infotech state of the art report, Volume 2, pp. 1~17. Maidenhead, UK: Infotech International, 1979.

[27] Bennett, J. L. "Managing to meet usability requirements: Establishing and meeting software development goals", In Bennett, J. L., Case, D., Sandelin, J., & Smith, M. (Eds.), Visual display terminals: Usability issues and health concerns, pp. 161~184, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[28] Shneiderman, B. "Designing the user interface: Strategies for effective human-computer interaction", 1st edition, Addison-Wesley, 1987.

[29] Shneiderman, B. "Designing the user interface: Strategies for effective human-computer interaction", 2nd edition, Addison-Wesley, 1992.

[30] Shneiderman, B. "Designing the user interface: Strategies for effective human-computer interaction", 3rd edition, Addison-Wesley, 1998.

[31] Bevan, N. "Usability is quality of use", In Anzai, Y., Ogawa, K., & Mori, H. (Eds.), Symbiosis of human and artifact: Future computing and design for human-computer interaction, pp. 349~354. Amsterdam, The Netherlands: Elsevier.

[32] Löwgren, J. "Perspectives on usability", 1995 IDA Technical Report LiTH-IDA-R-95-23, Linköping, Sweden: Department of Computer and Information Science, Linköping University.

[33] Dix, A. J.; Finlay, J.; Beale, R.; Abowd, G. "Human Computer Interaction", 2nd edition, Prentice Hall, 1998.

[34] Quesenbery, W. "What does usability mean: Looking beyond 'ease of use.'", Proceedings of 48th Annual Conference Society for Technical Communication, 2001.

[35] Quesenbery, W. "Dimensions of usability", In Albers, M., & Mazur, B. (Eds.), Content and complexity: Information design in technical communication. Mahwah, NJ: Lawrence Erbaum Associates, 2003.

[36] ISO. "Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability", (ISO 9241-11:1998).

[37] Lund, A. (1998). "The need for a standardized set of usability metrics", Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting, pp. 688~691, Santa Monica, CA, 1998.

[38] Faulkner, X. "Usability engineering", Macmillan, 2000.

[39] Barnum, C. "Usability testing and research", Longman, 2002.

[40] Opaluch, R. "Usability Metrics", In Ratner, J. (Eds.), Human factors and Web development, 2nd edition, pp. 101~122, Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

[41] Brooke, J. "SUS: A Quick and Dirty Usability Scale", In Jordan, P. W.; Thomas, B.; Weerdmeester, B. A. & McClelland, I. L. (eds), Usability Evaluation in Industry, pp. 189~194, London, UK, Taylor & Francis, 1996.

[42] van Welie, M., van der Veer, G. C., & Eliëns, A. "Breaking down usability", Proceedings of INTERACT 99, pp. 613~620, Amsterdam, The Netherlands: IOS Press.

[43] Nielsen, J., & Mack, R. L. "Usability inspection methods", John Wiley & Sons, 1994.

[44] Cockton, G.; Lavery, D.; Woolrych, A. "Inspection-based evaluations", In Jacko, J. A., & Sears, A. (Eds.), The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications, pp. 1020~1040. Mahwah, NJ: Lawrence Erbaum Associates, 2002.

[45] Fu, L.; Salvendy, G.; Turley, L. "Who finds what in usability evaluation?", Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting, pp. 1341~1345, Santa Monica, CA: HFES, 1998.

[46] Human-Computer Interaction, Volume 13 Issue 3, 1998 - Special Issue on Experimental Comparisons of Usability Evaluation Methods.

[47] Hartson, H. R.; Andre, T. S.; Williges, R. C. "Criteria for evaluating usability evaluation methods", International Journal of Human-Computer Interaction, Volume 13 Issue 4, pp. 373~410, 2001.

[48] Scriven, M. "Evaluation thesaurus", 4th edition, Sage Publications, 1991.

[49] Nielsen, J. "Usability engineering at a discount", In Salvendy, G. & Smith, M. J. (Eds.), Designing and using human-computer interfaces and knowledge based systems, pp. 394~401, Amsterdam, The Netherlands: Elsevier, 1989.

[50] Virzi, R. A. "Refining the test phase of usability evaluation: How many subjects is enough?", Human Factors, Volume 34 Issue 4, pp. 457~468, 1992.

[51] Lewis, J. R. "Sample sizes for usability studies: Additional considerations", Human Factors, Volume 36 Issue 2, pp. 368~378, 1994.

[52] Shneiderman, B., "Pushing human-computer interaction research to empower every citizen: Universal Usability", Communications of the ACM, Volume 43 issue 5, pp. 84~91, May 2000.

[53] Gould, J. G.; Boies, S. J.; Lewis, C. "Making usable, useful, productivity-enhancing computer applications", Communications of the ACM, Volume 34 issue 1, pp. 74~85, 1991.

[54] Gould, J. D.; Boies, S. J.; Ukelson, J. "How to design usable systems", In Helander, M. G.; Landauer, T. K. & Prabhu, P. V. (Eds.), Handbook of human-computer interaction, 2nd edition, pp. 231~254. Amsterdam, The Netherlands: North-Holland, 1997.

[55] Mao,J. Y.; Vredenburg, K.; Smith, P. W.; Carey, T. "User-centered design methods in practice: a survey of the state of the art", Proceedings of the 2001 conference of the Center for Advanced Studies on Collaborative research.

[56] ISO 13407: Human-Centered Design Processes for Interactive Systems.

[57] ISO TR 18529: Human-Centered Lifecycle Process Descriptions.

[58] Gulliksen, J.; Göransson, B.; Boivie, I.; Blomkvist, S.; Persson, J.; Cajander, Å. "Putting usability back on track: Key principles for applying user-centered systems design", Behaviour & Information Technology, Volume 22 issue 6, pp.397~409, 2003.

[59] Good, M.; Spine, T. M.; Whiteside, J.; George, P. "User-derived impact analysis as a tool for usability engineering", Proceedings of CHI 86, pp. 241~246, 1986.

[60] Schaffer, E. "Institutionalization of usability : a step-by-step guide", Boston: Addison-Wesley, 2004.

[61] Nielsen, J. "Designing web usability", Indianapolis, Ind.: New Riders, 2000.

[62] Ratner, J. (Eds.), "Human factors and Web development", 2nd edition, Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

[63] Pearrow, M. "Web site usability handbook", Rockland, Mass.: Charles River Media, Inc., 2000.

[64] Reiss, E. "Practical information architecture: a hands-on approach to structuring successful websites", Harlow: Addison Wesley, 2000.

[65] Dustin, E.; Rashka, J.; McDiarmid, D. "Quality Web systems: performance, security, and usability", Boston: Addison-Wesley, 2002.

[66] Constantine, L. L.; Lockwood, L. A. D. "Usage-centered engineering for web applications", IEEE Computer, Volume 19 issue 2, pp. 42~50, 2002.

[67] Keevil, B. "Measuring the usability index of your Web site", Proceedings of the 16th annual international conference on Computer documentation, pp. 271~277, 1998.

[68] Spool, J. M.; Scanlon, T.; Snyder, C.; Shroeder, W.; DeAnelo, T. "Web site usability: a designer's guide", San Francisco: Morgan Kaufmann Publishers, 1999.

[69] Krug, S. "Don't make me think!: a common sense approach to Web usability", Indianapolis, Ind.: Que, 2000.

[70] Nielsen, J.; Tahir, M. "Homepage usability: 50 websites deconstructed", Indianapolis, IN: New Riders, 2002.

[71] Ivory, M. Y.; Hearst, M. A. "The state of the art in automating usability evaluation of user interfaces", ACM Computing Surveys (CSUR), Volume 33 Issue 4, pp. 470~516, December 2001.

[72] Ivory, M. Y.; Sinha, R. R.; Hearst, M. A. "Empirically validated web page design metrics", Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 53~60, March 2001.

[73] Babiker, E. M. "A metric for hypertext usability", Proceedings of the 9th annual international conference on Systems documentation, pp.95~104, October 1991.

[74] Fukuda, K. "Proposing new metrics to evaluate web usability for the blind", CHI '05 extended abstracts on Human factors in computing systems, pp 1387~1390, April 2005.

[75] Ivory, M. Y.; Hearst, M. A. "Web Site Analysis: Statistical profiles of highly-rated web sites", Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, pp. 367~374, April 2002.

[76] Juvina, I.; Oostendorp, H. "Predicting user preferences: from semantic to pragmatic metrics of Web navigation behavior", Proceedings of the conference on Dutch directions in HCI, pp. 1~4, June 2004.

[77] Chi, E. H.; Pirolli, P.; Pitkow, J. "The scent of a site: a system for analyzing and predicting information scent, usage, and usability of a Web site", Proceedings of the SIGCHI conference on Human factors in computing systems, CHI Letters volume 2 issue 1, pp. 161~168, April 2000.

[78] Chi, E. H.; Rosien, A.; Supattanasiri, G. "Web usability: The bloodhound project: automating discovery of web usability issues using the InfoScentπ simulator", Proceedings of the SIGCHI conference on Human factors in computing systems, Volume 5 Issue 1, pp. 505~512, April 2003.

[79] Hartson, H. R.; Castillo, J. C.; Kelso, J.; Neale, W. "Remote evaluation: the network as an extension of the usability laboratory", Proceedings of the SIGCHI conference on Human factors in computing systems: common ground, pp. 228~235, April 1996.

[80] Hartson, H. R.; Castillo, J. C. "Invited papers and panel: Remote evaluation for post-deployment usability improvement", Proceedings of the working conference on Advanced visual interfaces, pp. 22~29, May 1998.

[81] Winckler, M.; Freitas, C. "Usability remote evaluation for WWW", CHI '00 extended abstracts on Human factors in computing systems, pp. 131~132.

[82] Hong, J. I.; Heer, J.; Waterson, S.; Landay, J. "WebQuilt: A proxy-based approach to remote web usability testing", ACM Transactions on Information Systems (TOIS), Volume 19 Issue 3, pp. 263~285, July 2001.

[83] Thompson, K.; Rozanski, E.; Haake, A. "Learning: Here, there, anywhere: remote usability testing that works", Proceedings of the 5th conference on Information technology education, pp. 132~137, October 2004.

[84] Hornbæk, K. "Current practice in measuring usability: Challenges to usability studies and research", International Journal of Human-Computer Studies, Volume 64, pp. 79~102, 2005.

[85] Kirakowski, J. "The Software Usability Measurement Inventory: Background and usage", In Jordan, P.; Thomas, B.; Weerdmeester, B.; McClelland, I. (Eds.), Usability evaluation in industry, pp 169~178, London, UK: Taylor & Francis, 1996.

[86] Kirakowski, J.; Corbett, M. "SUMI: The software usability measurement inventory", British Journal of Educational Technology, Volume 24 Issue 3, pp. 210~212, 1993.

[87] Lewis, J. "IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use", International Journal of Human-Computer Interaction, Volume 7 Issue 1, pp 57~78, 1995.

[88] Lewis, J. "Tradeoffs in the design of the IBM computer usability satisfaction questionnaires", In Bullinger, H.; Ziegler, J. (Eds.), Human-computer interaction: Ergonomics and user interfaces, Proceedings of HCI International 99, Volume 1, pp 1023~1027, 1999, Mahwah, NJ: Lawrence Erbaum Associates.

[89] Kirakowski, J.; Corbett, M. "Measuring user satisfaction", Proceedings of HCI 88, pp 329~338, 1988.

[90] Kirakowski, J.; Dillon, A. "The computer user satisfaction inventory (CUSI): Manual and scoring key", Cork, Ireland: Human Factors Research Group, University College Cork, 1988.

[91] MUMMS's website: http://www.ucc.ie/hfrg/questionnaires/mumms/

[92] Lewis, J. "Psychometric evaluation of the post-study system usability questionnaire: The PSSUQ", Proceedings of the Human Factors Society 36th Annual Meeting, pp. 1259~1263, Santa Monica, CA: HFES., 1992.

[93] Lewis, J. "Psychometric evaluation of the PSSUQ using data from five years of usability studies", International Journal of Human-Computer Interaction, Volume 14 Issue 3, pp. 463~488, 2002.

[94] Chin, J.; Diehl, V.; Norman, K. "Development of an instrument measuring user satisfaction of the human-computer interface", Proceedings of CHI 88, pp. 213~218. New York, NY: ACM, 1988.

[95] Harper, B.; Norman, K. "Improving user satisfaction: The questionnaire for user interaction satisfaction version 5.5", Proceedings of the 1st Annual Mid-Atlantic Human Factors Conference, pp. 224~228, 1993.

[96] WAMMI's website: http://www.wammi.com/

[97] McGee, M. "Master usability scaling: magnitude estimation and master scaling applied to usability measurement", Proceedings of CHI 2004, pp 335~342.

[98] McGee, M. "Usability magnitude estimation", Proceedings of HFES, 47th Annual Meeting, pp 691~695, 2003.

[99] Berglund, M. "Quality assurance in environmental psychophysics", In Bolanowski, S.; Gescheider, G. (Eds.), Ratio scaling of psychological magnitude: In honor of the memory of S.S. Stevens, Lawrence Erlbaum Associates, 1991.

[100] Sauro, J.; Kindlund, E. "A Method to Standardize Usability Metrics into a Single Score", Proceedings of the Conference in Human Factors in Computing Systems (CHI 2005), pp. 401~409, Portland, USA, 2005.

[101] Sauro, J.; Kindlund, E. "Using a Single Usability Metric (SUM) to Compare the Usability of Competing Products", Proceeding of the Human Computer Interaction International Conference (HCII 2005), Las Vegas, USA, 2005.

[102] Gupta, P.; Gilbert, J. "Speech Usability Metric: Evaluating Spoken Language Systems", 11th International Conference on Human-Computer Interaction, Las Vegas, USA, 2005.

[103] Hartson, R.; Hix, D. "CS5714: Usability Engineering", available at: http://courses.cs.vt.edu/~cs5714/fall2004/Class%20notes/class%20notes.pdf, Computer Science Department of Virginia Tech, 2004.

[104] Gould, J.; Lewis, C. "Design for usability: key principles and what designers think", Communications of the ACM Volume 28 Issue 3, pp. 300~311, 1985.

[105] Whiteside, J.; Bennett, J.; Holtzblatt, K. "Usability engineering: our experience and evolution", In Helander, M. (Ed.), Handbook of Human–Computer Interaction, pp. 791~817, Elsevier, 1988.

[106] Forlizzi, J.; Ford, S. "The building blocks of experience: an early framework for interaction designers", Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques, pp. 419~423, August 2000.

[107] Marcus, A. "Fast forward: The cult of cute: the challenge of user experience design", interactions, Volume 9 Issue 6, pp. 29~34, November 2002.

[108] McClelland, I. "Development consortium: 'User experience' design a new form of design practice takes shape", CHI '05 extended abstracts on Human factors in computing systems, pp. 1096~1097, April 2005.

[109] Streitz, N.; Magerkurth, C.; Prante, T.; Röcker, C. "Ambient intelligence: the next generation of user centeredness: From information design to experience design: smart artefacts and the disappearing computer", interactions, Volume 12 Issue 4, pp. 21~25, July 2005.

[110] Norman, D. A. "The Psychology of Everyday Things", Basic Books, 1988. ISBN:0465067093.

[111] Norman, D.A. "Cognitive engineering", Chapter 3 in Norman and Draper ed., User centered system design: New perspectives on human computer interaction, pp. 31~61, Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[112] Snowberry, K.; Parkinson, S.; Sisson, N. "Computer Display Menus", Ergonomics, 26, pp. 699~712, 1983.

[113] Schultz, E.; Curran, P. "MENU STRUCTURE AND ORDERING OF MENU SELECTION: INDEPENDENT OR INTERACTIVE EFFECTS?", ACM SIGCHI Bulletin, Volume 18 Issue 2, pp. 69~71, October 1986.

[114] Tolle, J.; Prasse, M.; Gott, R. "EFFECTS OF VARIATION IN MENU LENGTH AND NUMBER-OF WINDOWS ON USER SEARCH TIME", ACM SIGCHI Bulletin, Volume 18 Issue 3, pp. 73~74, January 1987.

[115] Gonzales, J. "A theory of organization", Proceedings of the 12th annual international conference on Systems documentation: technical communications at the great divide, pp. 145~155, October 1994.

[116] Larson, K.; Czerwinski, M. "Web page design: implications of memory, structure and scent for information retrieval", Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 25~32, January 1998.

[117] Miller, D. P. "The depth/breadth tradeoff in hierarchical computer menus", Proceedings of the Human Factors Society, pp. 296~300, 1981.

[118] Kiger, J. I. "The depth/breadth tradeoff in the design of menu-driven interfaces", International Journal of Man-Machine Studies, 20, pp. 201~213, 1984.

[119] Jacko, J. A.; Slavendy, G. "Hierarchical Menu Design: breadth, depth and task complexity", Perceptual and Motor skills, 82, pp. 1187~1201, 1996.

[120] Zaphiris, P.; Mtei, L. "Depth vs Breadth in the Arrangement Web Links", 1997, AvailabIe at http://otal.umd.edu/SHORE/bs04/.

[121] Miller, G. A. "The magical number seven plus or minus two: Some limits on our capacity for processing information", Psychological Review, 63, pp. 81~97, 1956.

[122] Holzschlag, Molly E. "Using HTML 4", 6th edition, Macmillan Computer Publishing, 1999.

[123] Berners-Lee, T.; Masinter, L.; McCahill, M. "Uniform Resource Locators (URL)", Internet RFC 1738, December 1994. Available at URL: ftp://ftp.nordu.net/rfc/rfc1738.txt.

[124] Tullis, Thomas S.; Stetson, Jacqueline N. "A Comparison of Questionnaires for Assessing Website Usability", UPA 2004.

[125] Jordan, Patrick W. "An Introduction to Usability", London: Taylor & Francis Books Ltd, 1998.

[126] WebCalendar's official website URL: http://www.k5n.us/webcalendar.php

[127] VCalendar’s official website URL:
http://www.ultraapps.com/app_overview.php?app_id=19

[128] Salvendy, G. "Handbook of human factors and ergonomics", University of Michigan: J. Wiley, 1997.

APPENDIX A

TRADITIONAL USABILITY TESTING DATA

Table A.1 WebCalendar 1.0.5 task 1 user performance data

Subject Code	Task Completion Time (Sec.)	Number of Incomplete Tasks	Error Rate	Error Time (Sec.)	Success Ratio	Help Time (Sec.)	Help Frequency
2007100201	200	0	0	0	1	0	0
2007100301	158	0	1	0	1	0	0
2007100401	233	1	1	0	0	0	0
2007100803	218	0	1	7	1	0	0
2007100901	270	0	1	0	1	0	0
2007100902	428	0	0	0	1	0	0
2007101101	200	0	0	0	1	0	0
2007101201	374	0	1	0	1	0	0
2007101302	583	0	0	0	1	0	0
2007101701	154	0	0	0	1	0	0
Average:	281.80		0.50	0.70	0.90	0.00	0.00
Total:		1					

Table A.2 WebCalendar 1.0.5 task 2 user performance data

Subject Code	Task Completion Time (Sec.)	Number of Incomplete Tasks	Error Rate	Error Time (Sec.)	Success Ratio	Help Time (Sec.)	Help Frequency
2007100201	72	0	0	0	1	0	0
2007100301	170	0	0	0	1	0	0
2007100401	36	1	1	0	0	0	0
2007100803	68	0	0	0	1	0	0
2007100901	119	0	0	0	1	0	0
2007100902	121	0	0	0	1	0	0
2007101101	113	0	0	0	1	0	0
2007101201	134	0	0	0	1	0	0
2007101302	158	1	0	0	0	0	0
2007101701	68	0	0	0	1	0	0
Average:	105.90		0.10	0.00	0.80	0.00	0.00
Total:		2					

Table A.3 WebCalendar 1.0.5 task 3 user performance data

Subject Code	Task Completion Time (Sec.)	Number of Incomplete Tasks	Error Rate	Error Time (Sec.)	Success Ratio	Help Time (Sec.)	Help Frequency
2007100201	163	0	0	0	1	0	0
2007100301	355	0	0	0	1	0	0
2007100401	90	1	0	0	0	0	0
2007100803	157	1	0	0	0	0	0
2007100901	318	0	1	0	1	0	0
2007100902	204	0	0	0	1	0	0
2007101101	191	0	1	38	1	0	0
2007101201	285	1	0	0	0	0	0
2007101302	92	0	0	0	1	0	0
2007101701	173	0	1	63	1	0	0
Average:	202.80		0.30	10.10	0.70	0.00	0.00
Total:		3					

Table A.4 WebCalendar 1.0.5 task 4 user performance data

Subject Code	Task Completion Time (Sec.)	Number of Incomplete Tasks	Error Rate	Error Time (Sec.)	Success Ratio	Help Time (Sec.)	Help Frequency
2007100201	28	0	0	0	1	0	0
2007100301	73	1	1	0	0	0	0
2007100401	64	1	2	0	0	0	0
2007100803	31	0	0	0	1	0	0
2007100901	65	0	0	0	1	0	0
2007100902	56	0	0	0	1	0	0
2007101101	24	0	0	0	1	0	0
2007101201	97	1	1	0	0	0	0
2007101302	41	0	0	0	1	0	0
2007101701	56	0	0	0	1	0	0
Average:	53.50		0.40	0.00	0.70	0.00	0.00
Total:		3					

Table A.5 VCalendar 1.5.3.1 task 1 user performance data

Subject Code	Task Completion Time (Sec.)	Number of Incomplete Tasks	Error Rate	Error Time (Sec.)	Success Ratio	Help Time (Sec.)	Help Frequency
2007100302	229	0	0	0	1	0	0
2007100402	223	0	0	0	1	0	0
2007100701	342	0	0	0	1	0	0
2007100801	122	1	1	0	0	0	0
2007100802	154	1	1	0	0	0	0
2007100903	271	0	1	32	1	0	0
2007101001	254	0	0	0	1	0	0
2007101202	129	0	0	0	1	0	0
2007101301	420	0	0	0	1	0	0
2007101702	320	0	0	0	1	0	0
Average:	246.40		0.30	3.20	0.80	0.00	0.00
Total:		2					

Table A.6 VCalendar 1.5.3.1 task 2 user performance data

Subject Code	Task Completion Time (Sec.)	Number of Incomplete Tasks	Error Rate	Error Time (Sec.)	Success Ratio	Help Time (Sec.)	Help Frequency
2007100302	354	0	1	46	1	0	0
2007100402	289	0	0	0	1	0	0
2007100701	109	1	1	0	0	0	0
2007100801	192	0	0	0	1	0	0
2007100802	100	1	0	0	0	0	0
2007100903	129	1	1	0	0	0	0
2007101001	264	0	0	0	1	0	0
2007101202	59	1	2	0	0	0	0
2007101301	852	0	0	0	1	0	0
2007101702	259	0	0	0	1	0	0
Average:	260.70		0.50	4.60	0.60	0.00	0.00
Total:		4					

Table A.7 VCalendar 1.5.3.1 task 3 user performance data

Subject Code	Task Completion Time (Sec.)	Number of Incomplete Tasks	Error Rate	Error Time (Sec.)	Success Ratio	Help Time (Sec.)	Help Frequency
2007100302	126	1	0	0	0	0	0
2007100402	164	1	0	0	0	0	0
2007100701	239	1	0	0	0	0	0
2007100801	254	1	0	0	0	0	0
2007100802	155	1	0	0	0	0	0
2007100903	178	1	0	0	0	0	0
2007101001	142	1	0	0	0	0	0
2007101202	241	1	0	0	0	0	0
2007101301	468	1	0	0	0	0	0
2007101702	304	1	0	0	0	0	0
Average:	227.10		0.00	0.00	0.00	0.00	0.00
Total:		10					

Table A.8 VCalendar 1.5.3.1 task 4 user performance data

Subject Code	Task Completion Time (Sec.)	Number of Incomplete Tasks	Error Rate	Error Time (Sec.)	Success Ratio	Help Time (Sec.)	Help Frequency
2007100302	123	0	0	0	1	0	0
2007100402	66	0	0	0	1	0	0
2007100701	43	0	0	0	1	0	0
2007100801	54	0	0	0	1	0	0
2007100802	137	0	0	0	1	0	0
2007100903	49	0	0	0	1	0	0
2007101001	70	0	0	0	1	0	0
2007101202	45	0	0	0	1	0	0
2007101301	45	0	0	0	1	0	0
2007101702	155	0	0	0	1	0	0
Average:	78.70		0.00	0.00	1.00	0.00	0.00
Total:		0					

APPENDIX B

QUEST EXPERIMENT DATA

B.1 WebCalendar 1.0.5 QUEST usability testing data

B.1.1 Goal-task presentation data

Table B.1 P_1 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_1	P_{1_f}	P_{1_b}	P_1	P_{1_f}	P_{1_b}	P_1	P_{1_f}	P_{1_b}	P_1	P_{1_f}	P_{1_b}
2007100201	0.1999	53.7619	269	0.2186	26.6667	122	0.2935	63.0952	215	0.2692	28.0000	104
2007100301	0.2839	30.0952	106	0.2186	26.6667	122	0.3025	48.0952	159	0.3000	15.0000	50
2007100401	0.2148	39.0952	182	0.1739	8.0000	46	0.1739	8.0000	46	0.2041	20.0000	98
2007100803	0.2540	48.7619	192	0.2186	26.6667	122	0.2104	25.6667	122	0.2692	28.0000	104
2007100901	0.2839	30.0952	106	0.2186	26.6667	122	0.3151	50.0952	159	0.2692	28.0000	104
2007100902	0.2212	51.7619	234	0.2186	26.6667	122	0.2505	49.0952	196	0.2692	28.0000	104
2007101101	0.2839	30.0952	106	0.2186	26.6667	122	0.3029	62.0952	205	0.2692	28.0000	104
2007101201	0.2839	30.0952	106	0.2186	26.6667	122	0.2104	25.6667	122	0.3000	15.0000	50
2007101302	0.2827	31.0952	110	0.2403	20.6667	86	0.3038	47.0952	155	0.2692	28.0000	104
2007101701	0.2839	30.0952	106	0.2186	26.6667	122	0.3184	83.0952	261	0.2692	28.0000	104
Average:	0.2592			0.2163			0.2681			0.2689		

Table B.2 P_2 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_2	P_{2_f}	P_{2_b}	P_2	P_{2_f}	P_{2_b}	P_2	P_{2_f}	P_{2_b}	P_2	P_{2_f}	P_{2_b}
2007100201	0.0000	0	6	0.0000	0	3	0.0000	0	6	0.0000	0	3
2007100301	0.0000	0	3	0.0000	0	3	0.0000	0	5	0.0000	0	1
2007100401	0.0000	0	5	0.0000	0	1	0.0000	0	1	0.0000	0	3
2007100803	0.0000	0	5	0.0000	0	3	0.0000	0	3	0.0000	0	3
2007100901	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007100902	0.0000	0	6	0.0000	0	3	0.0000	0	5	0.0000	0	3
2007101101	0.0000	0	3	0.0000	0	3	0.0000	0	5	0.0000	0	3
2007101201	0.0000	0	3	0.0000	0	3	0.0000	0	3	0.0000	0	1
2007101302	0.0000	0	4	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101701	0.0000	0	3	0.0000	0	3	0.0000	0	6	0.0000	0	3
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.3 P_3 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_3	P_{3_f}	P_{3_b}	P_3	P_{3_f}	P_{3_b}	P_3	P_{3_f}	P_{3_b}	P_3	P_{3_f}	P_{3_b}
2007100201	0.0000	0	6	0.0000	0	3	0.1667	1	6	0.0000	0	3
2007100301	0.0000	0	3	0.0000	0	3	0.0000	0	5	0.0000	0	1
2007100401	0.4000	2	5	0.0000	0	1	0.0000	0	1	0.0000	0	3
2007100803	0.0000	0	5	0.0000	0	3	0.0000	0	3	0.0000	0	3
2007100901	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007100902	0.1667	1	6	0.0000	0	3	0.2000	1	5	0.0000	0	3
2007101101	0.0000	0	3	0.0000	0	3	0.2000	1	5	0.0000	0	3
2007101201	0.0000	0	3	0.0000	0	3	0.0000	0	3	0.0000	0	1
2007101302	0.0000	0	4	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101701	0.0000	0	3	0.0000	0	3	0.1667	1	6	0.0000	0	3
Average:	0.0567			0.0000			0.0733			0.0000		

Table B.4 P_4 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_4	P_{4_f}	P_{4_b}	P_4	P_{4_f}	P_{4_b}	P_4	P_{4_f}	P_{4_b}	P_4	P_{4_f}	P_{4_b}
2007100201	0.0000	0	9	0.0000	0	3	0.0000	0	13	0.0000	0	3
2007100301	0.0000	0	3	0.0000	0	3	0.0000	0	6	0.0000	0	1
2007100401	0.0000	0	5	0.0000	0	1	0.0000	0	1	0.0000	0	3
2007100803	0.0000	0	11	0.0000	0	3	0.0000	0	3	0.0000	0	3
2007100901	0.0000	0	5	0.0000	0	3	0.0000	0	7	0.0000	0	3
2007100902	0.0000	0	7	0.0000	0	3	0.0000	0	5	0.0000	0	3
2007101101	0.0000	0	5	0.0000	0	3	0.0000	0	7	0.0000	0	3
2007101201	0.0000	0	3	0.0000	0	3	0.0000	0	3	0.0000	0	1
2007101302	0.0000	0	4	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101701	0.0000	0	3	0.0000	0	3	0.0000	0	8	0.0000	0	3
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.5 P_5 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_5	P_{5_f}	P_{5_b}	P_5	P_{5_f}	P_{5_b}	P_5	P_{5_f}	P_{5_b}	P_5	P_{5_f}	P_{5_b}
2007100201	0.0000	0	6	0.0000	0	3	0.0000	0	6	0.0000	0	3
2007100301	0.0000	0	3	0.0000	0	3	0.0000	0	5	0.0000	0	1
2007100401	0.0000	0	5	0.0000	0	1	0.0000	0	1	0.0000	0	3
2007100803	0.0000	0	5	0.0000	0	3	0.0000	0	3	0.0000	0	3
2007100901	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007100902	0.0000	0	6	0.0000	0	3	0.0000	0	5	0.0000	0	3
2007101101	0.0000	0	3	0.0000	0	3	0.0000	0	5	0.0000	0	3
2007101201	0.0000	0	3	0.0000	0	3	0.0000	0	3	0.0000	0	1
2007101302	0.0000	0	4	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101701	0.0000	0	3	0.0000	0	3	0.0000	0	6	0.0000	0	3
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.6 P_6 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_6	P_{6_f}	P_{6_b}	P_6	P_{6_f}	P_{6_b}	P_6	P_{6_f}	P_{6_b}	P_6	P_{6_f}	P_{6_b}
2007100201	1.0000	6	6	1.0000	3	3	1.0000	6	6	0.6667	2	3
2007100301	1.0000	3	3	1.0000	3	3	1.0000	5	5	1.0000	1	1
2007100401	0.8000	4	5	1.0000	1	1	1.0000	1	1	0.6667	2	3
2007100803	1.0000	5	5	1.0000	3	3	1.0000	3	3	0.6667	2	3
2007100901	1.0000	3	3	1.0000	3	3	1.0000	4	4	0.6667	2	3
2007100902	0.8333	5	6	1.0000	3	3	0.8000	4	5	0.6667	2	3
2007101101	1.0000	3	3	1.0000	3	3	1.0000	5	5	0.6667	2	3
2007101201	1.0000	3	3	1.0000	3	3	1.0000	3	3	1.0000	1	1
2007101302	1.0000	4	4	1.0000	2	2	1.0000	4	4	0.6667	2	3
2007101701	1.0000	3	3	1.0000	3	3	1.0000	6	6	0.6667	2	3
Average:	0.9633			1.0000			0.9800			0.7333		

Table B.7 P_7 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_7	P_{7_f}	P_{7_b}	P_7	P_{7_f}	P_{7_b}	P_7	P_{7_f}	P_{7_b}	P_7	P_{7_f}	P_{7_b}
2007100201	0.1111	1	9	0.0000	0	3	0.0000	0	13	0.0000	0	3
2007100301	0.3333	1	3	0.0000	0	3	0.0000	0	6	0.0000	0	1
2007100401	0.1667	1	6	0.0000	0	1	0.0000	0	1	0.0000	0	3
2007100803	0.0909	1	11	0.0000	0	3	0.0000	0	3	0.0000	0	3
2007100901	0.2000	1	5	0.0000	0	3	0.0000	0	7	0.0000	0	3
2007100902	0.1250	1	8	0.0000	0	3	0.0000	0	6	0.0000	0	3
2007101101	0.2000	1	5	0.0000	0	3	0.0000	0	7	0.0000	0	3
2007101201	0.3333	1	3	0.0000	0	3	0.0000	0	3	0.0000	0	1
2007101302	0.2500	1	4	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101701	0.3333	1	3	0.0000	0	3	0.0000	0	8	0.0000	0	3
Average:	0.2144			0.0000			0.0000			0.0000		

Table B.8 P_8 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_8	P_{8_f}	P_{8_b}	P_8	P_{8_f}	P_{8_b}	P_8	P_{8_f}	P_{8_b}	P_8	P_{8_f}	P_{8_b}
2007100201	1.0000	6	6	1.0000	3	3	1.0000	6	6	1.0000	3	3
2007100301	1.0000	3	3	1.0000	3	3	1.0000	5	5	1.0000	1	1
2007100401	1.0000	5	5	1.0000	1	1	1.0000	1	1	1.0000	3	3
2007100803	1.0000	5	5	1.0000	3	3	1.0000	3	3	1.0000	3	3
2007100901	1.0000	3	3	1.0000	3	3	1.0000	4	4	1.0000	3	3
2007100902	1.0000	6	6	1.0000	3	3	1.0000	5	5	1.0000	3	3
2007101101	1.0000	3	3	1.0000	3	3	1.0000	5	5	1.0000	3	3
2007101201	1.0000	3	3	1.0000	3	3	1.0000	3	3	1.0000	1	1
2007101302	1.0000	4	4	1.0000	2	2	1.0000	4	4	1.0000	3	3
2007101701	1.0000	3	3	1.0000	3	3	1.0000	6	6	1.0000	3	3
Average:	1.0000			1.0000			1.0000			1.0000		

Table B.9 P_9 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_9	P_{9_f}	P_{9_b}	P_9	P_{9_f}	P_{9_b}	P_9	P_{9_f}	P_{9_b}	P_9	P_{9_f}	P_{9_b}
2007100201	0.0000	0	6	0.0000	0	3	0.0000	0	6	0.0000	0	3
2007100301	0.0000	0	3	0.0000	0	3	0.0000	0	5	0.0000	0	1
2007100401	0.0000	0	5	0.0000	0	1	0.0000	0	1	0.0000	0	3
2007100803	0.0000	0	5	0.0000	0	3	0.0000	0	3	0.0000	0	3
2007100901	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007100902	0.0000	0	6	0.0000	0	3	0.0000	0	5	0.0000	0	3
2007101101	0.0000	0	3	0.0000	0	3	0.0000	0	5	0.0000	0	3
2007101201	0.0000	0	3	0.0000	0	3	0.0000	0	3	0.0000	0	1
2007101302	0.0000	0	4	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101701	0.0000	0	3	0.0000	0	3	0.0000	0	6	0.0000	0	3
Average:	0.0000			0.0000			0.0000			0.0000		

B.1.2 Goal-task interaction data

Table B.10 I_1 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	I_1	I_{1_f}	I_{1_b}	I_1	I_{1_f}	I_{1_b}	I_1	I_{1_f}	I_{1_b}	I_1	I_{1_f}	I_{1_b}
2007100201	0.0000	0	163	0.0000	0	70	0.0000	0	146	0.0000	0	29
2007100301	0.0000	0	37	0.0000	0	45	0.0000	0	58	0.0000	0	24
2007100401	0.0000	0	82	0.0000	0	24	0.0000	0	24	0.0000	0	29
2007100803	0.0000	0	107	0.0000	0	45	0.0000	0	45	0.0000	0	29
2007100901	0.0000	0	37	0.0000	0	45	0.0000	0	56	0.0000	0	29
2007100902	0.0000	0	111	0.0000	0	45	0.0000	0	100	0.0000	0	29
2007101101	0.0000	0	37	0.0000	0	45	0.0000	0	121	0.0000	0	29
2007101201	0.0000	0	37	0.0000	0	45	0.0000	0	70	0.0000	0	26
2007101302	0.0000	0	39	0.0000	0	45	0.0000	0	56	0.0000	0	29
2007101701	0.0000	0	37	0.0000	0	45	0.0000	0	125	0.0000	0	29
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.11 I_2 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	I_2	I_{2_f}	I_{2_b}	I_2	I_{2_f}	I_{2_b}	I_2	I_{2_f}	I_{2_b}	I_2	I_{2_f}	I_{2_b}
2007100201	0.0000	0	17	0.0000	0	4	0.0000	0	21	0.0000	0	2
2007100301	0.1250	1	8	0.0000	0	4	0.0000	0	15	0.0000	0	1
2007100401	0.1000	1	10	1.0000	1	1	0.0000	0	1	1.0000	2	2
2007100803	0.0588	1	17	0.0000	0	4	0.0000	0	8	0.0000	0	2
2007100901	0.0556	1	18	0.0000	0	4	0.0714	1	14	0.0000	0	2
2007100902	0.0000	0	15	0.0000	0	4	0.0000	0	12	0.0000	0	2
2007101101	0.0000	0	17	0.0000	0	4	0.1176	2	17	0.0000	0	2
2007101201	0.0909	1	11	0.0000	0	4	0.0000	0	7	1.0000	1	1
2007101302	0.0000	0	10	0.0000	0	1	0.0000	0	11	0.0000	0	2
2007101701	0.0000	0	10	0.0000	0	4	0.0526	1	19	0.0000	0	2
Average:	0.0430			0.1000			0.0242			0.2000		

Table B.12 I_3 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	I_3	I_{3f}	I_{3b}	I_3	I_{3f}	I_{3b}	I_3	I_{3f}	I_{3b}	I_3	I_{3f}	I_{3b}
2007100201	0.1176	2	17	0.0000	0	4	0.0952	2	21	0.0000	0	2
2007100301	0.1250	1	8	0.0000	0	4	0.2000	3	15	0.0000	0	1
2007100401	0.0000	0	10	0.0000	0	1	0.0000	0	1	0.0000	0	2
2007100803	0.1176	2	17	0.0000	0	4	0.2500	2	8	0.0000	0	2
2007100901	0.1111	2	18	0.0000	0	4	0.1429	2	14	0.0000	0	2
2007100902	0.1333	2	15	0.0000	0	4	0.2500	3	12	0.0000	0	2
2007101101	0.1176	2	17	0.0000	0	4	0.1765	3	17	0.0000	0	2
2007101201	0.1818	2	11	0.0000	0	4	0.2857	2	7	0.0000	0	1
2007101302	0.2000	2	10	0.0000	0	1	0.2727	3	11	0.0000	0	2
2007101701	0.2000	2	10	0.0000	0	4	0.1579	3	19	0.0000	0	2
Average:	0.1304			0.0000			0.1831			0.0000		

Table B.13 I_4 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1	Task 2	Task 3	Task 4
	I_4	I_4	I_4	I_4
2007100201	0	0	0	0
2007100301	0	0	0	1
2007100401	1	1	1	1
2007100803	0	0	1	0
2007100901	0	0	0	0
2007100902	0	0	0	0
2007101101	0	0	0	0
2007101201	0	0	1	1
2007101302	0	1	0	0
2007101701	0	0	0	0
Average:	0.1000	0.2000	0.3000	0.3000

B.1.3 Goal-task efficiency data

Table B.14 E for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	E	T_w	T	E	T_w	T	E	T_w	T	E	T_w	T
2007100201	0.6383	34	94	0.6750	13	40	0.8532	16	109	1.0000	0	8
2007100301	0.5000	44	88	0.7917	10	48	0.7284	44	162	0.0000	54	54
2007100401	0.2959	138	196	0.0000	9	9	0.0000	65	65	0.0000	19	19
2007100803	0.5097	76	155	0.6739	15	46	0.2538	97	130	1.0000	0	12
2007100901	0.8544	15	103	0.5443	36	79	0.5762	89	210	1.0000	0	16
2007100902	0.4762	132	252	1.0000	0	39	0.6930	35	114	1.0000	0	16
2007101101	0.8413	10	63	0.5506	40	89	0.7115	45	156	1.0000	0	6
2007101201	0.6879	44	141	1.0000	0	51	0.7333	16	60	0.0000	7	7
2007101302	0.1383	299	347	0.0000	132	132	0.6351	27	74	0.7727	5	22
2007101701	0.5915	29	71	1.0000	0	46	0.3906	78	128	0.7407	7	27
Average:	0.5534			0.6235			0.5575			0.6513		

B.1.4 Goal-task effectiveness data

Table B.15 R for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1	Task 2	Task 3	Task 4
	R	R	R	R
2007100201	1.0000	1.0000	1.0000	1.0000
2007100301	0.7000	1.0000	1.0000	0.0000
2007100401	0.1000	0.0000	0.0000	0.0000
2007100803	0.9000	1.0000	0.5000	1.0000
2007100901	0.9000	1.0000	0.9000	1.0000
2007100902	1.0000	1.0000	1.0000	1.0000
2007101101	1.0000	1.0000	1.0000	1.0000
2007101201	0.7000	1.0000	0.3000	0.0000
2007101302	1.0000	0.0000	1.0000	1.0000
2007101701	1.0000	1.0000	1.0000	1.0000
Average:	0.8300	0.8000	0.7700	0.7000

B.1.5 Goal-task satisfaction data

Table B.16 S for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1				Task 2				Task 3				Task 4			
	S	S_1	S_2	S_3	S	S_1	S_2	S_3	S	S_1	S_2	S_3	S	S_1	S_2	S_3
2007100201	0.8333	10	3	8	0.8333	9	2	8	0.8333	10	3	8	0.9667	10	0	9
2007100301	0.7000	10	5	6	0.6667	10	5	5	0.7667	10	5	8	0.7000	9	5	7
2007100401	0.6000	9	10	9	0.7000	9	6	8	0.6000	9	10	9	0.7333	9	6	9
2007100803	0.6333	10	7	6	0.9667	10	0	9	0.7667	9	5	9	1.0000	10	0	10
2007100901	0.8000	10	4	8	0.6333	10	6	5	0.6667	10	6	6	0.8667	10	3	9
2007100902	0.8333	10	3	8	1.0000	10	0	10	0.9667	10	1	10	0.9667	10	0	9
2007101101	0.6000	7	4	5	0.7667	9	3	7	0.7000	9	5	7	0.8333	9	2	8
2007101201	0.6000	9	7	6	0.6333	6	2	5	0.8000	8	2	8	0.8667	9	1	8
2007101302	0.6667	10	7	7	0.7000	9	7	9	0.9333	10	1	9	0.8667	9	0	7
2007101701	0.9333	10	1	9	0.9000	10	2	9	0.9000	10	2	9	0.8000	10	3	7
Average:	0.7200				0.7800				0.7933				0.8600			

B.1.6 Goal-task consistency data

Table B.17 C_1 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	C_1	C_{1_f}	C_{1_b}	C_1	C_{1_f}	C_{1_b}	C_1	C_{1_f}	C_{1_b}	C_1	C_{1_f}	C_{1_b}
2007100201	0.0000	0	6	0.0000	0	3	0.0000	0	6	0.0000	0	3
2007100301	0.0000	0	3	0.0000	0	3	0.0000	0	5	0.0000	0	1
2007100401	0.0000	0	5	0.0000	0	1	0.0000	0	1	0.0000	0	3
2007100803	0.0000	0	5	0.0000	0	3	0.0000	0	3	0.0000	0	3
2007100901	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007100902	0.0000	0	6	0.0000	0	3	0.0000	0	5	0.0000	0	3
2007101101	0.0000	0	3	0.0000	0	3	0.0000	0	5	0.0000	0	3
2007101201	0.0000	0	3	0.0000	0	3	0.0000	0	3	0.0000	0	1
2007101302	0.0000	0	4	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101701	0.0000	0	3	0.0000	0	3	0.0000	0	6	0.0000	0	3
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.18 C_2 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	C_2	C_{2_f}	C_{2_b}	C_2	C_{2_f}	C_{2_b}	C_2	C_{2_f}	C_{2_b}	C_2	C_{2_f}	C_{2_b}
2007100201	0.5062	123	243	0.3566	51	143	0.3655	106	290	0.3206	42	131
2007100301	0.3333	47	141	0.3566	51	143	0.4402	81	184	0.8000	16	20
2007100401	0.4845	94	194	0.8000	16	20	0.8000	16	20	0.2992	38	127
2007100803	0.4254	77	181	0.3566	51	143	0.3566	51	143	0.3206	42	131
2007100901	0.3333	47	141	0.3566	51	143	0.4167	70	168	0.3206	42	131
2007100902	0.5070	109	215	0.3566	51	143	0.5000	100	200	0.3206	42	131
2007101101	0.3333	47	141	0.3566	51	143	0.4545	85	187	0.3206	42	131
2007101201	0.3333	47	141	0.3566	51	143	0.3566	51	143	0.8182	18	22
2007101302	0.3734	59	158	0.7500	30	40	0.4132	69	167	0.3206	42	131
2007101701	0.3333	47	141	0.3566	51	143	0.3763	111	295	0.3206	42	131
Average:	0.3963			0.4403			0.4480			0.4162		

Table B.19 C_3 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1	Task 2	Task 3	Task 4
	C_3	C_3	C_3	C_3
2007100201	0	0	0	0
2007100301	0	0	0	0
2007100401	0	0	0	0
2007100803	0	0	0	0
2007100901	0	0	0	0
2007100902	0	0	0	0
2007101101	0	0	0	0
2007101201	0	0	0	0
2007101302	0	0	0	0
2007101701	0	0	0	0
Average:	0.0000	0.0000	0.0000	0.0000

Table B.20 C_4 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	C_4	C_{4_f}	C_{4_b}	C_4	C_{4_f}	C_{4_b}	C_4	C_{4_f}	C_{4_b}	C_4	C_{4_f}	C_{4_b}
2007100201	0.0000	0	6	0.0000	0	3	0.0000	0	6	0.0000	0	3
2007100301	0.0000	0	3	0.0000	0	3	0.0000	0	5	0.0000	0	1
2007100401	0.0000	0	5	0.0000	0	1	0.0000	0	1	0.0000	0	3
2007100803	0.0000	0	5	0.0000	0	3	0.0000	0	3	0.0000	0	3
2007100901	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007100902	0.0000	0	6	0.0000	0	3	0.0000	0	5	0.0000	0	3
2007101101	0.0000	0	3	0.0000	0	3	0.0000	0	5	0.0000	0	3
2007101201	0.0000	0	3	0.0000	0	3	0.0000	0	3	0.0000	0	1
2007101302	0.0000	0	4	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101701	0.0000	0	3	0.0000	0	3	0.0000	0	6	0.0000	0	3
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.21 C_5 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	C_5	C_{5_f}	C_{5_b}	C_5	C_{5_f}	C_{5_b}	C_5	C_{5_f}	C_{5_b}	C_5	C_{5_f}	C_{5_b}
2007100201	0.0000	0	243	0.0000	0	143	0.0000	0	290	0.0000	0	131
2007100301	0.0000	0	141	0.0000	0	143	0.0000	0	184	0.0000	0	20
2007100401	0.0000	0	194	0.0000	0	20	0.0000	0	20	0.0000	0	127
2007100803	0.0000	0	181	0.0000	0	143	0.0000	0	143	0.0000	0	131
2007100901	0.0000	0	141	0.0000	0	143	0.0000	0	168	0.0000	0	131
2007100902	0.0000	0	215	0.0000	0	143	0.0000	0	200	0.0000	0	131
2007101101	0.0000	0	141	0.0000	0	143	0.0000	0	187	0.0000	0	131
2007101201	0.0000	0	141	0.0000	0	143	0.0000	0	143	0.0000	0	22
2007101302	0.0000	0	158	0.0000	0	40	0.0000	0	167	0.0000	0	131
2007101701	0.0000	0	141	0.0000	0	143	0.0000	0	295	0.0000	0	131
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.22 C_6 for WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	C_6	C_{6_f}	C_{6_b}	C_6	C_{6_f}	C_{6_b}	C_6	C_{6_f}	C_{6_b}	C_6	C_{6_f}	C_{6_b}
2007100201	0.0225	10.0000	444	0.0214	5.0000	234	0.0355	16.0000	451	0.0820	15.0000	183
2007100301	0.0124	3.0000	242	0.0214	5.0000	234	0.0261	8.0000	307	0.2400	12.0000	50
2007100401	0.0472	15.0000	318	0.0435	2.0000	46	0.0435	2.0000	46	0.0838	15.0000	179
2007100803	0.0199	7.0000	352	0.0214	5.0000	234	0.0214	5.0000	234	0.0820	15.0000	183
2007100901	0.0124	3.0000	242	0.0214	5.0000	234	0.0250	7.0000	280	0.0820	15.0000	183
2007100902	0.0203	8.0000	394	0.0214	5.0000	234	0.0219	7.0000	320	0.0820	15.0000	183
2007101101	0.0124	3.0000	242	0.0214	5.0000	234	0.0578	19.0000	329	0.0820	15.0000	183
2007101201	0.0124	3.0000	242	0.0214	5.0000	234	0.0214	5.0000	234	0.2400	12.0000	50
2007101302	0.0148	4.0000	270	0.0364	4.0000	110	0.0251	7.0000	279	0.0820	15.0000	183
2007101701	0.0124	3.0000	242	0.0214	5.0000	234	0.0437	20.0000	458	0.0820	15.0000	183
Average:	0.0187			0.0251			0.0321			0.1138		

B.1.7 Navigation presentation data

Table B.23 P_1^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	P_1^{gt}	$P_{1_f}^{gt}$	$P_{1_b}^{gt}$	P_1^{gt}	$P_{1_f}^{gt}$	$P_{1_b}^{gt}$	P_1^{gt}	$P_{1_f}^{gt}$	$P_{1_b}^{gt}$	P_1^{gt}	$P_{1_f}^{gt}$	$P_{1_b}^{gt}$
2007100201	0.0368	6	163	0.0341	7	205	0.0181	10	551	0.0287	6	209
2007100301	0.0294	3	102	0.0341	7	205	0.0181	10	551	0.0287	6	209
2007100401	0.0294	3	102	0.0246	6	244	0.0336	10	298	0.0243	6	247
2007100803	0.0294	3	102	0.0246	6	244	0.0181	10	551	0.0243	6	247
2007100901	0.0294	3	102	0.0246	6	244	0.0246	6	244	0.0243	6	247
2007100902	0.0294	3	102	0.0341	7	205	0.0181	10	551	0.0243	6	247
2007101101	0.0368	6	163	0.0246	6	244	0.0246	6	244	0.0243	6	247
2007101201	0.0294	3	102	0.0341	7	205	0.0181	10	551	0.0243	6	247
2007101302	0.0294	3	102	0.0246	6	244	0.0246	6	244	0.0243	6	247
2007101701	0.0294	3	102	0.0341	7	205	0.0336	10	298	0.0287	6	209
Average:	0.0309			0.0294			0.0232			0.0256		

Table B.24 P_2^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	P_2^{gt}	$P_{2_f}^{gt}$	$P_{2_b}^{gt}$	P_2^{gt}	$P_{2_f}^{gt}$	$P_{2_b}^{gt}$	P_2^{gt}	$P_{2_f}^{gt}$	$P_{2_b}^{gt}$	P_2^{gt}	$P_{2_f}^{gt}$	$P_{2_b}^{gt}$
2007100201	0.0000	0	2	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100301	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100401	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100803	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100901	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100902	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007101101	0.0000	0	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101201	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007101302	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101701	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.25 P_3^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	P_3^{gt}	$P_{3_f}^{gt}$	$P_{3_b}^{gt}$	P_3^{gt}	$P_{3_f}^{gt}$	$P_{3_b}^{gt}$	P_3^{gt}	$P_{3_f}^{gt}$	$P_{3_b}^{gt}$	P_3^{gt}	$P_{3_f}^{gt}$	$P_{3_b}^{gt}$
2007100201	0.0000	0	2	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100301	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100401	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100803	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100901	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100902	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007101101	0.0000	0	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101201	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007101302	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101701	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.26 P_4^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	P_4^{gt}	$P_{4_f}^{gt}$	$P_{4_b}^{gt}$	P_4^{gt}	$P_{4_f}^{gt}$	$P_{4_b}^{gt}$	P_4^{gt}	$P_{4_f}^{gt}$	$P_{4_b}^{gt}$	P_4^{gt}	$P_{4_f}^{gt}$	$P_{4_b}^{gt}$
2007100201	1.0000	2	2	1.0000	2	2	1.0000	3	3	1.0000	2	2
2007100301	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	2	2
2007100401	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	2	2
2007100803	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	2	2
2007100901	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007100902	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	2	2
2007101101	1.0000	2	2	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007101201	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	2	2
2007101302	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007101701	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	2	2
Average:	1.0000			1.0000			1.0000			1.0000		

Table B.27 P_5^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	P_5^{gt}	$P_{5_f}^{gt}$	$P_{5_b}^{gt}$	P_5^{gt}	$P_{5_f}^{gt}$	$P_{5_b}^{gt}$	P_5^{gt}	$P_{5_f}^{gt}$	$P_{5_b}^{gt}$	P_5^{gt}	$P_{5_f}^{gt}$	$P_{5_b}^{gt}$
2007100201	1.0000	2	2	1.0000	2	2	1.0000	3	3	1.0000	2	2
2007100301	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	2	2
2007100401	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	2	2
2007100803	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	2	2
2007100901	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007100902	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	2	2
2007101101	1.0000	2	2	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007101201	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	2	2
2007101302	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007101701	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	2	2
Average:	1.0000			1.0000			1.0000			1.0000		

B.1.8 Navigation interaction data

Table B.28 I_1^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1	Nav. For Task 2	Nav. For Task 3	Nav. For Task 4
	I_1^{gt}	I_1^{gt}	I_1^{gt}	I_1^{gt}
2007100201	0	0	0	0
2007100301	0	0	0	1
2007100401	0	0	0	0
2007100803	0	0	0	0
2007100901	0	0	0	0
2007100902	0	0	0	0
2007101101	0	0	0	0
2007101201	0	0	0	0
2007101302	0	0	0	0
2007101701	0	0	0	0
Average:	0.0000	0.0000	0.0000	0.1000

B.1.9 Navigation satisfaction data

Table B.29 S^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	S^{gt}	S_1^{gt}	S_2^{gt}	S^{gt}	S_1^{gt}	S_2^{gt}	S^{gt}	S_1^{gt}	S_2^{gt}	S^{gt}	S_1^{gt}	S_2^{gt}
2007100201	0.8500	9	8	0.8500	9	8	0.8500	9	8	0.8500	9	8
2007100301	0.7500	10	5	0.5500	8	3	0.6500	8	5	0.6500	8	5
2007100401	0.6500	7	6	0.9500	10	9	1.0000	10	10	1.0000	10	10
2007100803	0.6500	9	4	0.9000	10	8	0.9000	10	8	0.9000	10	8
2007100901	1.0000	10	10	1.0000	10	10	1.0000	10	10	1.0000	10	10
2007100902	1.0000	10	10	1.0000	10	10	1.0000	10	10	1.0000	10	10
2007101101	0.6500	7	6	0.8000	9	7	0.9000	9	9	0.8000	9	7
2007101201	0.7000	8	6	0.6000	8	4	0.6000	8	4	0.7000	10	4
2007101302	0.9000	9	9	0.9000	9	9	0.9000	9	9	0.9000	9	9
2007101701	1.0000	10	10	1.0000	10	10	1.0000	10	10	1.0000	10	10
Average:	0.8150			0.8550			0.8800			0.8800		

B.1.10 Navigation consistency data

Table B.30 C_1^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	C_1^{gt}	$C_{1_f}^{gt}$	$C_{1_b}^{gt}$	C_1^{gt}	$C_{1_f}^{gt}$	$C_{1_b}^{gt}$	C_1^{gt}	$C_{1_f}^{gt}$	$C_{1_b}^{gt}$	C_1^{gt}	$C_{1_f}^{gt}$	$C_{1_b}^{gt}$
2007100201	0.0000	0	2	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100301	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100401	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100803	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100901	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100902	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007101101	0.0000	0	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101201	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007101302	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101701	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.31 C_2^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	C_2^{gt}	$C_{2_f}^{gt}$	$C_{2_b}^{gt}$	C_2^{gt}	$C_{2_f}^{gt}$	$C_{2_b}^{gt}$	C_2^{gt}	$C_{2_f}^{gt}$	$C_{2_b}^{gt}$	C_2^{gt}	$C_{2_f}^{gt}$	$C_{2_b}^{gt}$
2007100201	0.3926	64	163	0.2000	41	205	0.8258	455	551	0.2153	45	209
2007100301	0.1961	20	102	0.2000	41	205	0.8258	455	551	0.2153	45	209
2007100401	0.1961	20	102	0.5246	128	244	0.5805	173	298	0.5344	132	247
2007100803	0.1961	20	102	0.5246	128	244	0.8258	455	551	0.5344	132	247
2007100901	0.1961	20	102	0.5246	128	244	0.5246	128	244	0.5344	132	247
2007100902	0.1961	20	102	0.2000	41	205	0.8258	455	551	0.5344	132	247
2007101101	0.3926	64	163	0.5246	128	244	0.5246	128	244	0.5344	132	247
2007101201	0.1961	20	102	0.2000	41	205	0.8258	455	551	0.5344	132	247
2007101302	0.1961	20	102	0.5246	128	244	0.5246	128	244	0.5344	132	247
2007101701	0.1961	20	102	0.2000	41	205	0.5805	173	298	0.2153	45	209
Average:	0.2354			0.3623			0.6864			0.4387		

Table B.32 C_3^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	C_3^{gt}	$C_{3_f}^{gt}$	$C_{3_b}^{gt}$	C_3^{gt}	$C_{3_f}^{gt}$	$C_{3_b}^{gt}$	C_3^{gt}	$C_{3_f}^{gt}$	$C_{3_b}^{gt}$	C_3^{gt}	$C_{3_f}^{gt}$	$C_{3_b}^{gt}$
2007100201	0.0000	0	2	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100301	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100401	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100803	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007100901	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100902	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007101101	0.0000	0	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101201	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
2007101302	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101701	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	2
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.33 C_4^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	C_4^{gt}	$C_{4_f}^{gt}$	$C_{4_b}^{gt}$	C_4^{gt}	$C_{4_f}^{gt}$	$C_{4_b}^{gt}$	C_4^{gt}	$C_{4_f}^{gt}$	$C_{4_b}^{gt}$	C_4^{gt}	$C_{4_f}^{gt}$	$C_{4_b}^{gt}$
2007100201	0.0000	0	163	0.0000	0	205	0.0000	0	551	0.0000	0	209
2007100301	0.0000	0	102	0.0000	0	205	0.0000	0	551	0.0000	0	209
2007100401	0.0000	0	102	0.0000	0	244	0.0000	0	298	0.0000	0	247
2007100803	0.0000	0	102	0.0000	0	244	0.0000	0	551	0.0000	0	247
2007100901	0.0000	0	102	0.0000	0	244	0.0000	0	244	0.0000	0	247
2007100902	0.0000	0	102	0.0000	0	205	0.0000	0	551	0.0000	0	247
2007101101	0.0000	0	163	0.0000	0	244	0.0000	0	244	0.0000	0	247
2007101201	0.0000	0	102	0.0000	0	205	0.0000	0	551	0.0000	0	247
2007101302	0.0000	0	102	0.0000	0	244	0.0000	0	244	0.0000	0	247
2007101701	0.0000	0	102	0.0000	0	205	0.0000	0	298	0.0000	0	209
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.34 C_5^{gt} for locating WebCalendar 1.0.5 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	C_5^{gt}	$C_{5_f}^{gt}$	$C_{5_b}^{gt}$	C_5^{gt}	$C_{5_f}^{gt}$	$C_{5_b}^{gt}$	C_5^{gt}	$C_{5_f}^{gt}$	$C_{5_b}^{gt}$	C_5^{gt}	$C_{5_f}^{gt}$	$C_{5_b}^{gt}$
2007100201	0.0140	3.0000	214	0.0081	2.0000	247	0.0057	4.0000	698	0.0080	2.0000	251
2007100301	0.0081	1.0000	123	0.0081	2.0000	247	0.0057	4.0000	698	0.0080	2.0000	251
2007100401	0.0081	1.0000	123	0.0067	2.0000	297	0.0105	4.0000	382	0.0067	2.0000	300
2007100803	0.0081	1.0000	123	0.0067	2.0000	297	0.0057	4.0000	698	0.0067	2.0000	300
2007100901	0.0081	1.0000	123	0.0067	2.0000	297	0.0067	2.0000	297	0.0067	2.0000	300
2007100902	0.0081	1.0000	123	0.0081	2.0000	247	0.0057	4.0000	698	0.0067	2.0000	300
2007101101	0.0140	3.0000	214	0.0067	2.0000	297	0.0067	2.0000	297	0.0067	2.0000	300
2007101201	0.0081	1.0000	123	0.0081	2.0000	247	0.0057	4.0000	698	0.0067	2.0000	300
2007101302	0.0081	1.0000	123	0.0067	2.0000	297	0.0067	2.0000	297	0.0067	2.0000	300
2007101701	0.0081	1.0000	123	0.0081	2.0000	247	0.0105	4.0000	382	0.0080	2.0000	251
Average:	0.0093			0.0074			0.0070			0.0071		

B.2 VCalendar 1.5.3.1 QUEST usability testing data

B.2.1 Goal-task presentation data

Table B.35 P_1 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_1	P_{1_f}	P_{1_b}	P_1	P_{1_f}	P_{1_b}	P_1	P_{1_f}	P_{1_b}	P_1	P_{1_f}	P_{1_b}
2007100302	0.3951	40.7000	103	0.4346	133.000	306	0.3566	46.0000	129	0.5424	64.0000	118
2007100402	0.3951	40.7000	103	0.4535	78.0000	172	0.3657	49.0000	134	0.5269	49.0000	93
2007100701	0.3951	40.7000	103	0.4259	23.0000	54	0.4824	41.0000	85	0.5269	49.0000	93
2007100801	0.4500	27.0000	60	0.4191	57.0000	136	0.3657	49.0000	134	0.5269	49.0000	93
2007100802	0.3077	32.0000	104	0.5000	46.0000	92	0.3657	49.0000	134	0.5269	49.0000	93
2007100903	0.3951	40.7000	103	0.5000	46.0000	92	0.3657	49.0000	134	0.5269	49.0000	93
2007101001	0.3951	40.7000	103	0.4535	78.0000	172	0.4649	53.0000	114	0.5269	49.0000	93
2007101202	0.3951	40.7000	103	0.5246	32.0000	61	0.3657	49.0000	134	0.5269	49.0000	93
2007101301	0.3951	40.7000	103	0.4535	78.0000	172	0.3657	49.0000	134	0.5269	49.0000	93
2007101702	0.3951	40.7000	103	0.3797	71.0000	187	0.3524	37.0000	105	0.5269	49.0000	93
Average:	0.3919			0.4544			0.3850			0.5284		

Table B.36 P_2 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_2	P_{2_f}	P_{2_b}	P_2	P_{2_f}	P_{2_b}	P_2	P_{2_f}	P_{2_b}	P_2	P_{2_f}	P_{2_b}
2007100302	0.0000	0	3	0.1000	1	10	0.0000	0	4	0.0000	0	3
2007100402	0.0000	0	3	0.0000	0	6	0.0000	0	4	0.0000	0	3
2007100701	0.0000	0	3	0.0000	0	2	0.0000	0	3	0.0000	0	3
2007100801	0.0000	0	2	0.0000	0	5	0.0000	0	4	0.0000	0	3
2007100802	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007100903	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007101001	0.0000	0	3	0.0000	0	6	0.0000	0	4	0.0000	0	3
2007101202	0.0000	0	3	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101301	0.0000	0	3	0.0000	0	6	0.0000	0	4	0.0000	0	3
2007101702	0.0000	0	3	0.0000	0	6	0.0000	0	3	0.0000	0	3
Average:	0.0000			0.0100			0.0000			0.0000		

Table B.37 P_3 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_3	P_{3_f}	P_{3_b}	P_3	P_{3_f}	P_{3_b}	P_3	P_{3_f}	P_{3_b}	P_3	P_{3_f}	P_{3_b}
2007100302	0.3333	1	3	0.4000	4	10	0.7500	3	4	0.6667	2	3
2007100402	0.3333	1	3	0.3333	2	6	0.7500	3	4	0.6667	2	3
2007100701	0.3333	1	3	0.5000	1	2	0.6667	2	3	0.6667	2	3
2007100801	0.0000	0	2	0.2000	1	5	0.7500	3	4	0.6667	2	3
2007100802	0.3333	1	3	0.6667	2	3	0.7500	3	4	0.6667	2	3
2007100903	0.3333	1	3	0.6667	2	3	0.7500	3	4	0.6667	2	3
2007101001	0.3333	1	3	0.3333	2	6	0.5000	2	4	0.6667	2	3
2007101202	0.3333	1	3	1.0000	2	2	0.7500	3	4	0.6667	2	3
2007101301	0.3333	1	3	0.3333	2	6	0.7500	3	4	0.6667	2	3
2007101702	0.3333	1	3	0.5000	3	6	1.0000	3	3	0.6667	2	3
Average:	0.3000			0.4933			0.7417			0.6667		

Table B.38 P_4 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_4	P_{4_f}	P_{4_b}	P_4	P_{4_f}	P_{4_b}	P_4	P_{4_f}	P_{4_b}	P_4	P_{4_f}	P_{4_b}
2007100302	0.0000	0	4	0.1515	5	33	0.0000	0	4	0.0000	0	12
2007100402	0.0000	0	4	0.1364	3	22	0.0000	0	7	0.0000	0	3
2007100701	0.0000	0	4	0.0000	0	3	0.0000	0	3	0.0000	0	3
2007100801	0.0000	0	2	0.2000	1	5	0.0000	0	7	0.0000	0	3
2007100802	0.0000	0	3	0.0000	0	7	0.0000	0	7	0.0000	0	3
2007100903	0.0000	0	4	0.0000	0	6	0.0000	0	7	0.0000	0	3
2007101001	0.0000	0	4	0.1111	1	9	0.0000	0	6	0.0000	0	3
2007101202	0.0000	0	4	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101301	0.0000	0	4	0.1250	1	8	0.0000	0	6	0.0000	0	3
2007101702	0.0000	0	4	0.1429	1	7	0.0000	0	3	0.0000	0	3
Average:	0.0000			0.0867			0.0000			0.0000		

Table B.39 P_5 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_5	P_{5_f}	P_{5_b}	P_5	P_{5_f}	P_{5_b}	P_5	P_{5_f}	P_{5_b}	P_5	P_{5_f}	P_{5_b}
2007100302	0.0000	0	3	0.0000	0	10	0.0000	0	4	0.0000	0	3
2007100402	0.0000	0	3	0.0000	0	6	0.0000	0	4	0.0000	0	3
2007100701	0.0000	0	3	0.0000	0	2	0.0000	0	3	0.0000	0	3
2007100801	0.0000	0	2	0.0000	0	5	0.0000	0	4	0.0000	0	3
2007100802	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007100903	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007101001	0.0000	0	3	0.0000	0	6	0.0000	0	4	0.0000	0	3
2007101202	0.0000	0	3	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101301	0.0000	0	3	0.0000	0	6	0.0000	0	4	0.0000	0	3
2007101702	0.0000	0	3	0.0000	0	6	0.0000	0	3	0.0000	0	3
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.40 P_6 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_6	P_{6_f}	P_{6_b}	P_6	P_{6_f}	P_{6_b}	P_6	P_{6_f}	P_{6_b}	P_6	P_{6_f}	P_{6_b}
2007100302	0.6667	2	3	0.9000	9	10	0.7500	3	4	1.0000	3	3
2007100402	0.6667	2	3	1.0000	6	6	0.7500	3	4	1.0000	3	3
2007100701	0.6667	2	3	1.0000	2	2	1.0000	3	3	1.0000	3	3
2007100801	1.0000	2	2	1.0000	5	5	0.7500	3	4	1.0000	3	3
2007100802	0.6667	2	3	1.0000	3	3	0.7500	3	4	1.0000	3	3
2007100903	0.6667	2	3	1.0000	3	3	0.7500	3	4	1.0000	3	3
2007101001	0.6667	2	3	1.0000	6	6	1.0000	4	4	1.0000	3	3
2007101202	0.6667	2	3	1.0000	2	2	0.7500	3	4	1.0000	3	3
2007101301	0.6667	2	3	1.0000	6	6	0.7500	3	4	1.0000	3	3
2007101702	0.6667	2	3	0.8333	5	6	0.6667	2	3	1.0000	3	3
Average:	0.7000			0.9733			0.7917			1.0000		

Table B.41 P_7 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_7	P_{7_f}	P_{7_b}	P_7	P_{7_f}	P_{7_b}	P_7	P_{7_f}	P_{7_b}	P_7	P_{7_f}	P_{7_b}
2007100302	0.1667	1	6	0.2941	10	34	0.6000	3	5	0.7500	9	12
2007100402	0.1667	1	6	0.2273	5	22	0.5000	4	8	1.0000	3	3
2007100701	0.1667	1	6	0.6667	2	3	1.0000	3	3	1.0000	3	3
2007100801	0.5000	1	2	0.4000	2	5	0.5000	4	8	1.0000	3	3
2007100802	0.2500	1	4	0.4286	3	7	0.5000	4	8	1.0000	3	3
2007100903	0.1667	1	6	0.5000	3	6	0.5000	4	8	1.0000	3	3
2007101001	0.1667	1	6	0.3333	3	9	0.8333	5	6	1.0000	3	3
2007101202	0.1667	1	6	0.5000	1	2	0.6000	3	5	1.0000	3	3
2007101301	0.1667	1	6	0.3750	3	8	0.5714	4	7	1.0000	3	3
2007101702	0.1667	1	6	0.2500	2	8	0.5000	2	4	1.0000	3	3
Average:	0.2083			0.3975			0.6105			0.9750		

Table B.42 P_8 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_8	P_{8_f}	P_{8_b}	P_8	P_{8_f}	P_{8_b}	P_8	P_{8_f}	P_{8_b}	P_8	P_{8_f}	P_{8_b}
2007100302	1.0000	3	3	1.0000	10	10	1.0000	4	4	1.0000	3	3
2007100402	1.0000	3	3	1.0000	6	6	1.0000	4	4	1.0000	3	3
2007100701	1.0000	3	3	1.0000	2	2	1.0000	3	3	1.0000	3	3
2007100801	1.0000	2	2	1.0000	5	5	1.0000	4	4	1.0000	3	3
2007100802	1.0000	3	3	1.0000	3	3	1.0000	4	4	1.0000	3	3
2007100903	1.0000	3	3	1.0000	3	3	1.0000	4	4	1.0000	3	3
2007101001	1.0000	3	3	1.0000	6	6	1.0000	4	4	1.0000	3	3
2007101202	1.0000	3	3	1.0000	2	2	1.0000	4	4	1.0000	3	3
2007101301	1.0000	3	3	1.0000	6	6	1.0000	4	4	1.0000	3	3
2007101702	1.0000	3	3	1.0000	6	6	1.0000	3	3	1.0000	3	3
Average:	1.0000			1.0000			1.0000			1.0000		

Table B.43 P_9 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	P_9	P_{9_f}	P_{9_b}	P_9	P_{9_f}	P_{9_b}	P_9	P_{9_f}	P_{9_b}	P_9	P_{9_f}	P_{9_b}
2007100302	0.0000	0	3	0.0000	0	10	0.0000	0	4	0.0000	0	3
2007100402	0.0000	0	3	0.0000	0	6	0.0000	0	4	0.0000	0	3
2007100701	0.0000	0	3	0.0000	0	2	0.0000	0	3	0.0000	0	3
2007100801	0.0000	0	2	0.0000	0	5	0.0000	0	4	0.0000	0	3
2007100802	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007100903	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007101001	0.0000	0	3	0.0000	0	6	0.0000	0	4	0.0000	0	3
2007101202	0.0000	0	3	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101301	0.0000	0	3	0.0000	0	6	0.0000	0	4	0.0000	0	3
2007101702	0.0000	0	3	0.0000	0	6	0.0000	0	3	0.0000	0	3
Average:	0.0000			0.0000			0.0000			0.0000		

B.2.2 Goal-task interaction data

Table B.44 I_1 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	I_1	I_{1_f}	I_{1_b}	I_1	I_{1_f}	I_{1_b}	I_1	I_{1_f}	I_{1_b}	I_1	I_{1_f}	I_{1_b}
2007100302	0.0741	2	27	0.0397	6	151	0.0769	2	26	0.0536	3	56
2007100402	0.0741	2	27	0.0392	4	102	0.0408	2	49	0.1429	3	21
2007100701	0.0741	2	27	0.0000	0	27	0.1000	2	20	0.1429	3	21
2007100801	0.1176	2	17	0.0233	2	86	0.0408	2	49	0.1429	3	21
2007100802	0.0870	2	23	0.0455	2	44	0.0408	2	49	0.1429	3	21
2007100903	0.0741	2	27	0.0455	2	44	0.0408	2	49	0.1429	3	21
2007101001	0.0741	2	27	0.0392	4	102	0.0317	2	63	0.1429	3	21
2007101202	0.0741	2	27	0.1000	2	20	0.0769	2	26	0.1429	3	21
2007101301	0.0741	2	27	0.0392	4	102	0.0408	2	49	0.1429	3	21
2007101702	0.0741	2	27	0.0471	4	85	0.0769	2	26	0.1429	3	21
Average:	0.0797			0.0419			0.0567			0.1339		

Table B.45 I_2 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	I_2	I_{2_f}	I_{2_b}	I_2	I_{2_f}	I_{2_b}	I_2	I_{2_f}	I_{2_b}	I_2	I_{2_f}	I_{2_b}
2007100302	0.0000	0	16	0.0189	1	53	0.0000	0	8	0.0000	0	11
2007100402	0.0000	0	13	0.0000	0	31	0.0000	0	11	0.0000	0	2
2007100701	0.0000	0	15	0.2500	1	4	0.0000	0	5	0.0000	0	2
2007100801	0.1667	1	6	0.0000	0	25	0.0000	0	10	0.0000	0	2
2007100802	0.1429	1	7	0.0000	0	6	0.0000	0	10	0.0000	0	2
2007100903	0.0556	1	18	0.1429	1	7	0.0000	0	11	0.0000	0	2
2007101001	0.0000	0	14	0.0000	0	17	0.0000	0	10	0.0000	0	2
2007101202	0.0000	0	12	0.6667	2	3	0.0000	0	8	0.0000	0	2
2007101301	0.0000	0	14	0.0000	0	16	0.0000	0	10	0.0000	0	2
2007101702	0.0000	0	14	0.0000	0	15	0.0000	0	6	0.0000	0	2
Average:	0.0365			0.1078			0.0000			0.2000		

Table B.46 I_3 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	I_3	I_{3f}	I_{3b}	I_3	I_{3f}	I_{3b}	I_3	I_{3f}	I_{3b}	I_3	I_{3f}	I_{3b}
2007100302	0.1875	3	16	0.2264	12	53	0.1250	1	8	0.4545	5	11
2007100402	0.2308	3	13	0.1290	4	31	0.0909	1	11	0.5000	1	2
2007100701	0.2000	3	15	0.2500	1	4	0.0000	0	5	0.5000	1	2
2007100801	0.1667	1	6	0.2000	5	25	0.1000	1	10	0.5000	1	2
2007100802	0.2857	2	7	0.3333	2	6	0.1000	1	10	0.5000	1	2
2007100903	0.1667	3	18	0.2857	2	7	0.0909	1	11	0.5000	1	2
2007101001	0.2143	3	14	0.2353	4	17	0.0000	0	10	0.5000	1	2
2007101202	0.2500	3	12	0.3333	1	3	0.1250	1	8	0.5000	1	2
2007101301	0.2143	3	14	0.2500	4	16	0.1000	1	10	0.5000	1	2
2007101702	0.2143	3	14	0.3333	5	15	0.1667	1	6	0.5000	1	2
Average:	0.2130			0.2576			0.0898			0.4955		

Table B.47 I_4 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1	Task 2	Task 3	Task 4
	I_4	I_4	I_4	I_4
2007100302	0	0	1	0
2007100402	0	0	1	0
2007100701	0	1	1	0
2007100801	1	0	1	0
2007100802	1	1	1	0
2007100903	0	1	1	0
2007101001	0	0	1	0
2007101202	0	1	1	0
2007101301	0	0	1	0
2007101702	0	0	1	0
Average:	0.2000	0.4000	1.0000	0.0000

B.2.3 Goal-task efficiency data

Table B.48 E for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	E	T_w	T	E	T_w	T	E	T_w	T	E	T_w	T
2007100302	0.5310	53	113	0.1824	242	296	0.0000	106	106	0.1000	90	100
2007100402	0.5412	78	170	0.2018	174	218	0.0000	135	135	0.4348	13	23
2007100701	0.4806	67	129	0.0000	40	40	0.0000	221	221	1.0000	0	7
2007100801	0.0000	92	92	0.4173	81	139	0.0000	237	237	0.3846	8	13
2007100802	0.0000	139	139	0.0000	71	71	0.0000	129	129	0.0494	77	81
2007100903	0.6142	49	127	0.0000	96	96	0.0000	157	157	0.4167	7	12
2007101001	0.4936	79	156	0.2379	157	206	0.0000	117	117	0.4483	16	29
2007101202	0.7500	15	60	0.0000	23	23	0.0000	203	203	0.3333	18	27
2007101301	0.6256	73	195	0.0538	738	780	0.0000	438	438	0.3636	7	11
2007101702	0.4828	90	174	0.3253	112	166	0.0000	270	270	0.2639	53	72
Average:	0.4519			0.1419			0.0000			0.3795		

B.2.4 Goal-task effectiveness data

Table B.49 R for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1	Task 2	Task 3	Task 4
	R	R	R	R
2007100302	1.0000	0.9000	0.0000	1.0000
2007100402	1.0000	0.9000	0.0000	1.0000
2007100701	1.0000	0.0000	0.0000	1.0000
2007100801	0.0000	0.6000	0.0000	1.0000
2007100802	0.0000	0.0000	0.0000	1.0000
2007100903	1.0000	0.0000	0.0000	1.0000
2007101001	1.0000	0.9000	0.0000	1.0000
2007101202	0.8000	0.0000	0.0000	1.0000
2007101301	1.0000	0.9000	0.0000	1.0000
2007101702	1.0000	0.9000	0.0000	1.0000
Average:	0.7800	0.5100	0.0000	1.0000

6.6.2.5 Goal-task satisfaction data

Table B.50 S for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1				Task 2				Task 3				Task 4			
	S	S_1	S_2	S_3	S	S_1	S_2	S_3	S	S_1	S_2	S_3	S	S_1	S_2	S_3
2007100302	0.9333	10	0	8	0.3000	3	7	3	0.2333	0	7	4	0.8667	10	1	7
2007100402	0.6000	8	6	6	0.1000	1	10	2	0.2000	3	8	1	0.8667	9	2	9
2007100701	0.6333	6	3	6	0.0000	0	10	0	0.2667	2	6	2	0.8667	8	0	8
2007100801	0.5667	8	8	7	0.3667	2	3	2	0.5000	5	6	6	0.8667	8	0	8
2007100802	0.6333	9	3	3	0.0667	0	8	0	0.3667	3	8	6	0.7333	9	5	8
2007100903	0.8000	9	3	8	0.1333	1	9	2	0.2667	3	9	4	0.8667	10	2	8
2007101001	0.9000	9	0	8	0.3667	1	4	4	0.6000	7	5	6	0.9667	10	0	9
2007101202	0.8667	8	0	8	0.4667	3	1	2	0.4333	5	4	2	0.8333	7	0	8
2007101301	0.8333	8	1	8	0.2333	2	7	2	0.4333	5	7	5	0.9333	10	0	8
2007101702	0.7000	6	1	6	0.2333	2	8	3	0.2667	5	10	3	0.8333	9	2	8
Average:	0.7467				0.2267				0.3567				0.8633			

B.2.6 Goal-task consistency data

Table B.51 C_1 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	C_1	C_{1_f}	C_{1_b}	C_1	C_{1_f}	C_{1_b}	C_1	C_{1_f}	C_{1_b}	C_1	C_{1_f}	C_{1_b}
2007100302	0.3333	1	3	0.4000	4	10	0.2500	1	4	0.3333	1	3
2007100402	0.3333	1	3	0.3333	2	6	0.2500	1	4	0.3333	1	3
2007100701	0.3333	1	3	0.0000	0	2	0.3333	1	3	0.3333	1	3
2007100801	0.5000	1	2	0.2000	1	5	0.2500	1	4	0.3333	1	3
2007100802	0.3333	1	3	0.3333	1	3	0.2500	1	4	0.3333	1	3
2007100903	0.3333	1	3	0.3333	1	3	0.2500	1	4	0.3333	1	3
2007101001	0.3333	1	3	0.3333	2	6	0.2500	1	4	0.3333	1	3
2007101202	0.3333	1	3	0.5000	1	2	0.2500	1	4	0.3333	1	3
2007101301	0.3333	1	3	0.3333	2	6	0.2500	1	4	0.3333	1	3
2007101702	0.3333	1	3	0.3333	2	6	0.3333	1	3	0.3333	1	3
Average:	0.3500			0.3100			0.2667			0.3333		

Table B.52 C_2 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	C_2	C_{2_f}	C_{2_b}	C_2	C_{2_f}	C_{2_b}	C_2	C_{2_f}	C_{2_b}	C_2	C_{2_f}	C_{2_b}
2007100302	0.2786	39	140	0.3158	126	399	0.2826	39	138	0.3585	38	106
2007100402	0.2786	39	140	0.4077	117	287	0.2806	39	139	0.3824	39	102
2007100701	0.2786	39	140	0.4333	39	90	0.3824	39	102	0.3824	39	102
2007100801	0.3939	39	99	0.4255	117	275	0.2806	39	139	0.3824	39	102
2007100802	0.2889	39	135	0.3750	39	104	0.2806	39	139	0.3824	39	102
2007100903	0.2786	39	140	0.3750	39	104	0.2806	39	139	0.3824	39	102
2007101001	0.2786	39	140	0.4077	117	287	0.4149	78	188	0.3824	39	102
2007101202	0.2786	39	140	0.0000	0	17	0.2806	39	139	0.3824	39	102
2007101301	0.2786	39	140	0.4077	117	287	0.2806	39	139	0.3824	39	102
2007101702	0.2786	39	140	0.3291	78	237	0.0000	0	53	0.3824	39	102
Average:	0.2911			0.3477			0.2763			0.3800		

Table B.53 C_3 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1	Task 2	Task 3	Task 4
	C_3	C_3	C_3	C_3
2007100302	0	0	0	0
2007100402	0	0	0	0
2007100701	0	0	0	0
2007100801	0	0	0	0
2007100802	0	0	0	0
2007100903	0	0	0	0
2007101001	0	0	0	0
2007101202	0	0	0	0
2007101301	0	0	0	0
2007101702	0	0	0	0
Average:	0.0000	0.0000	0.0000	0.0000

Table B.54 C_4 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	C_4	C_{4_f}	C_{4_b}	C_4	C_{4_f}	C_{4_b}	C_4	C_{4_f}	C_{4_b}	C_4	C_{4_f}	C_{4_b}
2007100302	0.0000	0	3	0.0000	0	10	0.0000	0	4	0.0000	0	3
2007100402	0.0000	0	3	0.0000	0	6	0.0000	0	4	0.0000	0	3
2007100701	0.0000	0	3	0.0000	0	2	0.0000	0	3	0.0000	0	3
2007100801	0.0000	0	2	0.0000	0	5	0.0000	0	4	0.0000	0	3
2007100802	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007100903	0.0000	0	3	0.0000	0	3	0.0000	0	4	0.0000	0	3
2007101001	0.0000	0	3	0.0000	0	6	0.0000	0	4	0.0000	0	3
2007101202	0.0000	0	3	0.0000	0	2	0.0000	0	4	0.0000	0	3
2007101301	0.0000	0	3	0.0000	0	6	0.0000	0	4	0.0000	0	3
2007101702	0.0000	0	3	0.0000	0	6	0.0000	0	3	0.0000	0	3
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.55 C_5 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	C_5	C_{5_f}	C_{5_b}	C_5	C_{5_f}	C_{5_b}	C_5	C_{5_f}	C_{5_b}	C_5	C_{5_f}	C_{5_b}
2007100302	0.0000	0	140	0.0000	0	399	0.0000	0	138	0.0000	0	106
2007100402	0.0000	0	140	0.0000	0	287	0.0000	0	139	0.0000	0	102
2007100701	0.0000	0	140	0.0000	0	90	0.0000	0	102	0.0000	0	102
2007100801	0.0000	0	99	0.0000	0	275	0.0000	0	139	0.0000	0	102
2007100802	0.0000	0	135	0.0000	0	104	0.0000	0	139	0.0000	0	102
2007100903	0.0000	0	140	0.0000	0	104	0.0000	0	139	0.0000	0	102
2007101001	0.0000	0	140	0.0000	0	287	0.0000	0	188	0.0000	0	102
2007101202	0.0000	0	140	0.0000	0	17	0.0000	0	139	0.0000	0	102
2007101301	0.0000	0	140	0.0000	0	287	0.0000	0	139	0.0000	0	102
2007101702	0.0000	0	140	0.0000	0	237	0.0000	0	53	0.0000	0	102
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.56 C_6 for VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Task 1			Task 2			Task 3			Task 4		
	C_6	C_{6_f}	C_{6_b}	C_6	C_{6_f}	C_{6_b}	C_6	C_{6_f}	C_{6_b}	C_6	C_{6_f}	C_{6_b}
2007100302	0.0338	7.0000	207	0.0978	61.0000	624	0.1086	24.0000	221	0.1546	30.0000	194
2007100402	0.0338	7.0000	207	0.0796	34.0000	427	0.1076	24.0000	223	0.1622	30.0000	185
2007100701	0.0338	7.0000	207	0.0952	12.0000	126	0.1356	24.0000	177	0.1622	30.0000	185
2007100801	0.0470	7.0000	149	0.0585	22.0000	376	0.1076	24.0000	223	0.1622	30.0000	185
2007100802	0.0363	7.0000	193	0.1326	24.0000	181	0.1076	24.0000	223	0.1622	30.0000	185
2007100903	0.0338	7.0000	207	0.1326	24.0000	181	0.1076	24.0000	223	0.1622	30.0000	185
2007101001	0.0338	7.0000	207	0.0796	34.0000	427	0.0971	27.0000	278	0.1622	30.0000	185
2007101202	0.0338	7.0000	207	0.2692	21.0000	78	0.1076	24.0000	223	0.1622	30.0000	185
2007101301	0.0338	7.0000	207	0.0796	34.0000	427	0.1076	24.0000	223	0.1622	30.0000	185
2007101702	0.0338	7.0000	207	0.0838	31.0000	370	0.1721	21.0000	122	0.1622	30.0000	185
Average:	0.0354			0.1109			0.1159			0.1614		

B.2.7 Navigation presentation data

Table B.57 P_1^{gt} for locating VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	P_1^{gt}	$P_{1_f}^{gt}$	$P_{1_b}^{gt}$	P_1^{gt}	$P_{1_f}^{gt}$	$P_{1_b}^{gt}$	P_1^{gt}	$P_{1_f}^{gt}$	$P_{1_b}^{gt}$	P_1^{gt}	$P_{1_f}^{gt}$	$P_{1_b}^{gt}$
2007100302	0.4564	68	149	0.5205	89	171	0.5205	89	171	0.5345	93	174
2007100402	0.5176	44	85	0.5205	89	171	0.5205	89	171	0.5345	93	174
2007100701	0.5176	44	85	0.5233	90	172	0.4831	114	236	0.4937	118	239
2007100801	0.4564	68	149	0.5233	90	172	0.5205	89	171	0.5345	93	174
2007100802	0.5176	44	85	0.5233	90	172	0.5205	89	171	0.5345	93	174
2007100903	0.5176	44	85	0.5233	90	172	0.5205	89	171	0.5345	93	174
2007101001	0.5176	44	85	0.5233	90	172	0.5205	89	171	0.5345	93	174
2007101202	0.4564	68	149	0.5233	90	172	0.5205	89	171	0.5345	93	174
2007101301	0.5176	44	85	0.5233	90	172	0.5205	89	171	0.5345	93	174
2007101702	0.5176	44	85	0.5233	90	172	0.5205	89	171	0.5345	93	174
Average:	0.4993			0.5227			0.5167			0.5304		

Table B.58 P_2^{gt} for locating VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	P_2^{gt}	$P_{2_f}^{gt}$	$P_{2_b}^{gt}$	P_2^{gt}	$P_{2_f}^{gt}$	$P_{2_b}^{gt}$	P_2^{gt}	$P_{2_f}^{gt}$	$P_{2_b}^{gt}$	P_2^{gt}	$P_{2_f}^{gt}$	$P_{2_b}^{gt}$
2007100302	0.5000	1	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100402	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100701	0.0000	0	1	0.0000	0	2	0.3333	1	3	0.3333	1	3
2007100801	0.5000	1	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100802	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100903	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101001	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101202	0.5000	1	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101301	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101702	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
Average:	0.1500			0.0000			0.0333			0.0333		

Table B.59 P_3^{gt} for locating VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	P_3^{gt}	$P_{3_f}^{gt}$	$P_{3_b}^{gt}$	P_3^{gt}	$P_{3_f}^{gt}$	$P_{3_b}^{gt}$	P_3^{gt}	$P_{3_f}^{gt}$	$P_{3_b}^{gt}$	P_3^{gt}	$P_{3_f}^{gt}$	$P_{3_b}^{gt}$
2007100302	0.0000	0	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100402	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100701	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	3
2007100801	0.0000	0	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100802	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100903	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101001	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101202	0.0000	0	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101301	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101702	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.60 P_4^{gt} for locating VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	P_4^{gt}	$P_{4_f}^{gt}$	$P_{4_b}^{gt}$	P_4^{gt}	$P_{4_f}^{gt}$	$P_{4_b}^{gt}$	P_4^{gt}	$P_{4_f}^{gt}$	$P_{4_b}^{gt}$	P_4^{gt}	$P_{4_f}^{gt}$	$P_{4_b}^{gt}$
2007100302	1.0000	2	2	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007100402	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007100701	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	3	3
2007100801	1.0000	2	2	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007100802	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007100903	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007101001	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007101202	1.0000	2	2	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007101301	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007101702	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
Average:	1.0000			1.0000			1.0000			1.0000		

Table B.61 P_5^{gt} for locating VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	P_5^{gt}	$P_{5_f}^{gt}$	$P_{5_b}^{gt}$	P_5^{gt}	$P_{5_f}^{gt}$	$P_{5_b}^{gt}$	P_5^{gt}	$P_{5_f}^{gt}$	$P_{5_b}^{gt}$	P_5^{gt}	$P_{5_f}^{gt}$	$P_{5_b}^{gt}$
2007100302	1.0000	2	2	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007100402	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007100701	1.0000	1	1	1.0000	2	2	1.0000	3	3	1.0000	3	3
2007100801	1.0000	2	2	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007100802	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007100903	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007101001	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007101202	1.0000	2	2	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007101301	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
2007101702	1.0000	1	1	1.0000	2	2	1.0000	2	2	1.0000	2	2
Average:	1.0000			1.0000			1.0000			1.0000		

B.2.8 Navigation interaction data

Table B.62 I_1^{gt} for locating VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1	Nav. For Task 2	Nav. For Task 3	Nav. For Task 4
	I_1^{gt}	I_1^{gt}	I_1^{gt}	I_1^{gt}
2007100302	0	0	0	0
2007100402	0	0	0	0
2007100701	0	1	0	0
2007100801	0	0	0	0
2007100802	0	0	0	0
2007100903	0	0	0	0
2007101001	0	0	0	0
2007101202	0	0	0	0
2007101301	0	0	0	0
2007101702	0	0	0	0
Average:	0.0000	0.1000	0.0000	0.0000

B.2.9 Navigation efficiency data

According to the navigation architecture of VCalendar 1.5.3.1, the reaching distances for Task 1, 3, and 4 are respectively 1, 3, and 4. Task 2 was not directly supported by VCalendar 1.5.3.1, in other words, the subjects had to come up with their own ways to make this task up on the fly. We assume the subjects realized this fact after they had searched for it to 1 level deeper than the least appropriate depth, i.e., Task 2's reaching distance was assumed to be at least 5. The breadth of the navigation architecture $W_{\max} = 31$.

Then, according to formula (5-10), we get the average probability reaching distance of the website:

For Case 1: $D_{ap} = 3.2500$;

For Case 2: $D_{ap} = 2.6667$.

According to formula (5-12), (5-13), and (5-11), we got the efficiency of the navigation system of the website:

For Case 1: $E_{nav} = 0.3938$;

For Case 2: $E_{nav} = 0.5250$.

B.2.10 Navigation effectiveness data

According to formula (5-14), the effectiveness of entire navigation system $R_{nav} = I_{nav}$.

B.2.11 Navigation satisfaction data

Table B.63 S^{gt} for locating VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	S^{gt}	S_1^{gt}	S_2^{gt}	S^{gt}	S_1^{gt}	S_2^{gt}	S^{gt}	S_1^{gt}	S_2^{gt}	S^{gt}	S_1^{gt}	S_2^{gt}
2007100302	0.8000	7	9	0.7500	7	8	0.8000	7	9	0.8000	7	9
2007100402	0.7500	8	7	0.7500	7	8	0.7500	7	8	0.7500	7	8
2007100701	0.8000	7	9	0.5000	5	5	0.8000	7	9	0.7500	7	8
2007100801	0.4000	5	3	0.5500	5	6	0.6000	6	6	0.6000	6	6
2007100802	0.5500	3	8	0.5500	4	7	0.8000	8	8	0.8000	8	8
2007100903	0.8000	7	9	0.8500	8	9	0.7000	7	7	0.7000	7	7
2007101001	0.8500	8	9	0.8000	8	8	0.7000	7	7	0.7000	7	7
2007101202	0.8000	8	8	0.7500	7	8	0.7500	8	7	0.8000	8	8
2007101301	0.6500	6	7	0.7000	7	7	0.7000	7	7	0.7000	7	7
2007101702	0.5500	4	7	0.4000	4	4	0.4000	4	4	0.4000	4	4
Average:	0.6950			0.6600			0.7000			0.7000		

B.2.12 Navigation consistency data

Table B.64 C_1^{gt} for locating VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	C_1^{gt}	$C_{1_f}^{gt}$	$C_{1_b}^{gt}$	C_1^{gt}	$C_{1_f}^{gt}$	$C_{1_b}^{gt}$	C_1^{gt}	$C_{1_f}^{gt}$	$C_{1_b}^{gt}$	C_1^{gt}	$C_{1_f}^{gt}$	$C_{1_b}^{gt}$
2007100302	0.5000	1	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100402	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100701	0.0000	0	1	0.0000	0	2	0.3333	1	3	0.3333	1	3
2007100801	0.5000	1	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100802	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100903	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101001	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101202	0.5000	1	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101301	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101702	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
Average:	0.1500			0.0000			0.0333			0.0333		

Table B.65 C_2^{gt} for locating VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	C_2^{gt}	$C_{2_f}^{gt}$	$C_{2_b}^{gt}$	C_2^{gt}	$C_{2_f}^{gt}$	$C_{2_b}^{gt}$	C_2^{gt}	$C_{2_f}^{gt}$	$C_{2_b}^{gt}$	C_2^{gt}	$C_{2_f}^{gt}$	$C_{2_b}^{gt}$
2007100302	0.3310	48	145	0.4561	78	171	0.4561	78	171	0.4425	77	174
2007100402	0.4588	39	85	0.4561	78	171	0.4561	78	171	0.4425	77	174
2007100701	0.4588	39	85	0.4535	78	172	0.3750	87	232	0.3660	86	235
2007100801	0.3310	48	145	0.4535	78	172	0.4561	78	171	0.4425	77	174
2007100802	0.4588	39	85	0.4535	78	172	0.4561	78	171	0.4425	77	174
2007100903	0.4588	39	85	0.4535	78	172	0.4561	78	171	0.4425	77	174
2007101001	0.4588	39	85	0.4535	78	172	0.4561	78	171	0.4425	77	174
2007101202	0.3310	48	145	0.4535	78	172	0.4561	78	171	0.4425	77	174
2007101301	0.4588	39	85	0.4535	78	172	0.4561	78	171	0.4425	77	174
2007101702	0.4588	39	85	0.4535	78	172	0.4561	78	171	0.4425	77	174
Average:	0.4205			0.4540			0.4480			0.4349		

Table B.66 C_3^{gt} for locating VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	C_3^{gt}	$C_{3_f}^{gt}$	$C_{3_b}^{gt}$	C_3^{gt}	$C_{3_f}^{gt}$	$C_{3_b}^{gt}$	C_3^{gt}	$C_{3_f}^{gt}$	$C_{3_b}^{gt}$	C_3^{gt}	$C_{3_f}^{gt}$	$C_{3_b}^{gt}$
2007100302	0.0000	0	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100402	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100701	0.0000	0	1	0.0000	0	2	0.0000	0	3	0.0000	0	3
2007100801	0.0000	0	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100802	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007100903	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101001	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101202	0.0000	0	2	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101301	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
2007101702	0.0000	0	1	0.0000	0	2	0.0000	0	2	0.0000	0	2
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.67 C_4^{gt} for locating VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	C_4^{gt}	$C_{4_f}^{gt}$	$C_{4_b}^{gt}$	C_4^{gt}	$C_{4_f}^{gt}$	$C_{4_b}^{gt}$	C_4^{gt}	$C_{4_f}^{gt}$	$C_{4_b}^{gt}$	C_4^{gt}	$C_{4_f}^{gt}$	$C_{4_b}^{gt}$
2007100302	0.0000	0	145	0.0000	0	171	0.0000	0	171	0.0000	0	174
2007100402	0.0000	0	85	0.0000	0	171	0.0000	0	171	0.0000	0	174
2007100701	0.0000	0	85	0.0000	0	172	0.0000	0	232	0.0000	0	235
2007100801	0.0000	0	145	0.0000	0	172	0.0000	0	171	0.0000	0	174
2007100802	0.0000	0	85	0.0000	0	172	0.0000	0	171	0.0000	0	174
2007100903	0.0000	0	85	0.0000	0	172	0.0000	0	171	0.0000	0	174
2007101001	0.0000	0	85	0.0000	0	172	0.0000	0	171	0.0000	0	174
2007101202	0.0000	0	145	0.0000	0	172	0.0000	0	171	0.0000	0	174
2007101301	0.0000	0	85	0.0000	0	172	0.0000	0	171	0.0000	0	174
2007101702	0.0000	0	85	0.0000	0	172	0.0000	0	171	0.0000	0	174
Average:	0.0000			0.0000			0.0000			0.0000		

Table B.68 C_5^{gt} for locating VCalendar 1.5.3.1 task 1, task 2, task 3, and task 4

Subject Code	Nav. For Task 1			Nav. For Task 2			Nav. For Task 3			Nav. For Task 4		
	C_5^{gt}	$C_{5_f}^{gt}$	$C_{5_b}^{gt}$	C_5^{gt}	$C_{5_f}^{gt}$	$C_{5_b}^{gt}$	C_5^{gt}	$C_{5_f}^{gt}$	$C_{5_b}^{gt}$	C_5^{gt}	$C_{5_f}^{gt}$	$C_{5_b}^{gt}$
2007100302	0.0889	16.0000	180	0.0300	6.0000	200	0.0300	6.0000	200	0.0290	6.0000	207
2007100402	0.0303	3.0000	99	0.0300	6.0000	200	0.0300	6.0000	200	0.0290	6.0000	207
2007100701	0.0303	3.0000	99	0.0297	6.0000	202	0.0674	19.0000	282	0.0655	19.0000	290
2007100801	0.0889	16.0000	180	0.0297	6.0000	202	0.0300	6.0000	200	0.0290	6.0000	207
2007100802	0.0303	3.0000	99	0.0297	6.0000	202	0.0300	6.0000	200	0.0290	6.0000	207
2007100903	0.0303	3.0000	99	0.0297	6.0000	202	0.0300	6.0000	200	0.0290	6.0000	207
2007101001	0.0303	3.0000	99	0.0297	6.0000	202	0.0300	6.0000	200	0.0290	6.0000	207
2007101202	0.0889	16.0000	180	0.0297	6.0000	202	0.0300	6.0000	200	0.0290	6.0000	207
2007101301	0.0303	3.0000	99	0.0297	6.0000	202	0.0300	6.0000	200	0.0290	6.0000	207
2007101702	0.0303	3.0000	99	0.0297	6.0000	202	0.0300	6.0000	200	0.0290	6.0000	207
Average:	0.0479			0.0298			0.0337			0.0326		